Title	A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners
Authors(s)	Ayala-Rivera, Vanessa; McDonagh, Patrick; Cerqueus, Thomas; Murphy, Liam, B.E.
Publication date	2014
Publication information	Transactions on Data Privacy, 7 (3): 337-370
Publisher	Transactions on Data Privacy
Link to online version	http://www.tdp.cat/issues11/abs.a169a14.php
Item record/more information	http://hdl.handle.net/10197/9109

A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners

Vanessa Ayala-Rivera[*], **Patrick McDonagh**[**], **Thomas Cerqueus**[*], **Liam Murphy**[*]

[*]Lero@UCD, School of Computer Science and Informatics, University College Dublin, Ireland.

[**]Lero@DCU, School of Electronic Engineering, Dublin City University, Ireland.

E-mail: vanessa.ayala-rivera@ucdconnect.ie, patrick.mcdonagh@dcu.ie, thomas.cerqueus@ucd.ie, liam.murphy@ucd.ie

Abstract. The vast amount of data being collected about individuals has brought new challenges in protecting their privacy when this data is disseminated. As a result, Privacy-Preserving Data Publishing has become an active research area, in which multiple anonymization algorithms have been proposed. However, given the large number of algorithms available and limited information regarding their performance, it is difficult to identify and select the most appropriate algorithm given a particular publishing scenario, especially for practitioners. In this paper, we perform a systematic comparison of three well-known k-anonymization algorithms to measure their efficiency (in terms of resources usage) and their effectiveness (in terms of data utility). We extend the scope of their original evaluation by employing a more comprehensive set of scenarios: different parameters, metrics and datasets. Using publicly available implementations of those algorithms, we conduct a series of experiments and a comprehensive analysis to identify the factors that influence their performance, in order to guide practitioners in the selection of an algorithm. We demonstrate through experimental evaluation, the conditions in which one algorithm outperforms the others for a particular metric, depending on the input dataset and privacy requirements. Our findings motivate the necessity of creating methodologies that provide recommendations about the best algorithm given a particular publishing scenario.

Keywords. Privacy-Preserving Data Publishing, k-Anonymity, Algorithms, Performance

1 Introduction

Currently, the volumes of generated data grow exponentially every year [32]. Among this data, there is an increasing amount of personal information contained within. This fact has attracted the attention of those interested in creating more tailored and personalized services, based on the demographic information available. For this reason, businesses and organizations in various sectors collect personal data that may be shared under different circumstances (for either monetary, societal or legal reasons). Nevertheless, this phenomenon has brought new challenges to protect the privacy of the people represented in the published datasets.

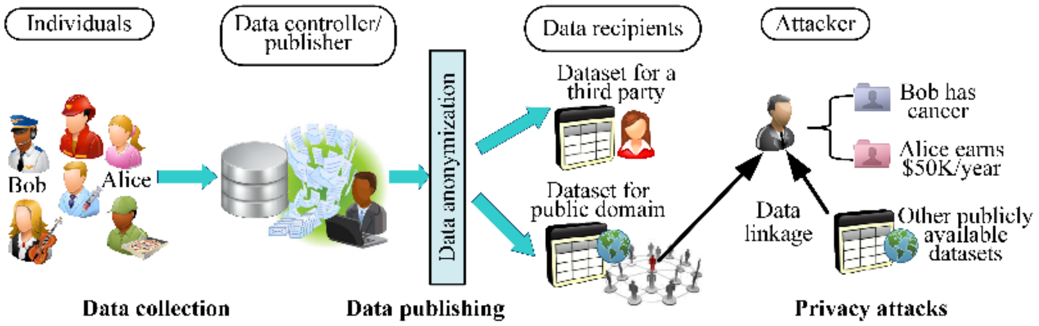


Figure 1: Overview of Privacy-Preserving Data Publishing

Consequently, Privacy-Preserving Data Publishing (PPDP) has become an area of interest for researchers and practitioners. A typical scenario of PPDP is depicted in Figure 1, which shows the different phases for the data processing. One key assumption of the PPDP model is that attackers can be found among the data recipients, who intend to uncover sensitive information about individuals. Therefore, the objective of PPDP techniques is to modify the data by making it less specific, in such a way that the individuals' privacy is protected; while aiming to retain the usefulness of the anonymized data. The essence of PPDP is to produce datasets that have good utility for a variety of tasks, as commonly, all the potential usage scenarios for the data are unknown at the time of publication. For example, under open data initiatives [2], it is impossible to identify all the data recipients. Thus, any data controller involved in the sharing of personal data needs to apply privacy-preserving mechanisms [35].

However, this is not a trivial task, as practitioners are not necessarily experts in the area of data privacy [27, 33]. Moreover, it is often the case that no methodologies exist that ensures that anonymization is conducted effectively in an organization. This can lead practitioners to employ simple methods of de-identification (e.g., removing all direct identifiers such as names and social security numbers), before releasing data.

Nonetheless, it has been proven that this approach alone is not enough to preserve privacy [10, 50, 57]. This issue occurs because it is still possible to combine different datasets or have background knowledge about individuals, in order to make inferences about their identity. The re-identification of an individual is achieved by linking attributes, known as *quasi-identifiers* (QIDs), such as gender, date of birth or ZIP code.

As a result, multiple anonymization algorithms have being proposed in the area of PPDP. Nevertheless, the selection of the most appropriate algorithm for a given publishing scenario is challenging for practitioners: Not only there is a plethora of anonymization algorithms from which one can choose, but every newly introduced algorithm claims a particular superiority over the others. The original experimental evaluations of these algorithms are usually limited, as they are mainly focused on demonstrating the benefits of the proposed algorithm over some of the previously proposed ones. This situation often narrows the scope of their evaluation with respect to the experimental configurations employed (e.g., using a single comparison metric, omitting scenarios). Additionally, in the cases where the authors introduce a new metric, the metric tends to favor the proposed algorithm due to the particular aspects measured by the metric.

The aforementioned situations might confuse practitioners, leading them to incorrectly assume that if an algorithm outperforms others for a particular metric, this algorithm can be considered the best regardless of the input parameters. However, this behavior is not guaranteed, as the performance of an algorithm can vary when a new input dataset with different characteristics is anonymized, or the configuration used to test the algorithm changes. Considering these challenges, we believe that there is a strong requirement to extend the existing evaluations of these anonymization algorithms to cover a more comprehensive set of experimental configurations. As a consequence, the objective of this work is to provide practitioners with more detailed explanations of the reasons behind the performance variations of the algorithms. The aim behind this is to facilitate the practical application and adoption of PPDP technology. In our work, a publishing scenario represents a situation where a data controller is interested in releasing microdata (involving personal data) to third parties. Some example scenarios are found in [35], and include: a hospital providing information about patient admissions, a school sharing student education data, a retailer sharing data about customers, etc.

In our study, we focus on k-anonymity because it is a fundamental principle of privacy and, contrary to other models which are too restrictive to be practical (e.g., entropy ℓ-diversity [47]) or difficult to be achieved for some scenarios (e.g., ℓ-diversity [45]), its conceptual simplicity has made it widely discussed and adopted in a variety of domains such as healthcare [28, 41], data mining [36, 66] and statistical disclosure control [25]. Moreover, despite the fact that some works [45, 47] have shown that k-anonymity is susceptible to certain attacks (e.g., homogeneity and background knowledge), every newly proposed model has weaknesses as well [18, 19, 26, 52]. In [38, 58], the authors point out the relevance of the k-anonymity model as a basis to propose more secure models. They argue that some of the newly introduced models offer enhancements to k-anonymity, so they cannot stand alone and must be accompanied by k-anonymity [59, 60, 63]; therefore, they cannot replace k-anonymity. Furthermore, the development of new algorithms based on k-anonymity is still undertaken by researchers [8, 38, 41, 49, 61]. Similarly, k-anonymity serves as base for new anonymization techniques on different contexts (e.g., social networks [64], location-based services [67, 69], etc.) Finally, the set of k-anonymization algorithms we selected for this study, have been widely cited by the community and have become representative in the PPDP area (this is further discussed in Section 3).

The results of this work show that the selection of the preferable algorithm in a given situation (i.e., publishing scenario) depends on multiple factors, such as the characteristics of the input dataset and the desired privacy requirements. Moreover, our findings motivate the necessity of creating methodologies that could assist data publishers in the process of making informed decisions about which anonymization algorithm to use for their particular scenario. This could be achieved by providing recommendations about the best performing algorithm, given a set of input parameters (e.g., datasets, privacy requirement).

In this paper the main contributions are:

- A comprehensive comparison of k-anonymization algorithms in terms of efficiency (resources usage) and effectiveness (data utility).
- An extensive analysis of the effects of the privacy parameters and some aspects of the datasets on the anonymization process.
- Identification and analysis of key factors to consider when choosing an anonymization algorithm and data utility metrics.
- An empirical demonstration that the "best" algorithm in a given situation is influenced by multiple factors.

This paper is structured as follows: Section 2 provides some background information and reviews related work. Section 3 presents an overview of the selected k-anonymization algorithms. Section 4 describes the datasets used. Section 5 presents our comparison methodology. Section 6 discusses our experimental evaluation and results. Finally, Section 7 draws some conclusions and provides some directions for future work.

2 Background and Related Work

The following paragraphs briefly summarize the main concepts and work related to the data privacy research area:

2.1 Privacy Models

Multiple models have been proposed to offer formal guarantees about the protection of an individual's privacy. These models have been developed considering different attack scenarios to the data. For example, assuming diverse levels of background knowledge from an attacker that could lead to information disclosure. Examples of well-known models are k-anonymity [53, 57], ℓ-diversity [47], t-closeness [45] and differential privacy [26].

Among these models, we focus on k-anonymity, because, as previously discussed in Section 1, this model is practical and can be easily achieved in most cases. Moreover, although it has been pointed out that k-anonymity is vulnerable to certain attacks, it enables general-purpose data publication with reasonable utility. This is in contrast to more robust models (e.g., differential privacy) which might hamper the utility of anonymized data in order to preserve a more rigorous guarantee of privacy [55]. These characteristics can make k-anonymity attractive to practitioners, who can adopt it in their organizations in order to have an anonymization strategy with a formal guarantee of privacy.

k-Anonymity. It was the first model proposed for microdata anonymization and it is the base from which further extensions have been developed. The definition of k-anonymity for a table is as follows [53]: *"Let T(A_1,...,A_n) be a table and QI be a quasi-identifier associated with it. T is said to satisfy k-anonymity with respect to QI iff each sequence of values in T[QI] appears at least with k occurrences in T[QI]"*.

The k-anonymity model consists in altering the QID attributes, mostly through generalization and suppression operations, to create groups of records sharing the same QID values called *equivalence classes* (*EQs*). Hence, making each record indistinguishable from a group of at least *k-1* other records. The objective is to make QIDs imprecise and therefore less informative, preventing linking an individual to a record in the released table [56]. In our comparison, we consider a set of algorithms implementing k-anonymity, as they constitute an important body of work in the literature.

2.2 Anonymization Operations

An anonymization algorithm can employ different operations to achieve the desired level of privacy. Among these, deterministic mechanisms represent a more suitable option when the aim is to preserve the truthfulness of the data [16]. Some examples of these are *bucketization* [65], *microaggregation* [21, 25], and *generalization and suppression* [53, 56, 57]. Since the algorithms evaluated in our comparative study use generalization and suppression, following, we provide more details about these operations.

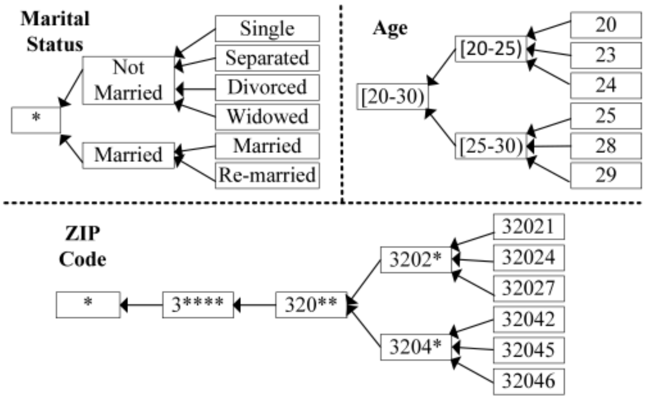


Figure 2: VGHs for *marital status*, *age* and *ZIP code*.

Generalization and Suppression. These operations have received much attention in the literature [4, 11, 36, 51]. Suppression consists in replacing some of the original data with a special value (e.g., "*") to indicate that this data is not disclosed. Generalization (also called *recoding*) consists in replacing the values of an attribute with less specific but consistent values; often employing a *value generalization hierarchy* (VGH), such as the ones shown in Figure 2. The values at the lowest level (right) are in the ground domain of the attribute, which correspond to the most specific values (original values). The highest level (left) showing the "*" value, corresponds to the maximum generalization or full suppression of the value. Recoding can be performed in a global (full-domain generalization) or local scheme. Local recoding can apply different rules to the same instances of attributes, in such a way that some instances remain with the specific values, while others are generalized. On the contrary, global recoding consists in applying the same generalization to all instances of an attribute, such that all values are generalized to the same level of the VGH. Global recoding is further classified in two types: single-dimensional [43, 56], which treats each attribute in the QID group independently; and multidimensional [44], which recodes a domain of n-vectors that are the cross product of the domains of the individual QID attributes [16].

Table 1: Microdata table of criminal records.

	ID	QIDs			SA
Tuple#	**Name**	**Marital Stat**	**Age**	**ZIP Code**	**Crime**
1	Joe	Separated	29	32042	Murder
2	Jill	Single	20	32021	Theft
3	Sue	Widowed	24	32024	Traffic
4	Abe	Separated	28	32046	Assault
5	Bob	Widowed	25	32045	Piracy
6	Amy	Single	23	32027	Indecency

As an illustrative example, consider Table 1 showing a table of criminal records. Among the attributes, *name* is the identifier (ID), *marital status*, *age* and *ZIP code* are the QIDs; and *crime* is the sensitive attribute (SA). Table 2 shows a 3-anonymous version of Table 1, which means that each tuple has at least two other tuples sharing the same values of the QIDs. To achieve anonymity, the ID has been removed and the QIDs have been generalized using

Table 2: A 3-anonymous version of Table 1 (k=3).

		QIDs			SA
Tuple#	**EQ**	**Marital Stat**	**Age**	**ZIP Code**	**Crime**
1	1	Not Married	[25-30)	3204*	Murder
4		Not Married	[25-30)	3204*	Assault
5		Not Married	[25-30)	3204*	Piracy
2	2	Not Married	[20-25)	3202*	Theft
3		Not Married	[20-25)	3202*	Traffic
6		Not Married	[20-25)	3202*	Indecency

a single-dimensional scheme: The *marital status* has been replaced with a less specific but semantically consistent description, the *age* values have been replaced with ranges of values and the last digit of the *ZIP code* has been replaced by a "*".

2.3 Anonymization Algorithms

There is a large body of k-anonymization algorithms proposed in the literature across the various data privacy domains: PPDP, Privacy-Preserving Data Mining (PPDM) and Statistical Disclosure Control (SDC). In [36] an approach using a genetic algorithm is proposed which aims to preserve classification information in anonymized data. An iterative bottom-up generalization algorithm is presented in [62] offering a minimal k-anonymization for classification. kACTUS [40] is another k-anonymization algorithm focused in preserving the privacy in classification tasks using multidimensional suppression. Top-down specialization [31] is an algorithm that goes from the most generalized state of a table and specializes it according to a search metric, offering minimal k-anonymization. k-optimize [11] offers optimal anonymity using subtree generalization and record suppression with pruning techniques. In the SDC area, methods using microaggregation techniques are widely used for anonymization. Some of the most relevant techniques are maximum distance to average vector method (MDAV) [34], multivariate fixed-size microaggregation [22], minimum spanning tree partitioning [42], MDAV-generic [25] and variable-size MDAV [54]. For a more comprehensive description of these and other privacy-preserving algorithms, the reader is referred to [16, 20, 30].

2.4 Related Work

The selection of an appropriate algorithm to protect privacy when disseminating data is a general concern for data publishers. As a result, the comparison of multiple anonymization algorithms to evaluate the tradeoff between the data utility and privacy they offer, represents an important body of research work.

The authors of [51] compare a set of clustering-based k-anonymization algorithms. They discuss that the effectiveness of an anonymization solution is better assessed by the utility that it provides to target applications. Further to this, a framework is presented in [13], to compare different PPDM algorithms. Similarly, in the area of SDC, comparative studies and frameworks have been presented to evaluate the performance of microdata anonymization methods in terms of information loss and disclosure risk [23, 37]. In [24], the authors compare of a set of representative SDC methods in order to provide guidance in choosing a particular method.

Even though PPDP is closely related to such areas (PPDM and SDC), several differences exist between them. In the case of PPDM, the developed solutions are tightly coupled with the data mining tasks under consideration. Additionally, since PPDM algorithms usually rely on privacy models, there are a priori guarantees over the level of data protection that the algorithms offer. In the field of SDC, these techniques do not often preserve data truthfulness, as the main objective is to publish datasets that preserve as many statistical properties as possible from the original dataset. Furthermore, SDC solutions usually focus on evaluating the effectiveness of the anonymized data in a statistical manner; which is validated after the anonymization process. Moreover, most of the solutions have been designed for numerical attributes [48]. Whereas PPDP can employ some of the approaches developed for PPDM and SDC, these solutions cannot be easily adapted to PPDP as the main objective in this area is not to identify a specific usage scenario for the anonymized data. Instead, PPDP releases the anonymized data to multiple recipients who can use the data in many different ways. Therefore, it would not be suitable to evaluate the anonymization methods using comparative studies that only consider special purpose metrics (i.e., application dependent). This is because measures that take into account a particular usage scenario may only capture the utility of the anonymized data based on the requirements for that scenario. Instead, we argue that a set of metrics that can be applicable to most publishing scenarios provides a better approach towards performing a systematic comparison. Moreover, in our study, we provide an in-depth discussion of the results of our experiments, explaining the variations in performance of the algorithms. Our analysis also allows for the identification of aspects that each utility metric can capture, such that they can be used effectively.

3 Evaluated Algorithms

In our comparative study, we have selected three k-anonymization algorithms using generalization and suppression. We have chosen these based on the following reasons: (1) these algorithms have been extensively cited in the literature, (2) these algorithms use different strategies of anonymization allowing a more comprehensive evaluation, (3) a public implementation of these algorithms is available and (4) these algorithms can be evaluated within the same framework, allowing for a more fair comparison. In the following section, we describe the algorithms relevant to the scope of this work. We also present a schematic representation and an example for each of the algorithms, with the objective of making them easily comprehensible for practitioners.

3.1 Datafly

Datafly [56] is a greedy heuristic algorithm that performs single-dimensional full-domain generalization. Figure 3 depicts the main steps of the Datafly algorithm. It counts the frequency over the QID set and if k-anonymity is not yet satisfied, it generalizes the attribute having the most distinct values until k-anonymity is satisfied. Whereas this algorithm guarantees a k-anonymous transformation, it does not provide the minimal generalization [53]. An example of how anonymization is performed using Datafly can be seen in Appendix A.

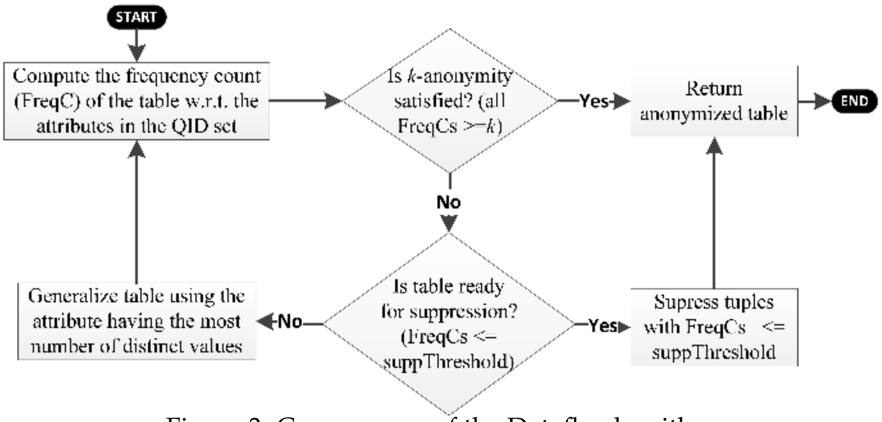

START
Compute the frequency count (FreqC) of the table w.r.t. the attributes in the QID set
Is k-anonymity satisfied? (all FreqCs >=k)
Yes
Return anonymized table
END
No
Generalize table using the attribute having the most number of distinct values
No
Is table ready for suppression? (FreqCs <= suppThreshold)
Yes
Supress tuples with FreqCs <= suppThreshold

Figure 3: Core process of the Datafly algorithm.

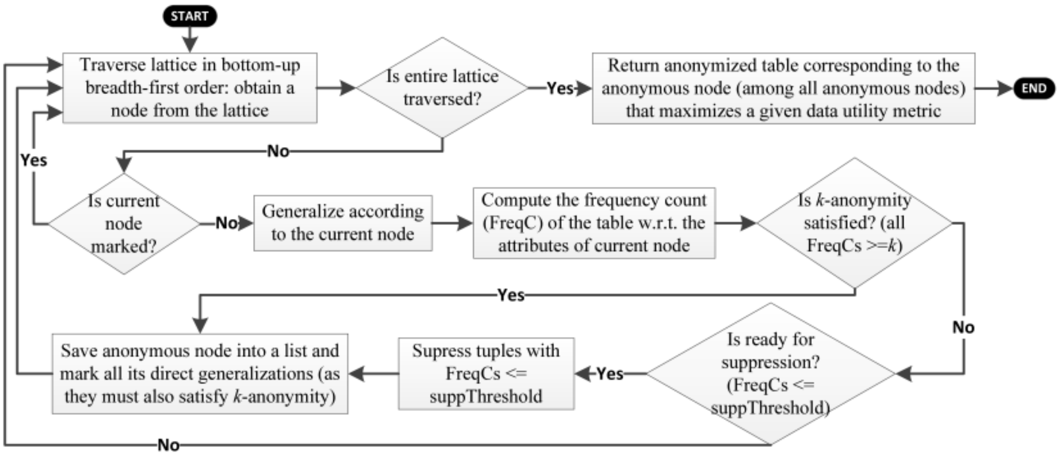

START
Traverse lattice in bottom-up breadth-first order: obtain a node from the lattice
Is entire lattice traversed?
Yes
Return anonymized table corresponding to the anonymous node (among all anonymous nodes) that maximizes a given data utility metric
END
No
Yes
Is current node marked?
No
Generalize according to the current node
Compute the frequency count (FreqC) of the table w.r.t. the attributes of current node
Is k-anonymity satisfied? (all FreqCs >=k)
Yes
No
Save anonymous node into a list and mark all its direct generalizations (as they must also satisfy k-anonymity)
Supress tuples with FreqCs <= suppThreshold
Yes
Is ready for suppression? (FreqCs <= suppThreshold)
No

Figure 4: Core process of the Incognito algorithm.

3.2 Incognito

Incognito [43] is a single-dimensional full-domain generalization algorithm that builds a generalization lattice and traverses it using a bottom-up breadth-first search. Figure 4 depicts the main steps of the Incognito algorithm. The number of valid generalizations for an attribute is defined by the depth of its VGH. Considering the VGHs in Figure 2, this would be: Two for *marital status* (M) and *age* (A); four for *ZIP code* (Z). An example of the generalization lattice created for that QID set can be seen in Appendix B. Each node in the lattice represents an anonymization solution. For example, the node <M1,A1,Z1> means that the three QIDs have been generalized once (one level up in their VGH), which is the solution shown in Table 2. To anonymize data, Incognito uses predictive tagging to reduce the search space. This means that, while traversing the lattice, if a node is found to satisfy k-anonymity, all of its direct generalizations can be pruned as it is guaranteed that they also satisfy k-anonymity. Unlike Datafly, Incognito produces an optimal solution. This means that the anonymized solution contains the maximal quantity of information according to a chosen information metric. The final solution is selected from a collection of solutions which all meet the given privacy requirement (e.g., satisfying k). In the implementation we evaluate, Incognito selects the solution that yields the maximum number of EQs. An example of how anonymization is performed using Incognito can be seen in Appendix C.

3.3 Mondrian

Mondrian [44] is a greedy multidimensional algorithm that partitions the domain space recursively into several regions, each of which contains at least k records. Figure 5 depicts the main steps of the Mondrian algorithm. It starts with the least specific (most generalized) value of the attributes in the QID set and specializes as partitions are performed on the data. To choose the dimension (i.e., attribute) on which to perform the partition, Mondrian uses the attribute with the widest (normalized) range of values. If multiple dimensions have the same width, the first one that enables an allowable cut (i.e., the cut does not cause a violation of k-anonymity) is selected. Once the dimension is selected, Mondrian uses a median partitioning approach to choose the *split value*, the value at which the partition will be performed. To find the median for an attribute, a *frequency sets* approach is used [44]. This is, the data is scanned adding up the frequencies for each of the unique values in the attribute until the median position is found. The value at which the median is found becomes the split value. An example of how anonymization is performed using Mondrian can be seen in Appendix D. In the original Mondrian paper, Mondrian was already compared with Incognito, however this comparison was only performed in terms of the data utility using the discernibility metric [11]. We replicate this experiment but also extend it by using other metrics to provide a more comprehensive study.

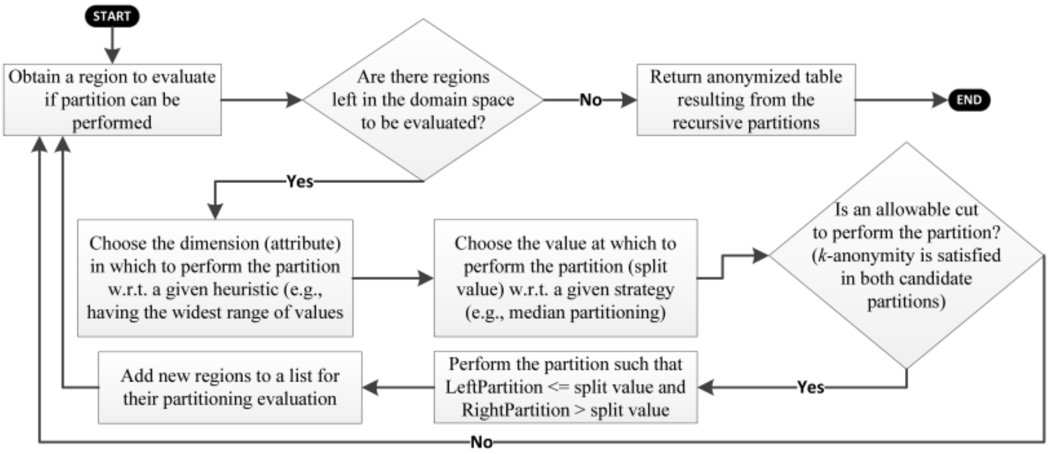

START
Obtain a region to evaluate if partition can be performed
Are there regions left in the domain space to be evaluated?
No
Return anonymized table resulting from the recursive partitions
END
Yes
Choose the dimension (attribute) in which to perform the partition w.r.t. a given heuristic (e.g., having the widest range of values
Choose the value at which to perform the partition (split value) w.r.t. a given strategy (e.g., median partitioning)
Is an allowable cut to perform the partition? (k-anonymity is satisfied in both candidate partitions)
Yes
Perform the partition such that LeftPartition <= split value and RightPartition > split value
Add new regions to a list for their partitioning evaluation
No

Figure 5: Core process of the Mondrian algorithm.

4 Datasets

In this section we provide a description of the datasets used in this comparative study and how they were configured for the performance evaluation of the algorithms.

4.1 Real Dataset

The first dataset is the *Adult* census dataset, from the UCI Machine Learning Repository [9] which has become the defacto benchmark to evaluate k-anonymization algorithms. We prepared this dataset by removing the records with missing values, thus leaving 30,162 valid records as in [36, 44, 45, 66]. The description for the *Adult* dataset is shown in Table 3. This table presents the attributes, their cardinalities (number of distinct values) and the height of the VGH defined for each attribute. In this last column, we specify within brackets, the number of VGH levels (i.e., available generalizations) of each particular attribute.

Table 3: Adult Dataset.

#	Attribute	Card.	Generalizations (Height)
1	Age	74	Taxonomy Tree (4) in 5-, 10-, 20-year ranges
2	Gender	2	Taxonomy Tree (1)
3	Race	5	Taxonomy Tree (1)
4	Marital Status	7	Taxonomy Tree (2)
5	Native Country	41	Taxonomy Tree (2)
6	Work Class	8	Taxonomy Tree (2)
7	Occupation	14	Taxonomy Tree (2)
8	Education	16	Taxonomy Tree (3)
9	Salary Class	2	Taxonomy Tree (1)

4.2 Synthetic Dataset

The second dataset is the *Irish* dataset, which we synthetically generated using Benerator [12], a Java open-source tool. This dataset was created by using the frequency count distributions from the Irish Census 2011 [1] as the probability weights in the data generation process. Further details about this generation process can be found in [7]. The original Irish Census dataset was composed of 3,550,246 records and it was scaled down to different sizes: 5k, 10k, 20k, 30k, 50k and 100k records. Figure 6 shows the comparison between the distribution from the original Irish Census data (a) and a synthetic dataset of 500k records (b) for the *age* attribute. It can be seen that the synthetic data preserves a high level of accuracy compared to the original distribution. The description for these datasets is shown in Table 4, which has the same structure as the previously described *Adult* dataset table.

Table 4: Irish Dataset.

#	Attribute	Card.	Generalizations (Height)
1	Age	70	Taxonomy Tree (4) in 5-, 10-, 20-year ranges
2	Gender	2	Taxonomy Tree (1)
3	County	34	Taxonomy Tree (3)
4	Marital Status	7	Taxonomy Tree (2)
5	Native Country	56	Taxonomy Tree (3)
6	Economic Status	9	Taxonomy Tree (3)
7	Industrial Group	22	Taxonomy Tree (2)
8	Education	10	Taxonomy Tree (3)
9	Field of Study	48	Taxonomy Tree (2)

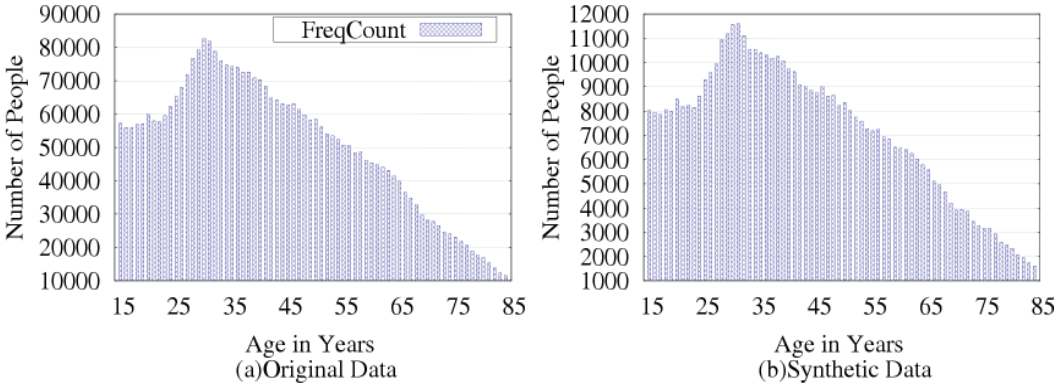


Figure 6: Data distribution for original and synthetic datasets for the *age* attribute.

5 Comparison Methodology

Performing a fair comparison of anonymization algorithms is inherently a challenging task, as every proposed algorithm uses different metrics and settings. The performance of the algorithms might vary among different combinations of datasets and input parameters (e.g., an algorithm may work well in some experimental configurations and perform poorly in others). As a result, it is important to assess the algorithms by defining a common configuration that reflects the parameters used in existing evaluations. Furthermore, a comparison requires the use of criteria that can be widely applicable to measure different aspects of the algorithms (e.g., efficiency and data utility). As previously discussed in Section 1, in PPDP all the potential usage scenarios are often unknown. Thus, in terms of data utility we have focused on metrics that can be applied to multiple scenarios (i.e., general-purpose metrics) and multiple types of algorithms. Below, we describe the metrics used in our comparison methodology.

5.1 Efficiency

An algorithm should be evaluated by the resources required to carry out the anonymization. This is an important dimension, as the anonymization process can be intensive in resource consumption. When resources are limited, they represent a constraint in the selection of an algorithm. Even when an algorithm achieves a good level of utility in the anonymized data, if it is not efficient in terms of memory consumption, CPU usage or execution time, then it might not be practical for use.

To measure the execution time, one could monitor the elapsed time for the different stages of the anonymization process: The data upload (either to memory or database), the anonymization itself and the data output. As the data upload and output steps do not change among algorithms, we will focus on measuring the anonymization time only.

To analyze the performance of the algorithms with respect to anonymization time, we examine the relationship between the anonymization time and the cost of the following *functional characteristics* of the algorithms: for Datafly, the number of generalization operations performed; for Incognito, the number of the nodes or generalization states evaluated in the created lattice; and for Mondrian, the number of partitions performed on the data.

In terms of computational resources, the memory and the CPU usage are also good indicators of efficiency. Occasionally, they may depend on the algorithm's implementation. For example, some implementations load all the data into main memory to speed up the execution of the anonymization. However, these types of approaches will not be scalable for large volumes of data. In this comparative study, we measure the memory consumption during the anonymization process. We do not report the results for CPU usage given that none of the algorithms were CPU intensive.

Given that the amounts of data to be anonymized can be considerably large, it is important to also assess the scalability of the algorithms. Whereas this might not represent an aspect to completely disregard an algorithm, depending on the planned workloads, the user should be aware of the efficiency of the algorithm to handle large datasets. In our work, scalability is also evaluated with respect to the memory usage and the anonymization time. We focus on the analysis of the trends followed by the algorithms as the size of the input dataset increases.

5.2 Data Utility

The lack of standardized metrics make it difficult to compare the algorithms to each other. For example, some metrics depend on the presence of a VGH to calculate data distortion [53, 56]. As they use the number of 'hops' or the depth of the VGH, these metrics are not applicable to other types of algorithms (e.g., those based on partitioning or clustering). In the statistics community, data utility is often measured by assessing the changes in the distribution of the underlying data (e.g., KL-divergence [39], L_1 norm [6]), or measuring homogeneity in clustering (e.g., sum of squares [22, 46]). Furthermore, other metrics measure the quality of the anonymization based on the utility of the data in a specific usage scenario. For example, for aggregate query answering [44, 68], for mining association rules [29] or for training classifiers [36]. These metrics are not well suited for PPDP because the data publisher usually does not know the exact usage scenarios for the data once published [23]. Otherwise, the data publisher could simply execute the target task on the original data and share the results with the interested parties instead of releasing the anonymized data. This lack of knowledge regarding possible applications has motivated the need for general-purpose metrics of utility, which use syntactic properties as a proxy for utility [15].

For our evaluation criteria, we have selected a set of general-purpose metrics because we believe that using metrics that can be widely applied to most of the anonymization algorithms represents a good step towards a standardization of their comparison. Below, we describe those metrics in the scope of this work.

Generalized Information Loss (GenILoss). This metric captures the penalty incurred when generalizing a specific attribute, by quantifying the fraction of the domain values that have been generalized [36]. In our evaluation, we use the normalized version of this metric, which was presented in [51]. Let L_i and U_i be the lower and upper bounds of an attribute *i*. A cell entry for attribute *i* is generalized to an interval *ij* defined by the lower L_{ij} and upper bound U_{ij} end points. The overall information loss of an anonymized table T^* can be calculated as:

$$GenILoss(T^*) = \frac{1}{|T| \cdot n} \times \sum_{i=1}^{n} \sum_{j=1}^{|T|} \frac{U_{ij} - L_{ij}}{U_i - L_i} \tag{1}$$

where T is the original table, n is the number of attributes and $|T|$ is the number of records.

This metric is based on the concept that data cell values that represent a larger range of values are less precise than the ones that represent a smaller range of values (e.g., *not married* is less specific than *single* or *divorced*). Lower values are desirable: 0 means no transformation (original data) and 1 means full suppression/maximum level of generalization of the data. Although this metric has only been employed in algorithms that use a generalization hierarchy, we employ it for non-hierarchical algorithms as well (e.g., Mondrian), as any interval of generalized values can be quantified in this way. Additionally, to calculate the penalty for categorical attributes (like *marital status*) using the above formula, we employ the approach of mapping each value to a numerical value (as explained in [36]). For example, for the *marital status* attribute (VGH depicted in Figure 2), *single* is mapped to 1, *separated* is mapped to 2 and so on, until *re-married* is mapped to 6. Therefore, the status of *not married* is represented by the interval [1-4], which covers the statuses from *single* to *widowed*.

To illustrate this metric consider Table 2 and the VGH depicted in Figure 2. For *marital status*, which is a categorical attribute, the GenILoss for cells with the value *not married* is $\frac{4-1}{6-1} = \frac{3}{5}$. For *age*, which is a numerical attribute, the GenILoss for cells with values in [25-30) is $\frac{29-25}{29-20} = \frac{4}{9}$. Finally, for *ZIP code*, the GenILoss for cells with the value *3204** is $\frac{6-4}{6-1} = \frac{2}{5}$. Following the GenILoss formula for the remaining cells, the GenILoss score for the whole table is $\frac{1}{6 \cdot 3} \times \frac{78}{9} = \frac{13}{27} = 0.48$.

Discernibility Metric (DM). This metric measures how indistinguishable a record is from others, by assigning a penalty to each record, equal to the size of the EQ to which it belongs [11]. If a record is suppressed, then it is assigned a penalty equal to the size of the input table. The overall DM score for a k-anonymized table T^* is defined by:

$$DM(T^*) = \sum_{\forall EQ s.t. |EQ| \geq k} |EQ|^2 + \sum_{\forall EQ s.t. |EQ| < k} |T| \cdot |EQ| \tag{2}$$

where T is the original table, $|T|$ is the number of records and $|EQ|$ is the size of the equivalence classes (anonymized groups) created after performing the anonymization. The idea behind this metric is that larger EQs represent more information loss, thus lower values for this metric are desirable. To illustrate this metric consider Table 2, as both EQs in the anonymized table have a size of 3, the DM score for the whole table is calculated as: $3^2 + 3^2 = 18$.

Average Equivalence Class Size Metric (C$_{AVG}$). This metric measures how well the creation of the EQs approaches the best case, where each record is generalized in an EQ of k records [44]. The objective is to minimize the penalty: a value of 1 would indicate the ideal anonymization in which the size of the EQs is the given k value. The overall C_{AVG} score for an anonymized table T^* is given by:

$$C_{AVG}(T^*) = \frac{|T|}{|EQs| \cdot k} \tag{3}$$

where T is the original table, $|T|$ is the number of records, $|EQs|$ is the total number of equivalence classes created and k is the privacy requirement. To illustrate this metric consider Table 2 showing 2 EQs, the C_{AVG} score for the whole table is calculated as: $\frac{6}{2 \cdot 3} = 1$.

6 Experimental Evaluation

6.1 Environment

The experiments were performed on an unloaded machine running 64-bit Ubuntu 12.04 LTS, with an Intel Xeon E5-2430 processor at 2.20GHz clock speed with 24 GB RAM, using Oracle Hotspot JVM version 7. The memory size of the machine allowed us to execute the experiments without triggering major garbage collections in Java (MaGC). Hence, obtaining neat and stable efficiency measurements, as any MaGC performance impact [14] was prevented. For the three algorithms, we used the implementations publicly available in the UT Dallas Anonymization Toolbox [3]. The fact that all algorithms are implemented using a common framework (e.g., using the same data structures for anonymization) allows for a fair performance comparison. In this implementation, the intermediate anonymization datasets are stored in a database. Therefore, the (RAM) memory is only consumed by data structures used during the anonymization, such as the generalization lattice, the list of the attributes which are part of the QID set and their corresponding VGHs. These implementations, written in Java, do not have a priori optimizations (i.e., pre-computation of frequency sets) that could give an advantage to one algorithm over the others. We introduced extra logic in the toolbox to measure the efficiency of the algorithms. Similarly, we developed a separate component to calculate the data utility metrics to measure the effectiveness of the algorithms. We used MySQL 5.5.34 as the database to store intermediary anonymization states, instead of the embedded SQLite database, which is the default in this toolbox. The reason for this choice is that MySQL is more scalable than SQLite when dealing with very large datasets.

6.2 Experimental Setup

For our experiment we used the *Adult* and *Irish* datasets as described in Section 4. The configurations used in these experiments are shown in Table 5. The parameters varied in these experiments are:

- $|QIDs|$: Defines the number of attributes that are part of the QID set.
- k-value: Defines the privacy level that must be satisfied by the anonymization algorithm. It represents the minimum size for the EQs in the anonymized solution.
- Dataset size: Corresponds to the number of records in the dataset.

The range of values used in these experiments were selected on the basis of those used in the original papers for Mondrian and Incognito. Additionally, we extended some of these parameters with higher values (e.g., in k-values) based on standards of generalization from the Census Bureau (e.g., any disclosed geographic region must contain at least 10,000 or 100,000 individuals)[18].

The suppression threshold defines the percentage of records that can be suppressed during the anonymization process in order to still consider a dataset as k-anonymous. It is set to zero for all the experiments, so all records are considered during the anonymization process.

Table 5: Experimental Configurations.

#	Experiment	Parameter settings	Datasets (size)
1	Varied $\|QIDs\|$	*k-value*= 2 , $\|QIDs\| \in$ [1..5]	Adult (30,162), Irish (30,000)
2	Varied k-value	*k-value* $\in$ {2, 5, 10, 25, 50, 100, 250, 500, 1000}, $\|QIDs\|$= 3	Adult (30,162), Irish (30,000)
3	Varied Size	*k-value*= 50, $\|QIDs\|$= 3	Irish (5k, 10k, 20k, 30k, 50k, 100k)

6.3 Results

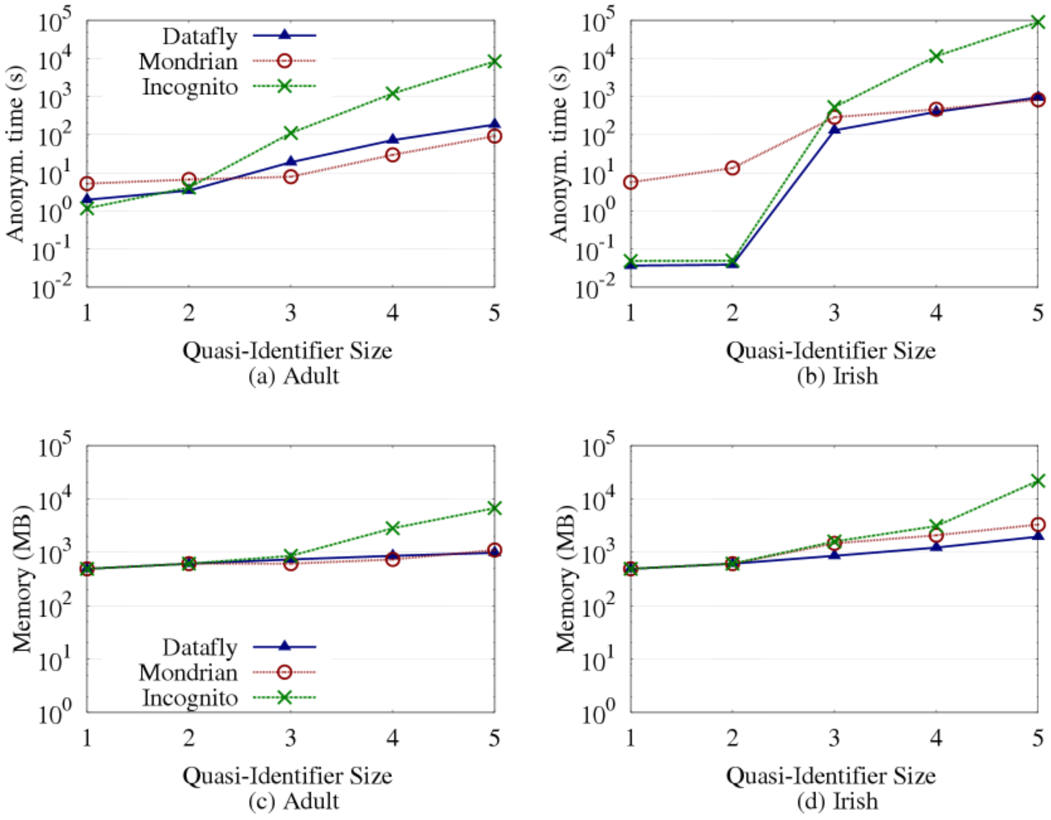
In this section we present the results for the experiments conducted using the configurations explained in the previous section. In each experiment, the results will be discussed in terms of the metrics relevant to that experiment. As explained previously in Section 5, lower values are better for all the metrics.

6.3.1 Experiment 1: Varied number of QIDs

In this experiment, we analyze the performance of the algorithms in both datasets as the number of QIDs increases. We use the notation $|QIDs|$ to refer to the number of attributes which are part of the QID set. A quasi-identifier of size *n* consists of the first *n* attributes listed for a dataset as shown in Tables 3 and 4 in Section 4. The experiments were carried out by varying the $|QIDs| \in$ [1..5].

Anonymization Time. Figures 7(a) and 7(b) show the results for the anonymization time as the number of QIDs increases. The first observation in these figures is that for the *Irish* dataset, Datafly and Incognito show an anonymization time close to 0 when $|QIDs| \in$ {1,2}. This behavior is caused by the fact that the original dataset already satisfies *2-anonymity* for this configuration, so these two algorithms do not perform any generalization. On the contrary, Mondrian anonymized the dataset because this algorithm does not check for k-anonymity as its stopping criteria. Instead, it tries to make all possible partitions to the data until no more cuts are allowed (even if the original dataset already satisfies k-anonymity).

Incognito is the worst performer for both datasets in almost all experimental configurations (except the ones where no generalization was performed). As expected, Incognito showed a time complexity of $O(2^{|QID|})$ [43, 17]. This can be observed in the figures, where Incognito shows the highest growth on the anonymization time with respect to the number of QIDs independently of the dataset. For example, for *Adult*, Incognito performs the anonymization between 13 and 90 times slower than Mondrian and between 5 and 45 times slower than Datafly, when $|QIDs| \geq$3. This increase is explained by the fact that the search space of anonymization solutions (nodes) becomes wider as more attributes are part of the QID set, which increases the time taken to traverse the generalization lattice and check for k-anonymity in each individual state. This growth, which happens in different degrees for each dataset, depends on the depth of the VGHs defined for each attribute, which determines the total number of nodes to be evaluated for the worst case (as it can be less if the lattice is pruned). For example, consider the case where $|QIDs|$ increases from 3 to 4, which shows the first major increase (note the logarithmic scale on the y-axis). For *Adult*,



(a) Adult

(b) Irish

(c) Adult

(d) Irish

Figure 7: Efficiency vs Number of QIDs (k=2).

the number of evaluated nodes grows from 17 (out of 20) to 50 (out of 60), whereas for *Irish* it grows from 10 (out of 40) to 91 (out of 120); this is why Incognito shows a more substantial rise for *Irish*.

We observed that Incognito performs well only when the number of quasi-identifiers is small. Moreover, it is important to consider not only the number of attributes in the QID set but also the number of possible generalization states that each attribute can have (i.e., VGH depth), as this factor influences the rate of increase for the anonymization time. Mondrian and Datafly show an acceptable performance (less than 16 minutes when $|QIDs|$=5). Overall, it can be noted that when comparing the performance of the three algorithms for both datasets, the algorithms perform better for *Adult* than for *Irish*; as the cost of the *functional characteristics* (mentioned in Section 5.1) is higher for *Irish*. Furthermore, the best performer regarding anonymization time is different in each dataset: Mondrian is the fastest for *Adult* once $|QIDs| \geq$3, while Datafly is the fastest for *Irish*.

Memory Consumption. Figures 7(c) and 7(d) show the results for memory consumption in the anonymization process as the number of QIDs increases. Datafly and Mondrian exhibit minor variance in terms of memory consumption but remain relatively stable as the number of QIDs increases. However, in the case of Incognito one disadvantage that can be observed is that it is memory intensive with respect to the increase in the number of

QIDs; reaching 21 GB of memory when $|QIDs|$=5. The growth of memory consumption in Incognito is possibly due to the increased number of nodes in the generalization lattice.

Generalized Information Loss (GenILoss). Figures 8(a) and 8(b) present the results for the data utility with respect to the GenILoss metric as the number of QIDs increases. It can be noted that the performance trends for each algorithm are different for both datasets. Datafly and Mondrian show an erratic behavior in *Adult* whereas for *Irish* they mostly show an increasing trend. Moreover, the information loss penalties for the *Adult* dataset are higher than in *Irish* for all the algorithms. For example, Mondrian performs well for *Irish* but it has the worst scores for *Adult* (when $|QIDs| \in$ {2,3}). The reason for such high values in Mondrian for *Adult* is the manner in which the partitioning is performed (median approach), as well as the skewed data distribution in the QIDs. More specifically, the fact that Mondrian is partitioning the data using a single attribute, instead of the two or three available in the QID set, causes the rest of the attributes in the QID set to retain their least specific values. This causes a high penalty for those attributes. For example, when $|QIDs|$=3 ({*age, gender, race*}), *age* is the only attribute where cuts are allowed, as *gender* and *race* cuts are discarded because the median partitioning condition is not satisfied.

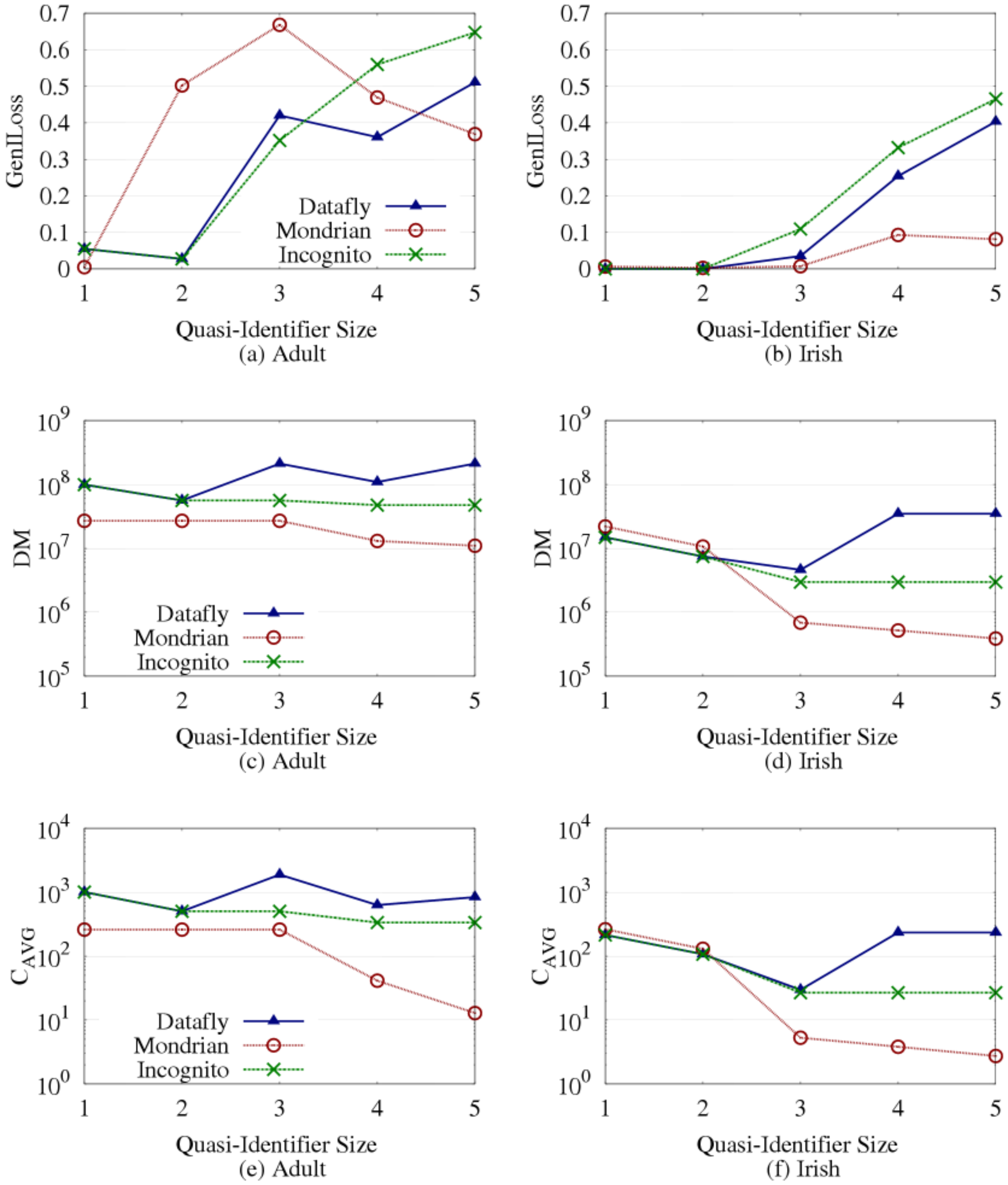
The approach used to find the median uses frequency sets, as discussed in Section 3.3. Let us consider the first iteration of Mondrian when the data is scanned to find the median value for the *gender* attribute. *Gender* has two values in its domain: *female* with a frequency count of 9,782 records and *male* with 20,380 records. When all records are considered, the median is found at position 15,081; which is reached when the frequency set for *male* is added to the count, consequently, the selected split value is *male*. Nevertheless, performing the cut at this value is not allowed because it does not satisfy the condition to have a minimum size of k records for each new partition. Hence, the data cannot be partitioned using this attribute and its values retain the least specific state, a characteristic mentioned in Section 3.3. This behavior persists in further iterations as the data distribution remains the same even at lower levels.

A similar situation occurs with the *race* attribute, which has the following distribution for *Adult*: *Amer-Indian-Eskimo* 286, *Asian-Pac-Islander* 895, *Black* 2,817, *Other* 231 and *White* 25,933. The selected split value is *White*, where the median value is found, but again, the cut is not allowable.

Whereas the median-partitioning technique aims to obtain uniform occupancy, this technique causes high information loss, when the data is skewed. This behavior demonstrates that Mondrian performs better with uniform distributions because the majority of the QIDs can be used to partition the data. Otherwise, the GenILoss score for Mondrian is high, as those attributes that could not be used for partitioning retain their least specific values.

For the other two algorithms, Datafly and Incognito, a notable trend in both datasets is that the GenILoss score tends to increase. This behavior is caused by the fact that in order to achieve k-anonymity, most of the attributes need to be generalized further, which degrades the utility of data. This confirms the curse of dimensionality, studied in [4], which is that the effectiveness of the anonymized data degrades when increasing dimensionality.

Discernibility Metric (DM). Figures 8(c) and 8(d) present the results for data utility with respect to DM as the number of QIDs increases. For the *Adult* dataset, it is clear that Mondrian is the best performer, as this algorithm aims to create finer EQs (EQs of small size). However for the *Irish* dataset, Mondrian performs the worst when $|QIDs| \in$ {1,2}. This is due to the fact that the original dataset already satisfies the k-anonymity for that configu-



(a) Adult

(b) Irish

(c) Adult

(d) Irish

(e) Adult

(f) Irish

Figure 8: Data utility vs Number of QIDs (k=2).

ration; nonetheless, Mondrian creates partitions. This situation causes the number of EQs (already formed in the original data) to decrease, but as a result, the size of each individual EQ (the number of records in each EQ) increases, which is not desirable for the DM metric. For example, when $|QIDs|$=1, the number of EQs in the original data is 70, whereas in the anonymized data using Mondrian it is 57.

Another observation is that DM does not capture the transformations performed on the QIDs. For example, when $|QIDs| \in$ [1..3] for *Adult*, Mondrian shows the same DM value (same level of data utility). This behavior is due to the fact that for these three experimental configurations, Mondrian performs partitions using only a single attribute (*age*); a situation that was explained in the previous GenILoss metric. Hence, the same EQs (anonymized groups) are always created for these three configurations. Once other attributes are added to the QIDs set (i.e., the number of QIDs increase), Mondrian is able to make partitions across all of them, thus improving the DM value.

Another aspect we identified that impacts the utility of the data when using Mondrian is the criteria used to select the dimension (attribute) in which the partition is performed (i.e., the attribute with the widest range of values), particularly, when multiple attributes have the same range of values. In the case of UT Dallas Toolbox implementation, it selects the first attribute; as discussed in Section 3.3. In our experiments, we observed that this decision affects the number of partitions that are created, by reducing the number of partitions in some cases.

Regarding Datafly and Incognito, they present the same values in terms of DM when $|QIDs| \in \{1,2\}$ as both algorithms achieve the same anonymization solution. For a larger number of QIDs, Incognito outperforms Datafly as the former finds the generalization that yields the maximum number of EQs (as explained in Section 3.2). Thus, in terms of metrics that are based on group sizes (such as DM and C$_{AVG}$), it is expected that Incognito will perform better than Datafly. However, Incognito does not outperform Mondrian, which benefits from its multidimensional approach.

When comparing the data utility of the algorithms between datasets with respect to DM, we observe that they perform better for *Irish*. The reason for this behavior is the *Irish* dataset distribution, which allows the algorithms to create finer EQs, hence reducing the DM score.

Average Equivalence Class Size (C$_{AVG}$). Figures 8(e) and 8(f) present the results for the C$_{AVG}$ metric as the number of QIDs increases. As can be seen in the figures, the trends for this metric are similar to the DM results for both datasets, but with different magnitudes. Mondrian appears to be superior in most of the scenarios, due to the number of the EQs created during anonymization; an aspect that Mondrian aims to maximize. Whenever the score for C$_{AVG}$ is the same for two or more algorithms, it means that those algorithms produced the same number of EQs. However, it does not necessarily imply that the size of their EQs is the same. This shows that C$_{AVG}$ does not capture the distribution of records among EQs.

Summary. For this experiment, the algorithms have a better performance for *Adult* than for *Irish* in terms of efficiency. For anonymization time and memory consumption, Datafly and Mondrian perform better. Both algorithms anonymized the data in reasonable time for the evaluated QID set sizes. While Datafly and Incognito exhibited an exponential growth, the magnitude of Incognito is much larger. Moreover, not only does the number of QIDs have an influence on time complexity, but so does the number of possible anonymization states (i.e., the height of the VGHs).

In terms of data utility, the algorithms showed a better performance in the *Irish* dataset thanks to the data distribution of the QIDs. For the data utility metrics based on the size of the EQs such as, DM and C$_{AVG}$, Mondrian outperforms the other algorithms. However, for GenILoss, it was shown that Mondrian is substantially affected by outliers, as median partitioning cannot be performed for some attributes, resulting in high GenILoss values.

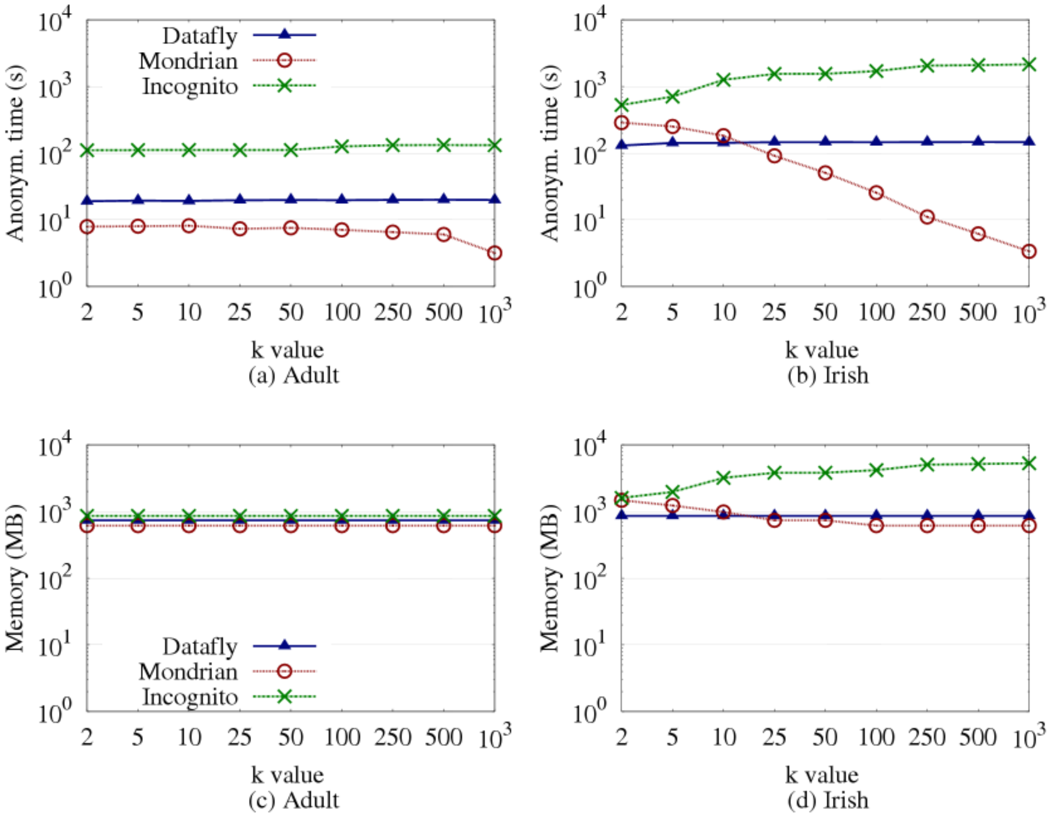


Figure 9: Efficiency vs Varied k-value ($|QIDs|$=3).

6.3.2 Experiment 2: Varied k value

In this experiment, we analyze the performance of the algorithms for both datasets as the value of k increases, using a configuration of $|QIDs|$=3.

Anonymization Time. Figures 9(a) and 9(b) show the results for anonymization time as the value of k increases. Intuitively, as the value of k increases, the number of generalizations needed to satisfy k-anonymity would be expected to increase (and consequently the anonymization time). This is because it is more difficult to satisfy higher levels of privacy. As can be seen in the figures, this increase is minimal for Datafly, where the trend looks almost steady for both datasets given the differences in the number of generalizations performed; between 4 and 5 for *Adult*, and between 2 and 7 for *Irish*.

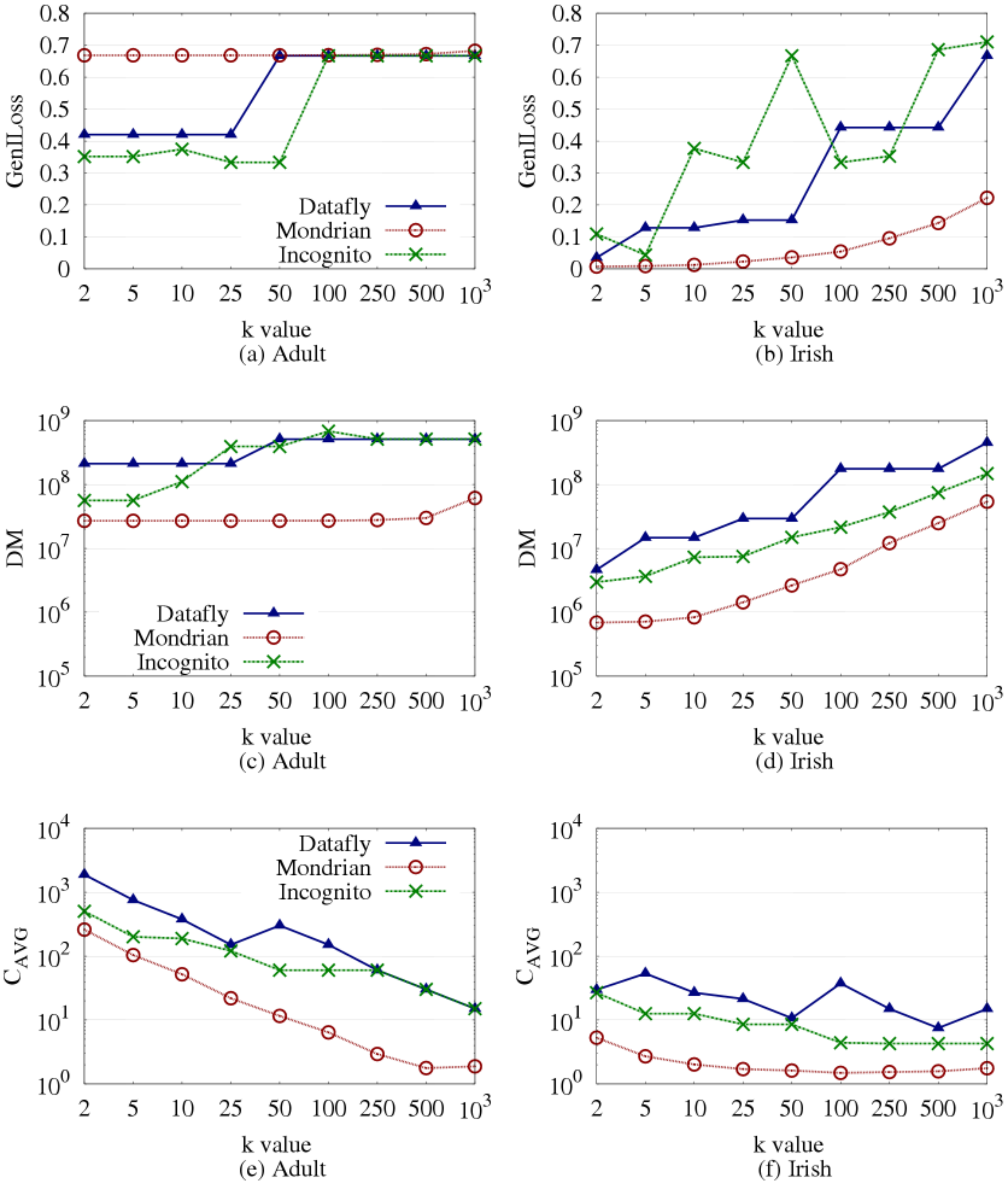
Incognito shows a low sensitivity to the growth of k for *Adult*, where the lowest anonymization time is 110.692 seconds (slightly below 2 minutes) and the highest is 132.59 seconds (slightly above 2 minutes). Incognito evaluates between 17 and 20 (out of 20) nodes from the generalization lattice for *Adult*. However, for *Irish*, the differences in the number of evaluated nodes is higher, between 10 and 39 (out of 40). Therefore, Incognito shows increasing trend for *Irish*, where the lowest anonymization time is 528.50 seconds (slightly below 9 minutes) and the highest is 2146.87 seconds (slightly below 36 minutes). The small differences in the number of generalizations or evaluated nodes indicate that varying the k parameter, with the $|QIDs|$ fixed, does not have a significant impact on the anonymization time for Datafly and Incognito for *Adult*.

On the other hand, Mondrian behaves differently, showing a decreasing trend for both datasets. This is because the number of possible partitions that can be performed (which satisfy k), decreases as the value of k increases. For example, for the upper and lower bounds of k, Mondrian creates between 15 and 57 partitions for *Adult* and between 32 and 5,664 partitions for *Irish*. The large difference between the number of partitions performed for *Adult* and *Irish* shows the influence that the data distribution of the QIDs has in Mondrian's performance, as the decreasing trend for this algorithm is more drastic for *Irish*.

Memory Consumption. Figures 9(c) and 9(d) present the results for memory consumption as the value of k increases. As can be seen in the figures, the variability in k does not have a major impact on the memory consumption for the algorithms. The trends for the three algorithms are relatively stable. Mondrian even showed a small decrease in memory consumption as k increases, whereas Incognito showed a small increase for one of the evaluated datasets. Datafly maintained a stable behavior for both datasets.

Generalized Information Loss (GenILoss). The results showing the utility of the anonymized data with respect to GenILoss as the value of k increases are depicted in Figures 10(a) and 10(b). For the *Adult* dataset, Mondrian shows the same information loss across all k values. This behavior is expected, as this metric captures the loss of precision in the generalized QID attributes, and for this experimental configuration (with a varying k value and a $|QIDs|$=3), Mondrian uses only one attribute for partitioning which causes the same GenILoss score regardless of the used k value. This situation also causes the GenILoss score of Mondrian to be higher for *Adult* than for *Irish*. This is because Mondrian starts the data partitioning with the attributes in their least specific state to gradually specialize them with each partition. So, if only one attribute is partitioned, the other attributes retain the most general value, incurring the maximum penalty for these attributes. For the same dataset (*Adult*), Datafly shows a sudden increase in the GenILoss score when the k value changes from 25 to 50. This occurs because Datafly produced one anonymization solution when $2\leq k \leq 25$ and another when $k \geq 50$. Incognito is the best performer with respect to GenILoss for this dataset, although, once k=100, the three algorithms reach the same GenILoss score.

For the *Irish* dataset, Incognito shows an erratic behavior. The high values (e.g. peaks shown when $k \in \{50,500,1000\}$) are caused because two out of three attributes in the QID set (i.e., *gender* and *county*) were generalized to their maximum level. The inclusion of the *gender* attribute in the generalization incurs a higher GenILoss score compared to the one incurred by other attributes that have a deeper VGH. This demonstrates the importance of not only considering the number of attributes in the QID set (which is fixed in this experiment), but also the VGH defined for the QIDs; as the inclusion of certain attributes in the generalization may degrade the data utility more. Based on this information, data publishers may consider excluding attributes (when possible) that deteriorate the data utility the most. Another alternative is to restructure the VGH for those attributes to offer more granularity in their generalization and avoid data degradation. For the same dataset (*Irish*), Datafly follows an increasing trend, showing a substantial rise when k changes from 50 to 100. This occurs because the number of generalizations for the *age* attribute were doubled (i.e., *age* values were grouped in wider intervals), making the resulting values less specific. Mondrian is shown to be the best performer for the *Irish* dataset, with the GenILoss value



(a) Adult (b) Irish

(c) Adult (d) Irish

(e) Adult (f) Irish

Figure 10: Data utility vs Varied k-value ($|QIDs|$=3).

increasing gradually, compared to the other algorithms. This is caused by the fact that, as k increases, less partitions are possible in the dataset. Thus, the QID values remain grouped in wider intervals. Even so, for this dataset (*Irish*), Mondrian shows the lowest information loss, as for this experimental configuration (with a varying k value and a $|QIDs|$=3), it was able to perform partitions across all attributes in the QID set.

Discernibility Metric (DM). In Figures 10(c) and 10(d), we can observe that the overall trend for the DM value is to increase for all three algorithms (much less so for *Adult* than for *Irish*). As the k value increases, more records are part of an EQ and thus, records are less distinguishable from each other. For the *Adult* dataset, the trends remain more steady, showing a low sensitivity to the growth of k value. Mondrian is the best performer, as the objective of this algorithm and its partitioning mechanism is to maximize the number of EQs (minimizing their size). In this experiment, as the k value increases, Mondrian creates less EQs but of a larger size; a situation clearly shown for the *Irish* dataset (i.e., the increasing trend). Intuitively, Incognito should be the second best performer as the implementation of this algorithm selects as optimal solution, the one that yields the maximum number of EQs. This is true in most of the cases, except for the *Adult* dataset when k=25 and 100. In this case, Datafly performs better than Incognito because, even though Incognito yielded more EQs than Datafly, one of the EQs is large, which incurred a significant impact to Incognito's DM score. For example, when k=25, Datafly creates 8 EQs, where the largest EQ has 9,362 records. Incognito creates 10 EQs but the largest EQ has 18,038 records; which results in a higher overall DM score.

Average Equivalence Class Size (C$_{AVG}$). Figures 10(e) and 10(f) present the results for data utility with respect to C$_{AVG}$ as the value of k increases. It is often the case that C$_{AVG}$ and DM show similar trends for the performance of the algorithms, as both metrics measure data utility based on the EQs created. However, in this experimental configuration (with a varying k value and a $|QIDs|$=3), they show different results. The decreasing trends that can be observed for the C$_{AVG}$ metric indicate that, as k increases, the average size of the created EQs gets closer to the ideal scenario, where the size of the EQs is equal to the given k. This ideal scenario happens when each record is generalized and grouped in an EQ composed of k records.

The results for the *Irish* dataset present lower values than the *Adult* one, this indicates that the EQs created for the Irish dataset are finer (smaller size). For example, consider Mondrian, which appears to be superior to the other algorithms. When k=2, Mondrian creates 58 EQs for *Adult* and 2,833 EQs for *Irish*. This means that the average size of the EQs for *Adult* is 520.03, whereas in *Irish* it is 10.58.

Summary. This second set of experiments offered some interesting findings. The increase in k has little or no impact on the efficiency of the algorithms (apart from Mondrian, as explained previously), given that a similar number of generalizations/number of nodes searched are performed. We can observe that the anonymization time decreases with the number of partitions. We can also observe that Mondrian outperforms the other algorithms with respect to the metrics that measure the size and number of the created EQs (e.g., DM and C$_{AVG}$). However, for the metrics that capture the transformation of the QID attributes (i.e., GenILoss), Mondrian may perform worse. This is because, when data is skewed, Mondrian cannot perform partitions across all attributes, resulting in high penalties.

6.3.3 Experiment 3: Varied dataset size

In this experiment, we analyze the scalability of the algorithms in terms of anonymization time and memory consumption, as the dataset size increases. We use the *Irish* dataset with different sizes. Given that this dataset is generated synthetically, we are able to use the same data distribution in our experiments.

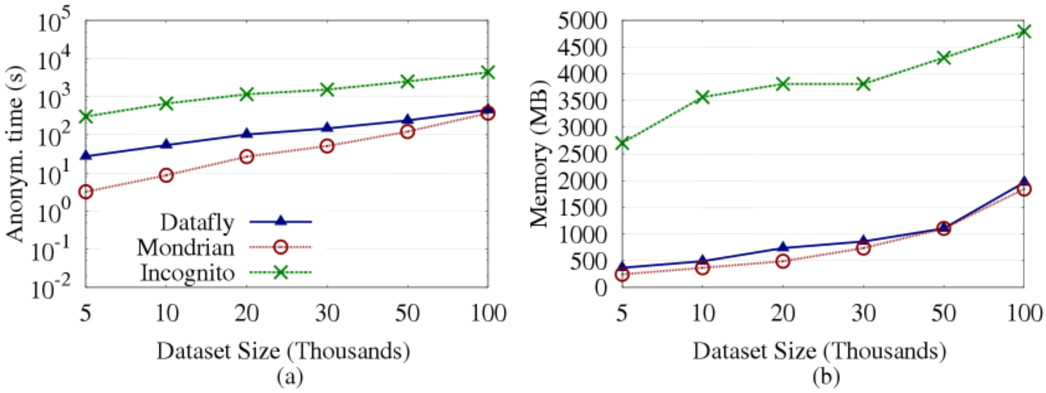


Figure 11: Performance vs Varied dataset size ($|QIDs|$=3, k=50).

Anonymization Time. Figure 11(a) presents the results for the anonymization time of the algorithms as the dataset size increases. It can be seen in the figures that the three algorithms follow a smooth growth trend. As the dataset size grows, the algorithms perform less generalizations/partitions which is expected to decrease the anonymization time. However, the increasing trend shown in the figures is due to the data volume itself; the time taken to anonymize the records. Mondrian is the best performer, with the largest dataset taking around 370.17 seconds (slightly above 6 minutes) to anonymize. The time complexity of Mondrian is O$(nlogn)$ [44, 17], where n is the number of tuples in the original table. For the same dataset, Datafly and Incognito take around 448 seconds (slightly above 7 minutes) and 4,344 sec (slightly above 72 minutes) respectively.

Memory Consumption. Figure 11(b) presents the comparison of the anonymization algorithms in terms of memory consumption as the dataset size increases. The three algorithms present an increasing trend which is mainly due to the growth of the volume of data. It should be noted that no Java MaGC occurred during the anonymization process that could have an impact in the algorithms' performance. Hence, the maximum memory consumption of each algorithm can be observed. Mondrian and Datafly present similar trends of increase. Incognito consumes more memory compared to the other algorithms due to the generalization lattice structure, which is maintained in main memory. The other two algorithms require memory for performing the frequency count to verify if k was satisfied (for Datafly), and calculating the median value for partitioning (for Mondrian).

Summary. In the third set of experiments, the best performers remained constant for the efficiency metrics across the different sizes of the dataset. This finding leads us to conclude that the performance of the algorithms are not affected by changes in their workload as long as the distribution of data is constant among workloads.

6.4 Discussion for Practitioners

Our results demonstrate how using numerous metrics in our comparative study allows for a wider understanding of the algorithms' performance in the data anonymization process.

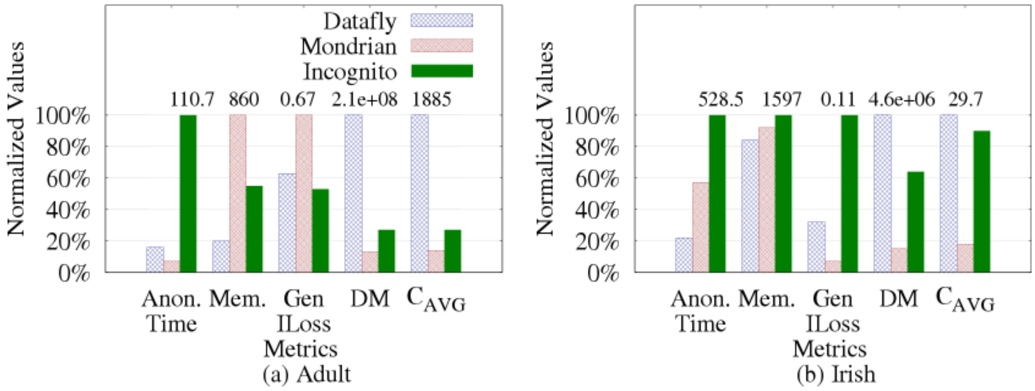


Figure 12: Performance comparison across all metrics ($|QIDs|$=3, k=2).

Based on an analysis of the results, we provide our key findings that could serve as guidelines to the data publishers to find the most suitable algorithm for their requirements (e.g., metric of interest).

- There are significant performance differences among the evaluated algorithms. Our evaluation showed that none of the algorithms outperformed the others across all the metrics (there is not "best for all" anonymization algorithm). Moreover, the best performing algorithm for one dataset was sometimes different for the other dataset. It was observed that both the privacy requirements and the data distribution, have a large impact on the performance of the algorithms. These differences are depicted in Figure 12, which shows the results across all the metrics used as evaluation criteria (efficiency and data utility), for a single experimental configuration (with $|QIDs|$=3 and k=2).

- Incognito was shown to be time consuming and memory intensive with respect to the size of the QIDs (i.e., presenting exponential growth). This time would be higher for deeper VGHs, as the search space is wider. This results from the time taken to traverse the lattice and check for k-anonymity in each individual state.

- Mondrian's data utility is significantly impacted by the data distribution and the mechanism used for partitioning. The median-partitioning approach is suitable for uniform distributions because the majority of the attributes can be used to partition the data. On the contrary, when data is skewed, the data cannot be partitioned across all attributes which impacts the information loss score. Thus, Mondrian performs poorly in this case. Furthermore, when anonymizing categorical values (which do not have a clear defined ordering), the order in which the values are processed to calculate the median plays an important role in the execution of the algorithm, as this may be the reason why the partitioning is not possible. Hence, this then impacts the data utility score. Mondrian terminates its execution once no more partitions are allowed. It is expected that this strategy benefits the data utility metrics which are based on the size of EQs (e.g., DM), as finer EQs will be created in the anonymization process. However, for datasets which originally satisfy k-anonymity, DM is not improved. This is because the number of EQs that are created may be lower than the number of EQs already present in the original dataset.

- Datafly was shown to be the second best performer in terms of anonymization time, memory consumption and GenILoss. For the metrics based on the size of EQs (i.e., DM and C$_{AVG}$), it performed the worst in almost all configurations.

- Regarding the strategies used by the algorithms: Datafly and Incognito use a hierarchy-based generalization; and Mondrian uses partition-based generalization. From a practitioner's perspective, the strategies matter in two aspects. Firstly, in the type of anonymization solution that the algorithms will produce. Secondly, the prerequisites that practitioners need to provide in order to apply the algorithms. In terms of the produced solution, Datafly and Incognito, which are based on user-defined generalization hierarchies (i.e., VGHs) will produce an anonymization which respects the constraints defined in the VGH. On the contrary, Mondrian is an uncontrolled anonymization, as the partitioning eliminates all hierarchical constraints, thus creating ranges of values dynamically. Based on this characteristic, Mondrian would be more suitable (and practical) for the anonymization of numerical datasets [5]. This is because in the case of categorical attributes, any semantics associated with the values can be compromised (e.g., to group together countries belonging to the same continent), as the data is partitioned without control over the groups created.

  In terms of the prerequisites of applicability, Datafly and Incognito require the specification of a VGH for each of the QID attributes. This task requires of knowledge of the domain in order to provide the adequate semantic background in the VGH. In the case of Mondrian, an ordering on each QID attribute needs to be imposed to perform the partitioning.

- Regarding the efficiency of the algorithms: (1) In terms of $|QIDs|$, Datafly and Mondrian are the best options when the number of QIDs is large. On the contrary, Incognito performs poorly for efficiency measures, especially when the generalization lattice is large and there is variability (distinct values) in the attributes. (2) In terms of k-anonymity level, Mondrian performed best as the required k increases. In general, the three algorithms handled well the increase in k, as there was not any major time increase for the algorithms. Mondrian even showed an improvement in time as the level of k increased. Regarding memory, the three algorithms showed a relatively stable consumption. (3) In terms of dataset size, Mondrian and Datafly performed better as the dataset size increased, exhibiting lower magnitudes of growth in anonymization time and memory consumption.

- Regarding the effectiveness of the algorithms: Mondrian was shown to be the best performer with respect to the metrics based on group size (i.e., DM and C$_{AVG}$). This means that Mondrian offers a finer granularity in the EQs, which is expected to improve the precision of the data results. An exception to this is when the original dataset already satisfies k-anonymity, as the data utility regarding these metrics may decrease. For the metric capturing the attributes' transformation (i.e., GenILoss), although there is not a clear best performer, Incognito and Mondrian showed good results. In particular, Mondrian would be better suited if the original data follows a uniform distribution.

Finally, we believe that to enable a systematic comparison of algorithms in PPDP, good metrics are those that can be applied to the majority of the algorithms without targeting a specific publishing scenario. Moreover, when evaluating different algorithms, a practitioner should also consider the features of the metric (strengths and weaknesses) that will be used for the evaluation in order to select the metric(s) of interest. Below, we discuss some of the features we identified for the general-purpose metrics used in our methodology:

- The DM metric does not capture the transformations performed on the QIDs. Thus, multiple anonymization solutions can have the same data utility with respect to DM, even though different QIDs have been anonymized in each solution. Also, the records in an EQ may not have been generalized, but even so, they will be penalized. However, it is expected that this type of metric accurately captures the data utility for certain applications like aggregate query answering [44].

- The C_{AVG} metric also suffers from the weakness previously mentioned for the DM metric. Additionally, it also fails to capture the granularity of the created EQs (i.e., the distribution of the records within each EQ).

- The GenILoss metric was shown to be sensitive to the VGH depth of the QIDs. A generalized attribute with a deeper VGH will incur in a lower penalty compared to the one incurred by other attributes that have a shallower VGH (e.g., *gender* attribute). This is an important aspect to consider as the inclusion of certain attributes may considerably degrade the data utility.

Even though these metrics suffer from above weaknesses, they can complement each other by capturing different aspects of the anonymized data.

7 Conclusions and Future Work

In this paper, we conducted a systematic performance evaluation, in terms of efficiency and data utility, of three of the most cited k-anonymization algorithms. Using publicly available implementations of the algorithms under a common framework (for a fair comparison), we identify the scenarios in which the algorithms performed well or poorly, in terms of the metric of interest and provide in-depth discussion of the reasons behind these behaviors. The results demonstrated that there is no best anonymization algorithm for all scenarios, but the best performing algorithm in a given situation is influenced by multiple factors. Based on our analysis, we provided insights about the factors to consider when selecting an anonymization algorithm and discussed the features (strengths and weaknesses) of a set of general-purpose utility metrics. Moreover, we motivate the importance of considering the needs of practitioners (who may not be data privacy experts), in order to offer guidelines that facilitate them with the adoption of privacy-preserving techniques. Additionally, these results provide evidence of the complexities of the selection process of anonymization algorithms, reflecting the necessity for creating methodologies that assist users in the process of identifying which anonymization algorithm is best suited for a particular publishing scenario.

In future work, we plan to investigate how best to use the identified behavior of the algorithms to create a metric (or a collection of metrics) that aims to standardize the comparison among algorithms, while also taking the user preferences into account. We also plan to explore how to automate the selection process of an anonymization algorithm by leveraging the derived knowledge with the ultimate goal of building a PPDP assisting tool to isolate the data publishers from the complexities of this process.

8 Acknowledgments

This work was supported, in part, by Science Foundation Ireland grant 10/CE/I1855 to Lero - the Irish Software Engineering Research Centre (www.lero.ie). This work has also received support from Science Foundation Ireland (SFI) via grant 08/SRC/I1403 FAME SRC (Federated, Autonomic Management of End-to-End Communications Services - Scientific Research Cluster).

References

[1] Central Statistics Office Databases. http://www.cso.ie/en/databases/.

[2] OpenData websites. http://www.data.gov/, http://data.gov.uk/.

[3] UTD Anonymization ToolBox. http://cs.utdallas.edu/dspl/cgi-bin/toolbox/.

[4] C. C. Aggarwal. On k-Anonymity and the Curse of Dimensionality. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 901–909. VLDB Endowment, 2005.

[5] M. R. S. Aghdam and N. Sonehara. On Enhancing Data Utility in k-Anonymization for Data without Hierarchical Taxonomies. *International Journal of Cyber-Security and Digital Forensics*, 2(2):12–22, 2013.

[6] D. Agrawal and C. C. Aggarwal. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '01, pages 247–255, 2001.

[7] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, and L. Murphy. Synthetic Data Generation using Benerator Tool. Technical report, University College Dublin, UCD-CSI-2013-03, 2013.

[8] K. S. Babu, N. Reddy, N. Kumar, M. Elliot, and S. K. Jena. Achieving k-anonymity Using Improved Greedy Heuristics for Very Large Relational Databases. *Transactions on Data Privacy*, 6(1):1–17, 2013.

[9] K. Bache and M. Lichman. UCI Machine Learning Repository, 2013.

[10] M. Barbaro and T. Zeller. A Face Is Exposed for AOL Searcher No . 4417749, 2006.

[11] R. J. Bayardo and R. Agrawal. Data Privacy Through Optimal k-Anonymization. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pages 217–228, 2005.

[12] V. Bergmann. Data Benerator Tool. http://databene.org/databene-benerator/.

[13] E. Bertino, I. N. Fovino, and L. P. Provenza. A Framework for Evaluating Privacy Preserving Data Mining Algorithms. *Data Mining and Knowledge Discovery*, 11(2):121–154, 2005.

[14] S. M. Blackburn, P. Cheng, and K. S. McKinley. Myths and Realities: The Performance Impact of Garbage Collection. *SIGMETRICS Performance Evaluation Review*, 32(1):25–36, 2004.

[15] J. Brickell and V. Shmatikov. The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 70–78, 2008.

[16] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala. Privacy-Preserving Data Publishing. *Foundations and Trends in Databases*, 2(1–2):1–167, 2009.

[17] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. *Secure Data Management In Decentralized Systems*, pages 323–353, 2007.

[18] C. Clifton and T. Tassa. On Syntactic Anonymity and Differential Privacy. *Transactions on Data Privacy*, 6(2):161–183, 2013.

[19] F. K. Dankar and K. El Emam. Practicing Differential Privacy in Health Care: A Review. *Transactions on Data Privacy*, 6(1):35–67, 2013.

[20] J. Domingo-Ferrer. A Survey of Inference Control Methods for Privacy-Preserving Data Mining. In C. C. Aggarwal and P. Yu, editors, *Privacy-Preserving Data Mining: Models and Algorithms*, volume 34 of *Advances in Database Systems*, pages 53–80. Springer US, 2008.

[21] J. Domingo-Ferrer, A. Martínez-Ballesté, J. M. Mateo-Sanz, and F. Sebé. Efficient Multivariate Data-Oriented Microaggregation. *The VLDB Journal*, 15(4):355–369, 2006.

[22] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical Data-Oriented Microaggregation for Statistical Disclosure Control. *IEEE Trans. on Knowl. and Data Eng.*, 14(1):189–201, 2002.

[23] J. Domingo-Ferrer, J. M. Mateo-Sanz, and V. Torra. Comparing SDC Methods for Microdata on the Basis of Information Loss and Disclosure. In *Proceedings of ETK-NTTS 2001, Luxemburg: Eurostat*, pages 807–826, 2001.

[24] J. Domingo-Ferrer and V. Torra. A Quantitative Comparison of Disclosure Control Methods for Microdata. *Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies*, pages 113–135, 2001.

[25] J. Domingo-Ferrer and V. Torra. Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation. *Data Min. Knowl. Discov.*, 11(2):195–212, 2005.

[26] C. Dwork. Differential Privacy. *Automata, Languages and Programming*, 4052:1–12, 2006.

[27] K. El Emam. Data Anonymization Practices in Clinical Research: A Descriptive Study. Technical report, Access to Information and Privacy Division of Health Canada, Ottawa, 2006.

[28] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley. A Globally Optimal k-Anonymity Method for the De-Identification of Health Data. *JAMIA*, 16(5):670–682, 2009.

[29] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy Preserving Mining of Association Rules. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 217–228, 2002.

[30] B. C. M. Fung, K. Wang, A. W.-C. Fu, and P. S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. Chapman & Hall/CRC, 1st edition, 2010.

[31] B. C. M. Fung, K. Wang, and P. S. Yu. Top-Down Specialization for Information and Privacy Preservation. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pages 205–216, 2005.

[32] J. Gantz and D. Reinsel. The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. Technical report, IDC, sponsored by EMC, 2012.

[33] D. Goodin. Poorly anonymized logs reveal NYC cab drivers' detailed whereabouts. http://arstechnica.com/tech-policy/2014/06/poorly-anonymized-logs-reveal-nyc-cab-drivers-detailed-whereabouts/.

[34] A. Hundepool, A. V. de Wetering, R. Ramaswamy, L. Franconi, A. Capobianchi, P. P. de Wolf, J. Domingo-Ferrer, V. Torra, R. Brand, and S. Giessing. μ-Argus User's Manual version 3.2, 2003.

[35] Information Commissioner's Office. Data Sharing Code of Practice. Technical report, ICO, 2011.

[36] V. S. Iyengar. Transforming Data to Satisfy Privacy Constraints. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 279–288, 2002.

[37] A. F. Karr, C. N. Kohnen, A. Oganian, J. P. Reiter, and A. P. Sanil. A Framework for Evaluating the Utility of Data Altered to Protect Confidentiality. *The American Statistician*, 60:224–232, 2006.

[38] B. Kenig and T. Tassa. A Practical Approximation Algorithm for Optimal k-Anonymity. *Data Min. Knowl. Discov.*, 25(1):134–168, 2012.

[39] D. Kifer and J. Gehrke. Injecting Utility into Anonymized Datasets. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, pages 217–228, 2006.

[40] S. Kisilevich, L. Rokach, Y. Elovici, and B. Shapira. Efficient Multidimensional Suppression for

K-Anonymity. *IEEE Trans. Knowl. Data Eng.*, 22(3):334–347, 2010.

[41] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. Flash: Efficient, Stable and Optimal K-Anonymity. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*, SOCIALCOM-PASSAT '12, pages 708–717, 2012.

[42] M. Laszlo and S. Mukherjee. Minimum Spanning Tree Partitioning Algorithm for Microaggregation. *IEEE Trans. on Knowl. and Data Eng.*, 17(7):902–911, 2005.

[43] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient Full-domain K-Anonymity. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 49–60, 2005.

[44] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian Multidimensional K-Anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*, ICDE '06, page 25, 2006.

[45] N. Li and T. Li. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Proceedings of the 23rd International Conference on Data Engineering*, ICDE '07, pages 106–115, 2007.

[46] J.-L. Lin, T.-H. Wen, J.-C. Hsieh, and P.-C. Chang. Density-based Microaggregation for Statistical Disclosure Control. *Expert Syst. Appl.*, 37(4):3256–3263, 2010.

[47] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-Diversity: Privacy Beyond k-Anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.

[48] S. Martínez, D. Sánchez, and A. Valls. Semantic Adaptive Microaggregation of Categorical Microdata. *Computers & Security*, 31(5):653–672, 2012.

[49] S. Morton, M. Mahoui, P. J. Gibson, and S. Yechuri. An Enhanced Utility-Driven Data Anonymization Method. *Transactions on Data Privacy*, 5(2):469–503, 2012.

[50] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, 2008.

[51] M. E. Nergiz and C. Clifton. Thoughts on k-Anonymization. *Data and Knowledge Engineering*, 63(3):622–645, 2007.

[52] A. Pinto. A Comparison of Anonymization Protection Principles. In *International Conference on Information Reuse and Integration*, pages 207–214, 2012.

[53] P. Samarati. Protecting Respondents' Identities in Microdata Release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010–1027, 2001.

[54] A. Solanas and A. Martínez-Ballesté. V-MDAV: A Multivariate Microaggregation With Variable Group Size. In *17th COMPSTAT Symposium of the IASC*, pages 917–925, 2006.

[55] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez. Improving the Utility of Differentially Private Data Releases via k-Anonymity. In *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, TRUSTCOM '13, pages 372–379, 2013.

[56] L. Sweeney. Achieving K-anonymity Privacy Protection Using Generalization and Suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

[57] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[58] T. Tassa, A. Mazza, and A. Gionis. k-Concealment: An Alternative Model of k-Type Anonymity. *Transactions on Data Privacy*, 5(1):189–222, 2012.

[59] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving Anonymization of Set-valued Data. *Proceedings of the VLDB Endowment*, 1(1):115–125, 2008.

[60] T. M. Truta, A. Campan, and P. Meyer. Generating Microdata with P-Sensitive K-Anonymity Property. In *Proceedings of the 4th VLDB Conference on Secure Data Management*, SDM'07, pages 124–141, 2007.

[61] C. Wang, L. Liu, and L. Gao. Research on K-Anonymity Algorithm in Privacy Protection. *Advanced Materials Research*, 756-759:3471–3475, 2013.

[62] K. Wang, P. S. Yu, and S. Chakraborty. Bottom-Up Generalization: A Data Mining Solution to Privacy Protection. In *Proceedings of the 4th IEEE International Conference on Data Mining*, ICDM '04, pages 249–256, 2004.

[63] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang. (α, k)-anonymity: An Enhanced k-Anonymity Model for Privacy-Preserving Data Publishing. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 754–759, 2006.

[64] X. Wu, X. Ying, K. Liu, and L. Chen. A Survey of Privacy-Preservation of Graphs and Social Networks. In C. C. Aggarwal and H. Wang, editors, *Managing and Mining Graph Data*, volume 40 of *Advances in Database Systems*, pages 421–453. Springer US, 2010.

[65] X. Xiao and Y. Tao. Anatomy: Simple and Effective Privacy Preservation. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, VLDB '06, pages 139–150. VLDB Endowment, 2006.

[66] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu. Utility-Based Anonymization for Privacy Preservation with Less Information Loss. *SIGKDD Explor. Newsl.*, 8(2):21–30, 2006.

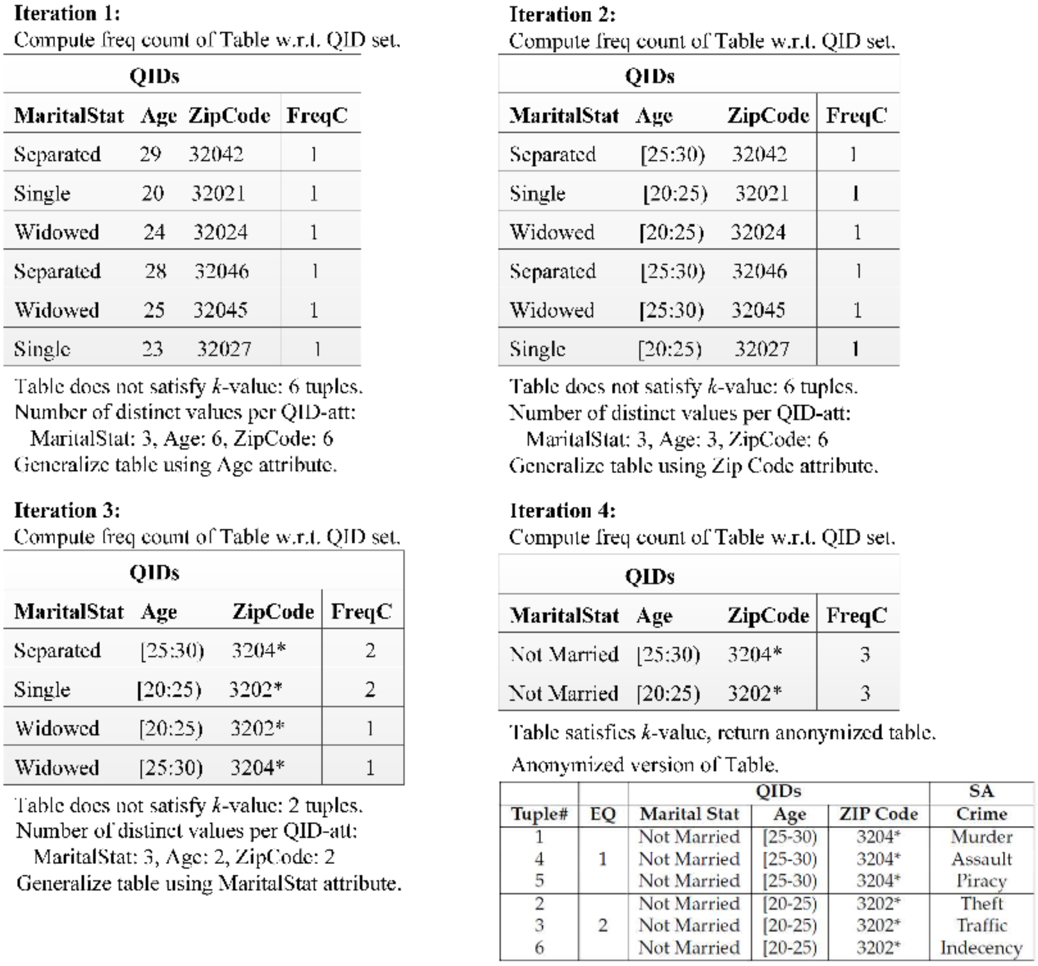
[67] T. Xu and Y. Cai. Feeling-based Location Privacy Protection for Location-based Services. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 348–357, 2009.

[68] Q. Zhang, N. Koudas, D. Srivastava, and T. Yu. Aggregate Query Answering on Anonymized Tables. In *Proceedings of the 23rd International Conference on Data Engineering*, ICDE '07, pages 116–125, 2007.

[69] G. Zhong and U. Hengartner. A Distributed k-Anonymity Protocol for Location Privacy. In *IEEE International Conference on Pervasive Computing and Communications*, PerCom '09, 2009.

Appendix A Datafly Example

In Figure 13 we provide an example of how the anonymization of Table 1 is carried out using the Datafly algorithm. We show the intermediary anonymizations performed to the data (i.e., iterations) and the final solution produced. The input parameters for this anonymization are: k=2, suppThreshold=0, QIDs={MaritalStat, Age, ZIP Code}. The generalizations are guided by the VGHs presented in Figure 2.



Iteration 1:
Compute freq count of Table w.r.t. QID set.

QIDs			
MaritalStat	**Age**	**ZipCode**	**FreqC**
Separated	29	32042	1
Single	20	32021	1
Widowed	24	32024	1
Separated	28	32046	1
Widowed	25	32045	1
Single	23	32027	1

Table does not satisfy k-value: 6 tuples.
Number of distinct values per QID-att:
MaritalStat: 3, Age: 6, ZipCode: 6
Generalize table using Age attribute.

Iteration 2:
Compute freq count of Table w.r.t. QID set.

QIDs			
MaritalStat	**Age**	**ZipCode**	**FreqC**
Separated	[25:30)	32042	1
Single	[20:25)	32021	1
Widowed	[20:25)	32024	1
Separated	[25:30)	32046	1
Widowed	[25:30)	32045	1
Single	[20:25)	32027	1

Table does not satisfy k-value: 6 tuples.
Number of distinct values per QID-att:
MaritalStat: 3, Age: 3, ZipCode: 6
Generalize table using Zip Code attribute.

Iteration 3:
Compute freq count of Table w.r.t. QID set.

QIDs			
MaritalStat	**Age**	**ZipCode**	**FreqC**
Separated	[25:30)	3204*	2
Single	[20:25)	3202*	2
Widowed	[20:25)	3202*	1
Widowed	[25:30)	3204*	1

Table does not satisfy k-value: 2 tuples.
Number of distinct values per QID-att:
MaritalStat: 3, Age: 2, ZipCode: 2
Generalize table using MaritalStat attribute.

Iteration 4:
Compute freq count of Table w.r.t. QID set.

QIDs			
MaritalStat	**Age**	**ZipCode**	**FreqC**
Not Married	[25:30)	3204*	3
Not Married	[20:25)	3202*	3

Table satisfies k-value, return anonymized table.
Anonymized version of Table.

		QIDs			SA
Tuple#	**EQ**	**Marital Stat**	**Age**	**ZIP Code**	**Crime**
1	1	Not Married	[25-30)	3204*	Murder
4		Not Married	[25-30)	3204*	Assault
5		Not Married	[25-30)	3204*	Piracy
2	2	Not Married	[20-25)	3202*	Theft
3		Not Married	[20-25)	3202*	Traffic
6		Not Married	[20-25)	3202*	Indecency

Figure 13: A schematic representation of the Datafly algorithm.

Appendix B Example of Generalization Lattice

This appendix provides the generalization lattice (in Figure 14) created for the QID attributes of Table 1: marital status (M), age (A) and ZIP code (Z). The available generalization states (i.e., nodes) are defined by the VGHs shown in Figure 2.

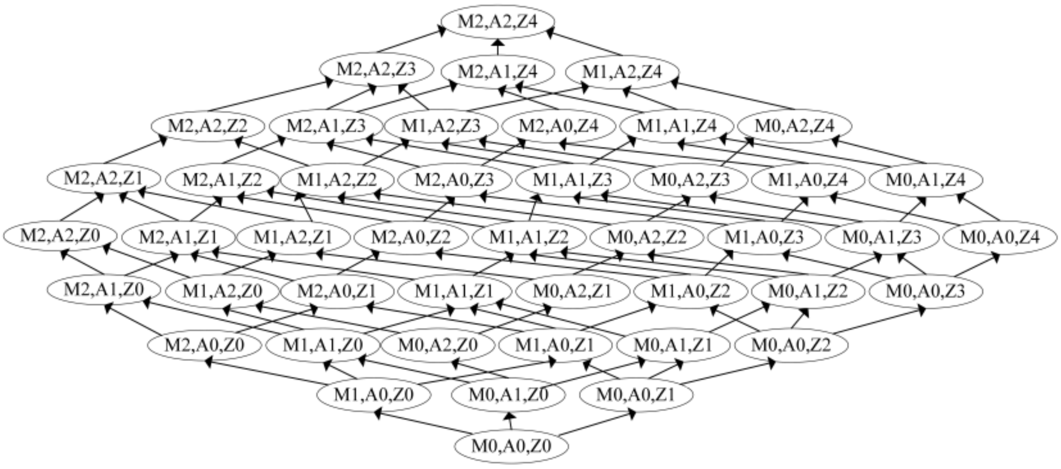
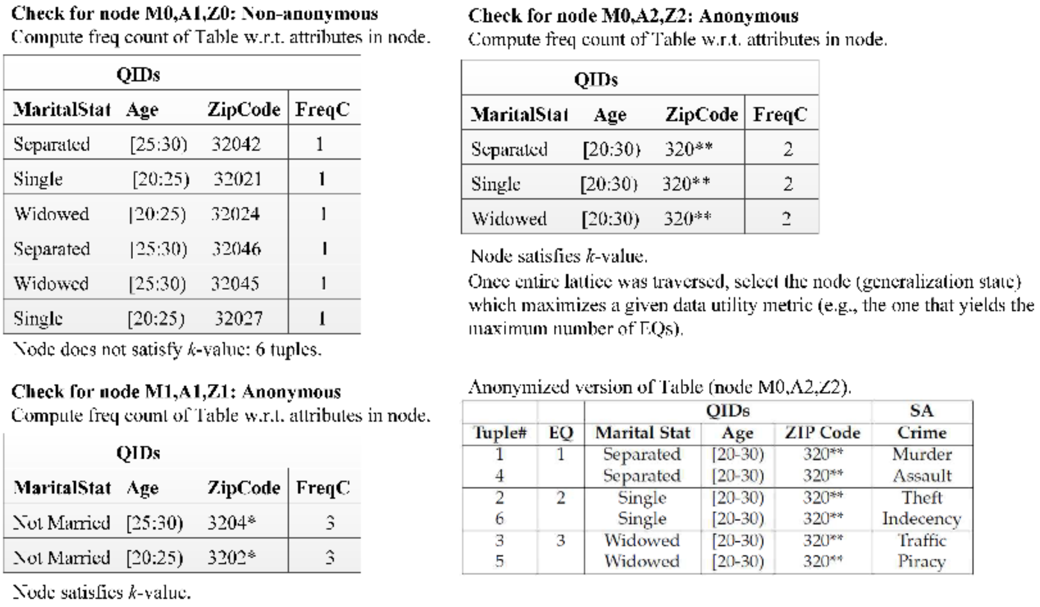


Figure 14: Generalization lattice for QIDs: marital status (M), age (A) and ZIP code (Z).

Appendix C Incognito Example

Figure 15 provides an example of how the anonymization of Table 1 is carried out using Incognito. The anonymization is guided by the generalization lattice presented in Appendix B. We show how some of the lattice nodes are checked for k-anonymity. We also present the final anonymization solution, which is selected among a collection of solutions that satisfy k (e.g., the one that maximizes the number of EQs). The input parameters for this anonymization are: k=2, suppThreshold=0, QIDs={MaritalStat, Age, ZIP Code}.



Check for node M0,A1,Z0: Non-anonymous
Compute freq count of Table w.r.t. attributes in node.

QIDs			
MaritalStat	Age	ZipCode	FreqC
Separated	[25:30)	32042	1
Single	[20:25)	32021	1
Widowed	[20:25)	32024	1
Separated	[25:30)	32046	1
Widowed	[25:30)	32045	1
Single	[20:25)	32027	1

Node does not satisfy k-value: 6 tuples.

Check for node M1,A1,Z1: Anonymous
Compute freq count of Table w.r.t. attributes in node.

QIDs			
MaritalStat	Age	ZipCode	FreqC
Not Married	[25:30)	3204*	3
Not Married	[20:25)	3202*	3

Node satisfies k-value.

Check for node M0,A2,Z2: Anonymous
Compute freq count of Table w.r.t. attributes in node.

QIDs			
MaritalStat	Age	ZipCode	FreqC
Separated	[20:30)	320**	2
Single	[20:30)	320**	2
Widowed	[20:30)	320**	2

Node satisfies k-value.
Once entire lattice was traversed, select the node (generalization state) which maximizes a given data utility metric (e.g., the one that yields the maximum number of EQs).

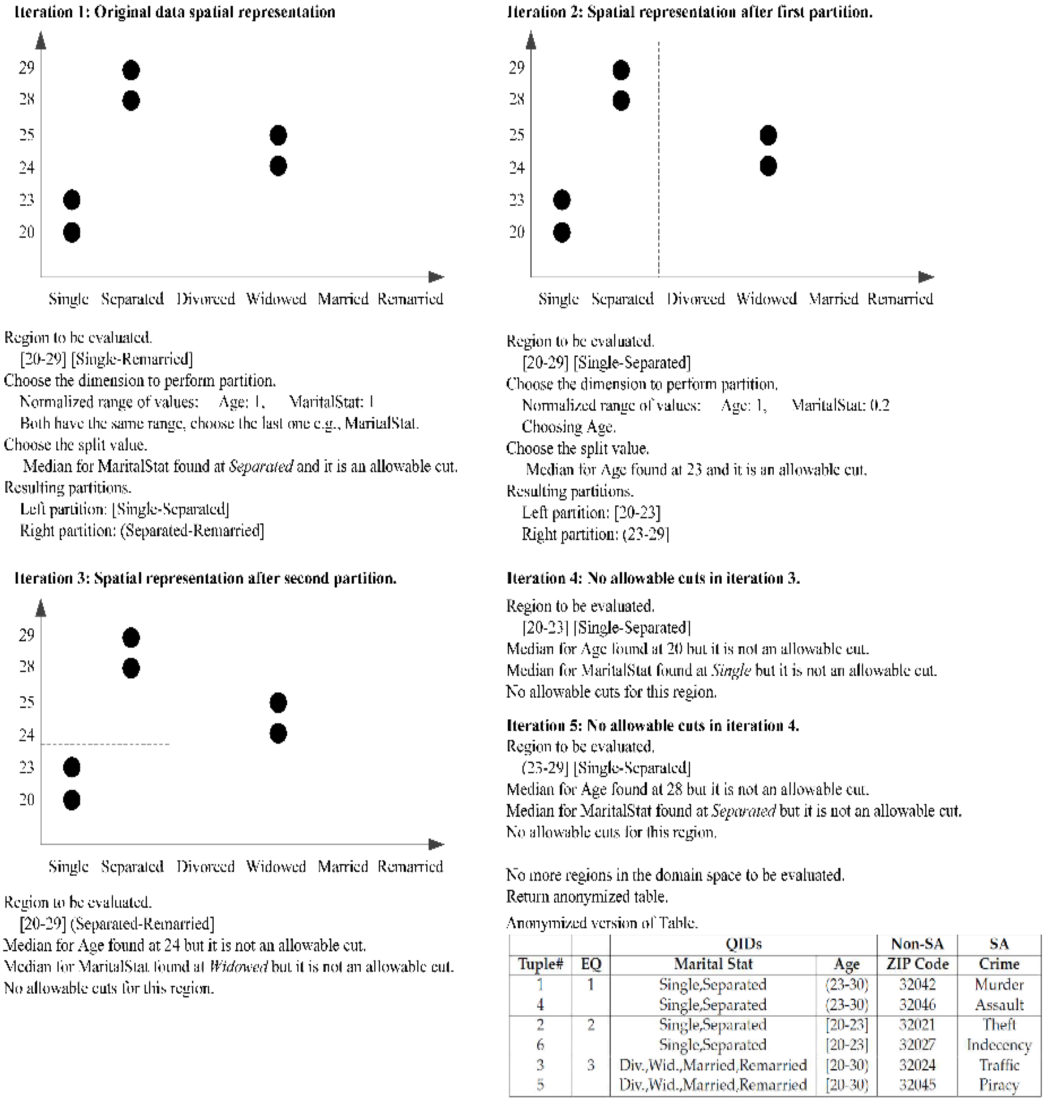
Anonymized version of Table (node M0,A2,Z2).

		QIDs			SA
Tuple#	EQ	Marital Stat	Age	ZIP Code	Crime
1	1	Separated	[20-30)	320**	Murder
4		Separated	[20-30)	320**	Assault
2	2	Single	[20-30)	320**	Theft
6		Single	[20-30)	320**	Indecency
3	3	Widowed	[20-30)	320**	Traffic
5		Widowed	[20-30)	320**	Piracy

Figure 15: A schematic representation of the Incognito algorithm.

Appendix D Mondrian Example

In Figure 16 we provide an example of how the anonymization of Table 1 is carried out using the Mondrian algorithm. In this example, we graphically show the two dimensional representation of QID values; the occurrence of a value is represented as a point. We also present some of the recursive partitions performed to the data and the final solution produced. For a clearer representation, we selected only 2 (out of 3) attributes as QIDs. The input parameters for this anonymization are: k=2, suppThreshold=0, QIDs={Age, Marital Status}. The heuristic employed in this example to know where perform the partitions is to choose the dimension with the widest (normalized) range of values. Similarly, the strategy to choose the split value is to perform median partitioning.



Iteration 1: Original data spatial representation

Region to be evaluated.
[20-29] [Single-Remarried]
Choose the dimension to perform partition.
Normalized range of values: Age: 1, MaritalStat: 1
Both have the same range, choose the last one e.g., MaritalStat.
Choose the split value.
Median for MaritalStat found at *Separated* and it is an allowable cut.
Resulting partitions.
Left partition: [Single-Separated]
Right partition: (Separated-Remarried]

Iteration 2: Spatial representation after first partition.

Region to be evaluated.
[20-29] [Single-Separated]
Choose the dimension to perform partition.
Normalized range of values: Age: 1, MaritalStat: 0.2
Choosing Age.
Choose the split value.
Median for Age found at 23 and it is an allowable cut.
Resulting partitions.
Left partition: [20-23]
Right partition: (23-29]

Iteration 3: Spatial representation after second partition.

Region to be evaluated.
[20-29] (Separated-Remarried]
Median for Age found at 24 but it is not an allowable cut.
Median for MaritalStat found at *Widowed* but it is not an allowable cut.
No allowable cuts for this region.

Iteration 4: No allowable cuts in iteration 3.

Region to be evaluated.
[20-23] [Single-Separated]
Median for Age found at 20 but it is not an allowable cut.
Median for MaritalStat found at *Single* but it is not an allowable cut.
No allowable cuts for this region.

Iteration 5: No allowable cuts in iteration 4.

Region to be evaluated.
(23-29] [Single-Separated]
Median for Age found at 28 but it is not an allowable cut.
Median for MaritalStat found at *Separated* but it is not an allowable cut.
No allowable cuts for this region.

No more regions in the domain space to be evaluated.
Return anonymized table.

Anonymized version of Table.

		QIDs		Non-SA	SA
Tuple#	EQ	Marital Stat	Age	ZIP Code	Crime
1	1	Single,Separated	(23-30)	32042	Murder
4		Single,Separated	(23-30)	32046	Assault
2	2	Single,Separated	[20-23]	32021	Theft
6		Single,Separated	[20-23]	32027	Indecency
3	3	Div.,Wid.,Married,Remarried	[20-30)	32024	Traffic
5		Div.,Wid.,Married,Remarried	[20-30)	32045	Piracy

Figure 16: A schematic representation of the Mondrian algorithm.