Abstracting Anonymization Techniques: A Prerequisite for Selecting a Generalization Algorithm

Feten Ben Fredj, Nadira Lammari, Isabelle Comyn-Wattiau

▶ **To cite this version:**

Feten Ben Fredj, Nadira Lammari, Isabelle Comyn-Wattiau. Abstracting Anonymization Techniques: A Prerequisite for Selecting a Generalization Algorithm. Knowledge-Based and Intelligent Information & Engineering Systems 19th Annual Conference, KES-2015, Sep 2015, Marina Bay Sands, Singapore. pp.206-215, 10.1016/j.procs.2015.08.120 . hal-03236910

HAL Id: hal-03236910
https://hal.science/hal-03236910v1

Submitted on 16 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 206 – 215

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information & Engineering Systems

Abstracting Anonymization Techniques: A Prerequisite for Selecting a Generalization Algorithm

Feten Ben Fredj[a,b], Nadira Lammari[a]*, Isabelle Comyn-Wattiau[a,c]

[a]*CEDRIC-CNAM, 292 rue Saint Martin, 75141, PARIS Cedex 03, France*
[b]*MIRACLE, Pôle technologique de Sfax, Route de Tunis Km 10 B.P. 242 SFAX 3021, Tunisie*
[c]*ESSEC Business School, 1 Av B. Hirsch, 95000 CERGY, France*

Abstract

The recent buzz around open data highlighted the crucial problem of anonymization in the context of data publishing. Many research efforts were devoted to the definition of techniques performing such an anonymization. However the selection of the most relevant technique and the adequate algorithm is complex. Successful decision depends firstly on the ability of data publishers to understand the anonymization techniques and their associated algorithms. In this paper, we focus on the choice of an algorithm among the different ones implementing one of the anonymization techniques, namely generalization. Through an abstraction process presented in this paper, we provide data publishers with simplified descriptions for the generalization technique and its algorithms. These descriptions facilitate the understanding of the algorithms by data publishers having low programing skills. We present also some other use cases of these abstractions as well as an experimentation conducted to validate them.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of KES International

Keywords: Anonymization; Privacy; Generalization Algorithm; Abstraction

1 Introduction

The open data initiative is an opportunity for making a huge amount of information accessible to end users. However, data publishers are facing the problem of releasing useful data without compromising privacy. The future

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: lammari@cnam.fr

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of KES International
doi:10.1016/j.procs.2015.08.120

of their business depends on their ability both to offer useful data to public, to analysts, to researchers, etc. and to gain the trust of data owners. The latter requires implementing processes that prevent the misuse of their sensitive information. Therefore, knowledge of anonymization techniques becomes necessary for data publishers. In particular, through this knowledge, they must choose the appropriate algorithm given their context.

Scrambling algorithms are very numerous. They are generally described thanks to their instantiation in a given context. Papers describing these algorithms focus on the experimentations leading to performance evaluation. Moreover, some surveys proposed in the literature are usage-oriented[1,2]. They mainly analyze different anonymization techniques highlighting their advantages and drawbacks in order to propose research directions. Others are technique-oriented[3,4]. Their comparisons of algorithms are mostly dedicated to researchers wishing to work on data anonymization and thus are not accessible to data publishers having generally low skills in data anonymization. Furthermore, existing tools are opaque [5]. Even if they propose several techniques, they implement, most of the time, only one algorithm per technique without describing it. Moreover, most of these tools do not provide guidance for the choice of algorithms or techniques. Let's note that SECRETA[6] proposes some guidance. However, this help consists in presenting an evaluation of the anonymized data set resulting from the user's original data set. If the user is unsatisfied, the tool offers him/her the possibility to adjust the input parameters of the algorithm. Therefore, its use requires a great amount of expertise. Finally, as far as we are aware, there exists neither knowledge bases where data publishers could seek the missing information nor approaches to guide them in the anonymization process. Our research questions are: How can a data publisher choose an anonymization technique and, in the set of algorithms implementing this technique, an anonymization algorithm? A first step toward answering these questions is to provide data publishers with simplified descriptions of the techniques and the algorithms, allowing them to understand them, even without the required programming skills. We are convinced that this step is necessary for decision making. These simplified descriptions are obtained through an abstraction process consisting in extracting and formalizing knowledge embedded in the literature in order to make it available through information resources. Facing the richness of the literature, in this paper, we concentrate our effort on the generalization technique for relational databases.

This paper presents some of our simplified descriptions and the process which helped us getting them. The remainder is organized as follows. Section 2 presents the generalization technique and a brief comparison of nine generalization algorithms. Section 3 details our abstraction process and its different outputs. It also describes an experimentation conducted to validate our results. Section 4 concludes the paper by describing some use cases illustrating how our contributions can be used and by sketching future research.

2 Preliminaries

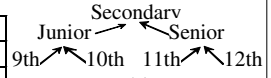
Privacy is one of the major concerns when publishing or sharing data. It refers to different forms of disclosure regarding the type of published or shared content. Identity disclosure, for instance, can occur when publishing or sharing personnel data. In our research, we focus on microdata (atomic data elements describing the individual objects) contained in relational databases. Each tuple has a value (microdata) for each relational attribute. The latter can be an explicit identifier, a quasi-identifier, a sensitive attribute or a non-sensitive one. An explicit identifier (EI) directly identifies an individual (e.g. social security number).

Explicit Identifier	Quasi Identifier		Sensitive attribute
Name	Age	Education	Disease
Alice	19	10th	Diabetes
Jean	19	9th	Cancer
Ines	27	9th	Flu
David	30	9th	Flu
Bob	23	11th	Cancer
Dupont	23	11th	Cancer

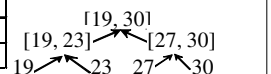
(a)

Age	Education	Disease
[19,23]	Junior	Diabetes
[19,23]	Junior	Cancer
[27,30]	Junior	Flu
[27,30]	Junior	Flu
[19,23]	Senior	Cancer
[19,23]	Senior	Cancer

(b)



(c)



(d)

Fig. 1. (a) Original data; (b) generalized data ; (c) & (d) generalization hierarchies

A quasi-identifier (QI) is an attribute set which, when linked to external information, enables the re-identification of individuals whose identifiers were removed ({sex, zip code, and birthdate} is a well-known quasi-identifier in many data sets). A sensitive attribute (SA) represents data that individuals don't want to divulgate, such as medical information. Non-sensitive attributes (NSA) are all attributes that are not included in previous categories. For instance, at Fig. 1a representing the original data set to be anonymized, the attributes "Age" and "Education" may constitute a QI. The attribute "Disease" is a sensitive attribute (SA).

Companies may implement specific techniques to protect their data from disclosure risk. Most of them are known as privacy preserving data publishing (PPDP) or mining (PPDM) techniques[1,2,3]. To our knowledge, the most familiar techniques for microdata anonymization are: data swapping[7], adding noise[8], micro-aggregation[9] and generalization[10]. In this paper, we focus on the generalization technique and its algorithms. The generalization is applied on a QI. Each QI attribute could be either continuous or categorical. A continuous attribute is numerical and may take an infinite number of different real values (e.g. "Age" in Fig. 1a). A categorical attribute takes a value in a limited set and arithmetic operations on it do not make sense (e.g. "Education" in Fig. 1a). Generalization technique requires the definition of a hierarchy for each attribute of the QI. Each hierarchy contains at least two levels. The root is the most general value. It represents the highest level. The leaves correspond to the original data values and constitute the lowest level. As an example, the tree at Fig. 1c represents a generalization hierarchy of the attribute "Education". The node "Junior" is at the level 1 of the Education hierarchy. To avoid possible re-identification of individuals, several privacy models have been proposed: k-anonymity, l-diversity, t-closeness, etc. In this paper, we place special emphasis on k-anonymity since all generalization algorithms are based on it. Let k be an integer. A table satisfies k-anonymity if each release of data is such that every combination of values of QI can be indistinctly matched to at least k individuals[11]. As an example, the table in Fig. 1b is a generalization of the original table of Fig. 1a satisfying 2-anonymity, regarding (Age, Education) QI.

The generalization technique is implemented thanks to several different algorithms. The best known are: μ-argus[12], Datafly[13], Samarati's algorithm[10], Incognito[14], Bottom Up Generalization[15], Top Down Specialization[16], Median Mondrian[17], Infogain Mondrian and LSD Mondrian[18]. In our previous work[19], we have compared these algorithms in terms of process models, having four main constituents: pre-requisites, inputs, process logic, and outputs. Regarding the inputs, all generalization algorithms require at least: (i) to set the value of k (corresponding to k-anonymity), (ii) to declare which columns constitute the QI, and (iii) to provide the generalization hierarchies. From a process point of view, we can notice that some algorithms are completely automatic. Moreover, some of them are bottom up processes[14] i.e. small groups of tuples are constituted and then iteratively merged until each group contains at least k rows (k-anonymity satisfaction)[20]. Others are top down processes[16] i.e. they start from a group containing all rows and iteratively split each group into two subgroups while preserving k-anonymity. Regarding the outputs, some algorithms compute an optimal k-anonymity solution but they are limited to small data sets[20]. Others are based on heuristics and thus do not guarantee the optimality. Finally, the algorithms perform three different generalizations that we define as: full-domain, sub-tree and multidimensional generalization. Full-domain means that, for a given QI column, all the values in the output table belong to the same level of the generalization hierarchy. Sub-tree means that values sharing the same direct parent in the hierarchy are necessarily generalized at the same level, taking the value of one of their common ancestors. Finally, in multidimensional generalizations, two identical values in the original table may lead to different generalized values (i.e. are not always generalized at the same level). In terms of usage scenario, let us note that bottom up generalization, top down specialization and InfoGain Mondrian produce data for classification tasks. LSD Mondrian is used when regression must be performed on data.

3 Abstraction approach

We aim at providing data publishers with a deep knowledge of generalization algorithms' behavior. Hence, we performed an abstraction process of all these algorithms, allowing us to map them into a common frame. As highlighted by Wing[21]: "the abstraction process introduces layers. In computer science, we work simultaneously with at least two, usually more, layers of abstraction: the layer of interest and the layer below; or the layer of interest and the layer above". In our case, using an example artefact of the algorithm (layer below), we defined an abstraction of the algorithm artefact (layer of interest). Moreover, in software and information systems engineering,

the motivations for abstraction are manifold. In our research, two main reasons strengthen our choices. First, through this abstraction process, we aim to describe the different steps of the generalization algorithms as simply as possible to facilitate their appropriation and adoption. Second, we wish to highlight the requirements of each algorithm in order to introduce them as meta-data. Taking into account these two motivations, we built an abstraction by parameterization. As defined by Navrat et al.[22], "abstraction by parameterization extracts an essential core of some computational elements and reifies them as a named element of its own, leaving parameters to be filled in when the abstraction is instantiated". Moreover, in order to reach an overall understanding of these algorithms, we conducted an inductive process, also called generalization in Navrat et al.[22], that presents all these algorithms using a common description. These two sub-processes are described in the following paragraphs.

3.1. Abstraction by parametrization of the generalization algorithms

In most papers, the generalization algorithms are presented in such a way that they can be directly translated into a program. Usually, they are usually partially instantiated using an example of a table to be anonymized. Their basic principles are textually described. Therefore they are dedicated to computer scientists or to professionals having a programming background. To produce a more abstract description of the generalization algorithms, we have filtered away all irrelevant information and sometimes added information in order to facilitate the extraction of content. Since an algorithm is a dynamic artefact, we chose to represent it via a flowchart. The latter is quite helpful in understanding the logic of complex problems. For lack of space, we present the results of our abstraction process for only three algorithms: Datafly, Top Down specialization (TDS) and Median Mondrian algorithms. Datafly was the first algorithm able to meet the k-anonymity requirement for a big set of real data. It combines generalization of data and suppression of tuples in order to avoid an excessive generalization which would reduce data usefulness. At each iteration, DataFly (a) generalizes the attributes having the highest number of distinct values, (b) and checks whether the resulting table complies with the k-anonymity. If the number of tuples which do not satisfy k-anonymity is equal or lower than k, then these tuples are removed and the algorithm stops. Otherwise, the algorithm performs another iteration of generalization. In the description of Fig. 2a, let PT represent the table to be anonymized, k the k-anonymity constraint threshold, DGH_{Ai} the generalization hierarchy of the attribute Ai, and MGT the resulting anonymized table.

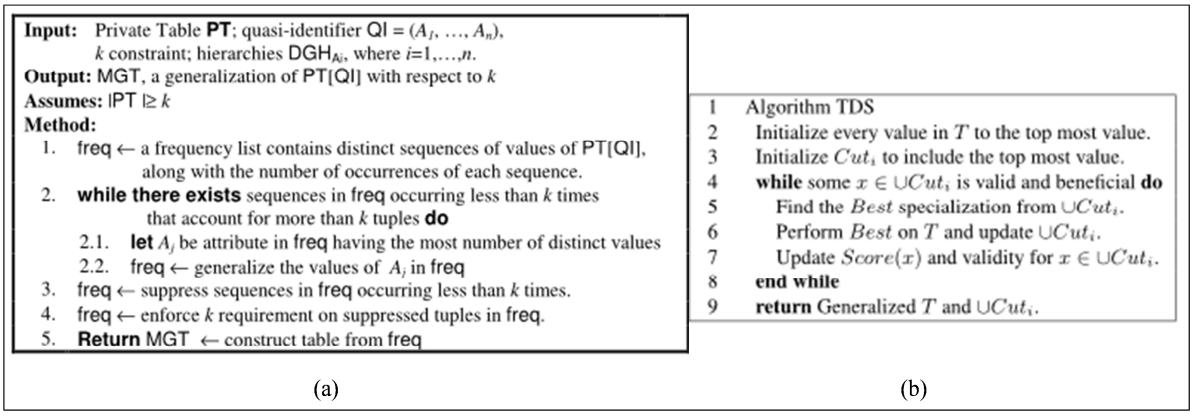


Fig. 2. (a) Datafly Algorithm[13]; (b) Top-Down Specialization Algorithm (TDS)[16]

Fig. 2(a) focuses more on the implementation of Datafly than on its functioning principle. The abstract presentation at Fig. 3a highlights the basic principle and therefore facilitates the understanding. In this abstraction, Datafly generalizes one attribute at a time. At each iteration, it selects (as mentioned in step 2) the one that has the best score (the highest number of distinct values in the current table). The execution stops when the k-anonymity is reached.

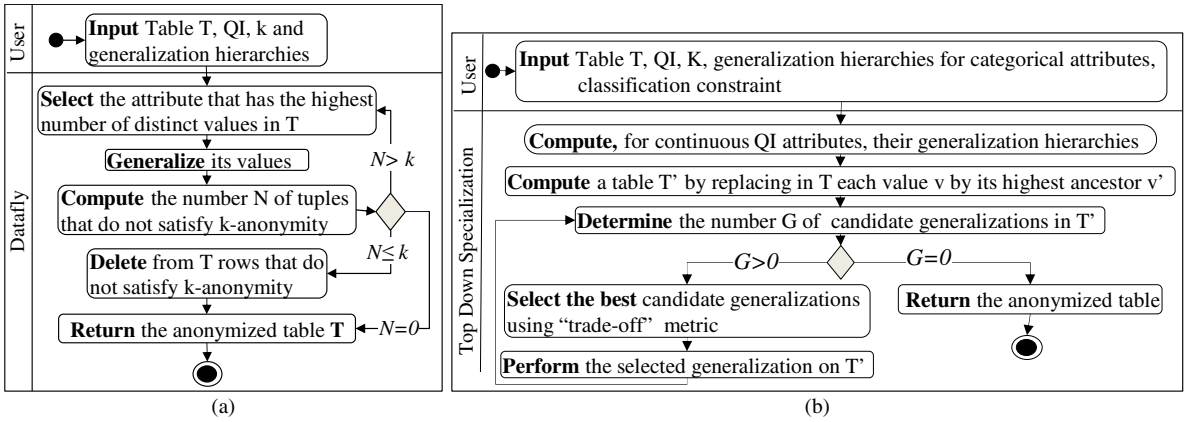
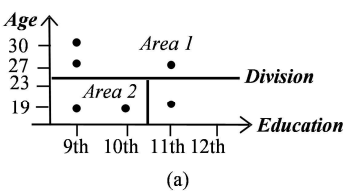


Fig. 3. (a) Abstraction of Datafly algorithm; (b) Abstraction of TDS algorithm

The second example describes the TDS algorithm extracted from Fung et al.[16] (Fig. 2b). It browses the generalization hierarchy from top to bottom. A high generalization of all the values of the original table preserves k-anonymity but can negatively impact the quality of the resulting table in terms of classification. Therefore TDS performs iterations to find the best specializations i.e. those that not only satisfy k-anonymity but also generate less anonymity loss, thus enabling better classification quality. Intuitively, in this algorithm, ∪CUTi represents all the candidate specializations (the process is a top down process in the sense that it browses the generalization hierarchies from top to down). At the initialization step (line 3) ∪CUTi is a single set, denoted CUTi in the algorithm. Valid and beneficial specializations (denoted x) are specializations that do not violate the k-anonymity and that respect the classification constraint (TDS is dedicated to data mining usage, and especially to classification algorithms). Best specializations are valid and beneficial specializations that, moreover, reach the best score (denoted Score (x) in the algorithm) in terms of security and quality. This score is computed using the trade-off metric[20]. The abstraction process of this algorithm leads to the model sketched at Fig. 3b.

Through this abstraction we exhibit the fact that TDS starts from a table T' which represents a high level of generalization, giving priority to security at the expense of quality. Then in order to find the best compromise between quality and security (the trade-off metric serves this purpose), TDS tries to get closer to the actual values of T.

The last example of generalization algorithm is Median Mondrian[17]. Its principle is to divide the set of individuals (tuples) represented in the table into groups such that each group contains at least k individuals. Then, the individuals of the same group will have the same value for their QI via the generalization process. More precisely, individuals (tuples of the original table) are represented, thanks to the values of their QI, in a multidimensional space where each dimension corresponds to an attribute of the QI (Fig. 4a). The splitting of the space into areas generates the groups. It is performed using the median value of the attribute.




```
Anonymize (partition)
    1. If (no allowable multidimensional cut for partition)
    2. then return Φ : partition → summary
    3. Else
      3.1 dim ← choose_dimension()
      3.2 fs ← frequency_set(partition, dim)
      3.3 splitVal ← find_median(fs)
      3.4 lhs ← {t ∈ partition : t.dim ≤ splitVal}
      3.5 rhs ← {t ∈ partition : t.dim > splitVal}
      3.6 return Anonymize (rhs) ∪ Anonymize (lhs)
```

(b)

Fig. 4. Principle of Median Mondrian and its algorithm[17]

At each iteration, the algorithm chooses a dimension and checks the possibility of splitting a group into two groups (i.e. splitting the area on the median value of this dimension). A group can be divided into two groups if each resulting group contains at least k individuals. If the division is not possible, the corresponding group is marked. The splitting process switches to another dimension when all groups are marked for the current dimension. It stops when all dimensions have been explored. Then the algorithm performs the possible generalizations, replacing the different values in the same area with the value of their first common parent in the generalization hierarchy (recoding process). As shown at Fig. 4b, the algorithm is recursive. At the first iteration, "partition" contains all the tuples of the table to anonymize. The function "choose_dimension()" (resp. "frequency_set(partition, dim)") returns the chosen dimension (resp. the set «fs» of values taken by a given dimension "dim" in a given partition "partition"). The function find_median(fs) returns the value "splitVal" of the median. "t.dim" is the value of a given dimension "dim" for the tuple "t". The summary consists of the generalization of a set of values belonging to the same partition. It is defined by a value range where the lower limit (resp. upper limit) corresponds to the smallest value (resp. to the largest value) in the partition. Our abstraction of Median Mondrian leads to the activity diagram at Fig. 5.

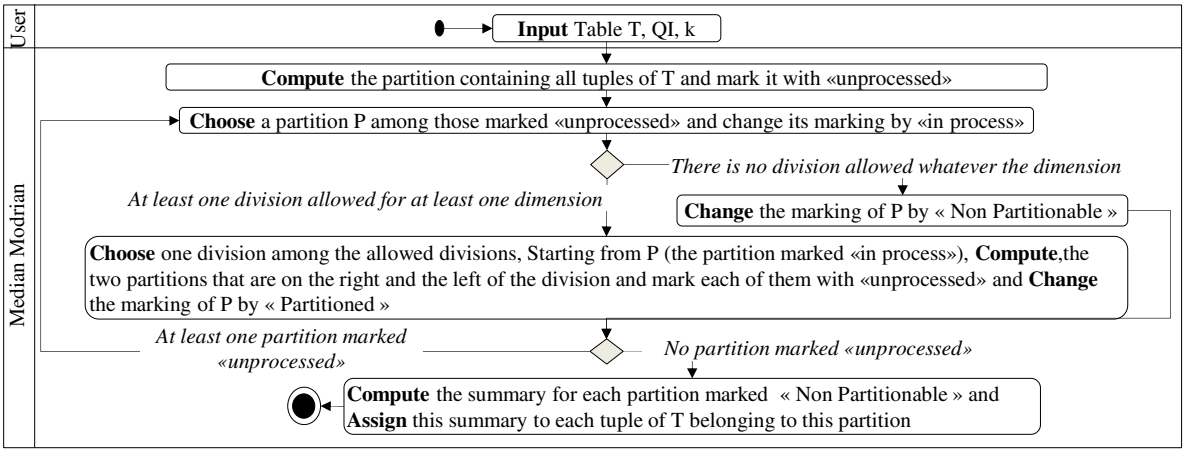


Fig. 5. Abstraction of Median Mondrian

In order to validate our contribution, we conducted an experiment. The objective was to evaluate the understandability of our abstraction by comparing it with those found in the literature. We performed the comparison for two algorithms, namely Datafly and Median Mondrian. A total of 12 participants were recruited. They were all either post-graduate students or researchers in computer science. Therefore, all of them were familiar with algorithmic and programming techniques but not aware of anonymization algorithms. To avoid any biased interpretation of the results, we have provided the participants with the same types of representations: our abstractions have been transformed into textual representation (algorithms).

The experiment lasted about four hours. First the participants had to fill a questionnaire about their level of knowledge in anonymization techniques (they had to evaluate their level using a scale of 1 to 10) and their programming skills (they had to say if they have ancient, recent or very recent programming skills). Second, all participants were given a brief presentation on anonymization with emphasis on the generalization technique. Then they received a copy of the slides and sheets of papers for taking down notes. Third, they were divided into two homogeneous groups based on their programming profiles. We have defined three profiles (1 for ancient, 2 for recent, and 3 for very recent). Then we provided all the participants with a small table to anonymize and asked them to execute manually three algorithms mentioned in their sheet without time limit. For the first group of participants we proposed, successively, our abstraction of Datafly, then the algorithm of Figure 2a and, finally, the Median Mondrian of Figure 4b. The second group had to execute our abstraction of Median Mondrian, followed by the Median Mondrian of Figure 4b and Datafly of Figure 2a. We attached to Median Mondrian and Datafly the explanation proposed by their authors. Whenever a participant met a problem to complete the execution of an

algorithm, he (or she) was invited to indicate on his/her sheet the reason for blockage. In the last phase of the experiment process, the participants had to answer the following questions:

- Was the initial presentation sufficient for being able to execute the algorithms? If your answer is no, please explain which information was missing.
- Did you detect similarities between some of the algorithms you had to execute?
- Did you detect identical algorithms (even if differently described)?
- Do you think that one algorithm helped you in understanding another one?

The participants had also to assign to each algorithm a level of difficulty on a scale from 1 to 10.

Table 1. Synthesis of data collected from the experiment

	Datafly of Fig. 3a			Datafly of Fig. 2a			Median Mondrian of Fig. 5			Median Mondrian of Fig. 4b		
Class	Erroneous	Correct	Partial	Erroneous	Correct	Partial	Erroneous	Correct	Partial	Erroneous	Correct	Partial
Profile 1	20%	0%	0%	20%	0%	0%	20%	20%	0%	20%	0%	10%
Profile 2	0%	40%	0%	0%	20%	20%	0%	20%	0%	0%	20%	10%
Profile 3	0%	40%	0%	0%	20%	20%	0%	40%	0%	0%	40%	0%

Following the experiment, we started analyzing the collected information. 12 participants executed three algorithms each. We rejected three illegible executions out of the 36. For each algorithm, the legible executions have been grouped into three classes. The first (resp. the second one) gathered all the correct executions (resp. all the partial executions). The last class contained all the erroneous executions. For the partial executions we have deduced from the comments of the participants three reasons for blocking. The first one is related to the interpretation of the while instruction of the original Datafly (Fig. 2a). The second one was the disability to understand the data structure ”freq” in the same algorithm. Finally, the third reason for blocking was the double recursion of the original Median Mondrian (Fig. 4b). Table 1 summarizes the results. For each algorithm and each profile, the percentages of erroneous, correct and partially correct executions are mentioned.

All the participants who have executed our abstractions didn’t face blocking difficulties. Moreover, only those that had ancient knowledge obtained erroneous executions and they are few. Only those who first performed the execution of Datafly abstraction have then proposed a correct execution of the original Datafly. The same observation emerged for Median Mondrian. They unanimously mentioned in the post questionnaire that our abstraction helped them understanding the original algorithms. Moreover they have ranked the original algorithms as more difficult to understand than our abstractions. All the participants that met a blockage in the original Datafly (40%) have not executed our abstraction. It is the same for the original Median Mondrian. Finally, over the 20% who delivered an erroneous execution for the original Median Mondrian, 50% had not used our abstraction.

To sum up, this analysis revealed that the lack of intelligibility of the original algorithms impacts all participants whatever their programming skills. Threats to validity must be mentioned due to the limited size of our experimentation group and the restriction of the study to only two algorithms. However, this first analysis encourages us to persevere in this abstraction effort. We wanted to check if our algorithm representation was easier to understand than the classical one. Therefore, our pool of testers was composed of persons with programming skills. Thus, they were able to understand both representations. We evaluated both perceived usefulness and objective usefulness. Perceived usefulness was captured through direct queries regarding how they could understand the underlying logic of each algorithm. Objective usefulness was measured through the correctness of the result obtained by participants. Thus, if users with programming skills prefer better perform our abstraction, how much more data publishers with less programming skills will be at ease with it.

3.2. Generalization process of the abstracted algorithms.

The nine algorithms reviewed in our research are all based on generalization techniques. Our abstraction process led us to the identification of three categories using the parameterization and the generalization process described above. This abstraction process followed a bottom-up logic and a one-to-many mapping. Abstracting each algorithm

allowed us eliciting similarities and grouping algorithms into categories. Each category is also associated to an abstract representation. Hence we defined a one-to-many mapping between this representation and the different algorithms belonging to this category.

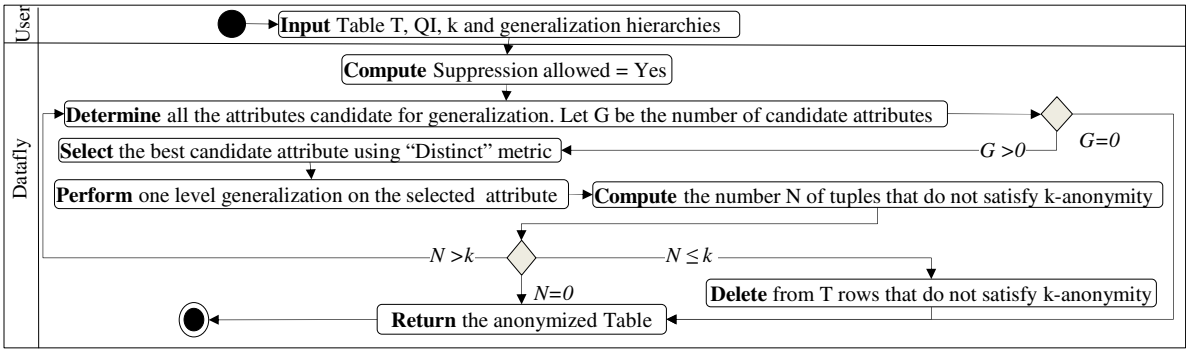


Fig. 6. Homogenized abstraction of Datafly algorithm

We have first grouped the algorithms into categories regarding their basic principles and their type of resulting generalization. The first category called MR Recoding[1] groups together the algorithms generating multidimensional generalizations (Median Mondrian, InfoGain Mondrian and LSD Mondrian). Their basic specificity is to consider together all the attributes of the QI. They iteratively divide the set of tuples into groups such that each group satisfies k-anonymity. Conversely, in the two other categories (LR Recoding and TR Recoding), the generalizations are not multidimensional. Moreover, at each iteration, they only deal with one attribute of the QI. LR Recoding gathers algorithms based on a lattice structure representing all the possible generalizations of the original table. Samarati and Incognito algorithms belong to this category. Finally, TR Recoding includes Datafly, μ-argus, bottom-up generalization and top down specialization algorithms. Their specificity is to build directly and iteratively an anonymized table. In order to provide each category with an abstraction, we had to homogenize the nine abstractions. For instance, we had to transform TR Recoding abstractions since some algorithms of this category are top down processes and others are bottom up, some of them include local or global suppressions and others only perform generalizations. Thus, we have introduced a parameter allowing or disallowing suppressions. For example, Datafly allows suppressions (Fig. 6).

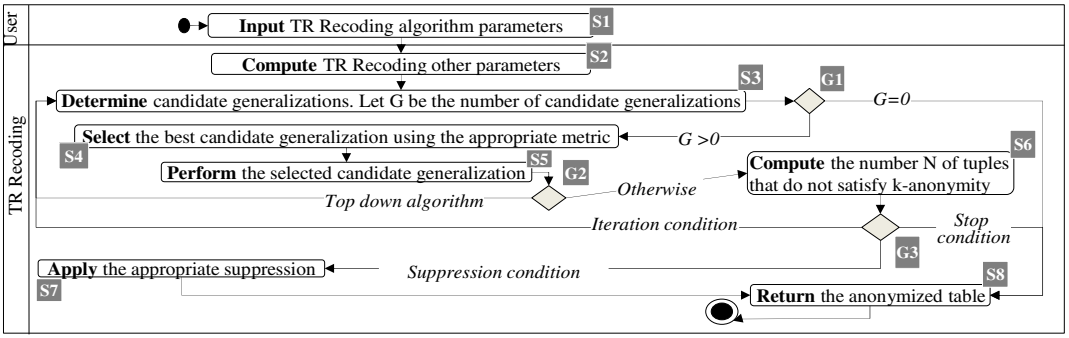
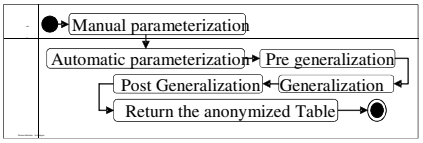


Fig. 7. Abstraction of TR Recoding algorithms

We also rephrased some instructions by defining new concepts in order to reach a unique abstraction for each category of algorithms and thus to allow parameterization. For instance, in order to homogenize the concepts

1 The term recoding is often used in papers to define the transformation of data in tables required by the generalization

"candidate generalization" and "best candidate generalization" with Datafly concepts, we introduced in the latter the concepts of "candidate attribute" and "best candidate attribute". As another example of standardization, since each algorithm has its own metric to select the best generalization, we introduced a parameter called "appropriate metric" in the abstraction of TR Recoding (Fig. 7). This parameter takes the value "Distinct metric" for DataFly and "Trade-off metric" in TDS. Thus, the abstraction of Fig. 7 may be instantiated into the four algorithms thanks to a correct parameterization. The same process allowed us to obtain an abstraction for all nine algorithms and for the three recoding categories. For space reasons, we cannot provide the reader with all the abstractions. Following this bottom-up logic, we have performed the same abstraction effort to homogenize the three types of recoding. We generated the abstraction of Fig. 8a which in fact represents the generalization technique. Figure 8b instantiates this process for Recoding TR. Step numbers refer to Fig. 7. Thus our recurring abstraction process resulted in a taxonomy of generalization algorithms (Fig. 9). We have defined an abstraction by parameterization using flowcharts for all the nodes of this taxonomy.



(a)

Parameterization components	Instantiation for TR recoding
Manual parameterization	*step 1*
Automatic parameterization	*step 2*
Pre generalization	*step 3 and 4*
Generalization	*step 5*
Post generalization	*step 6 and 7*

(b)

Fig. 8. (a) Abstraction of the generalization technique; (b) An instantiation of the generalization technique

Thus our recurring abstraction process resulted in a taxonomy of generalization algorithms (Fig. 8). We have built an abstraction by parameterization using flowcharts for all the nodes of this taxonomy.

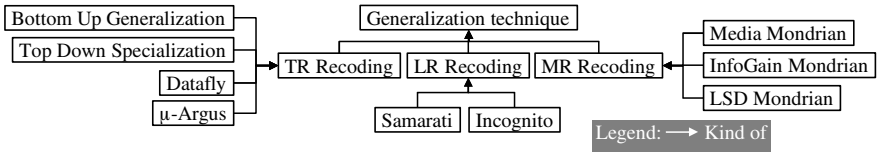


Fig. 9. A taxonomy of generalization algorithms

4 Conclusion

An extensive attention has been paid to privacy protection by statistics and computer science communities over these past years. A large body of research deals with anonymization techniques and algorithms. Our research is not devoted to new algorithms or new techniques. We want to help data publishers with low programming skills in understanding the existing techniques and algorithms. This help is made possible thanks to simplified representations associated to each technique and algorithm. In this paper, we focused on the generalization technique and on nine of its algorithms, since it is one of the most used techniques for tabular data. The simplified representations have been obtained through an abstraction process described in the paper. The abstraction effort allowed us to detect similar behaviors between the nine algorithms and thus to define three main categories of generalization algorithms. We also defined an abstract model for each category and finally an abstraction of the generalization technique. Finally, thanks to our categorization, we built a taxonomy of these generalization algorithms, helping a novice to understand how all these anonymization algorithms may be differentiated. For space reasons, the paper only contains some abstraction models. We conducted a controlled experiment, leading to encouraging results since participants found that the proposed abstractions were very useful to understand the algorithms whatever their programming skills. These abstractions constitute a first step towards the design of patterns that will be part of a catalog. The latter will be made available through a guidance approach we plan to design. A pattern documents either a technique or an algorithm through its intent, its context of use, its inputs, and its process with an illustrative example. The first three constituents of the pattern are extracted from our previous comparative study[19]. The process is described both through the existing algorithm and through our abstraction.

Another use case of our abstractions lies in the context of e-learning. Our abstractions may serve as a basis for the development of tutorials whose purpose is to assist data stewards, students and researchers in learning how to use anonymization algorithms and then to provide them with a well-informed usage of existing anonymization tools. To ensure a better transfer of knowledge, these tutorials could be contextualized according to the level of expertise of their potential users. Moreover, we are convinced that our taxonomy can be a starting point for the design of an ontology of anonymization techniques and algorithms. This ontology will exhibit conflicts between techniques and then will contribute to a definition of combined anonymization scenarios for a whole database. Finally, we believe that the availability of such knowledge should facilitate the adoption of new anonymization techniques and minimize the loss of competencies that arise when an employee leaves the company.

There are several avenues for future work. First, we want to perform the same effort of abstraction for other algorithms based on swapping technique for example. We aim to check if the process also leads to a useful abstraction. Second, we want to reach a higher level in the taxonomy by abstracting the different techniques (generalization, shuffling, swapping, etc.). Finally, we also expect to develop an ontology of techniques and algorithms. This ontology could be the main component for a decision support system helping a data publisher in selecting the suitable technique and algorithm given a context.

References

1. Fung, B. C. M., Wang, K., Chen, R., Yu, P. S.: Privacy preserving data publishing: a survey of recent developments. In ACM Computing Surveys (CSUR), Vol. 42(14) (2010)
2. Ilavarasi. B., Sathiyabhama A. K., Poorani. S.,.A survey on privacy preserving data mining techniques. In Int. Journal of Computer Science and Business Informatics, 7(1), (2013)
3. Xu, X., Ma, T., Tang, M.,Tian, W.: A survey of privacy preserving data publishing using generalization and suppression. In Int. Journal on Applied Mathematics & Information Sciences, Vol 8(3) pp 1103-1116 (2014)
4. Patel, L., Gupta, R.: A Survey of Perturbation Technique for Privacy-Preserving of Data. In Int. Journal of Emerging Technology and Advanced Engineering, Vol 3(6) (2013)
5. Vinogradov, S., Pastsyak, A.: Evaluation of Data Anonymization Tools. The 4th Int. Conference on Advances in Databases, Knowledge, and Data Applications DBKDA (2012)
6. Poulis G., Gkoulalas-Divanis A., Loukides G., Skiadopoulos S., Tryfonopoulos C.:SECRETA: A System for Evaluating and Comparing RElational and Transaction Anonymization algorithms. EDBT 2014.
7. Fienberg S, McIntyre J.: Data Swapping: Variations on a Theme by Dalenius and Reiss. J. Domingo-Ferrer and V. Torra (Eds.): PSD 2004, LNCS 3050, pp. 14–29, Springer (2004)
8. Brand R.: Microdata protection through noise addition. Inference Control in Statistical Databases. LNCS 2316, pp 97-116, Springer (2002)
9. Defays D., Nanopoulos P.: Panels of enterprises and confidentiality: the small aggregates method. In Proc. 92nd Symposium on Design and Analysis of Longitudinal Surveys. (1993)
10. Samarati, P.: Protecting respondents' identities in microdata release. IEEE Trans. on Knowledge and Data Engineering, Vol 13(6), (2001)
11. Sweeney, L.: k-Anonymity: A model for protecting privacy. In Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (5), pp 557-570 (2002)
12. Undepool, A., Willenborg, L.: μ- and τ-argus: Software for statistical disclosure control. 3rd Int. Seminar on Statistical Confidentiality (1996)
13. Sweeney, L.: Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), pp 571-588 (2002)
14. Lefevre, K., Dewitt, D. J., Ramakrichnon, R.: Incognito: Efficient full-domain k-anonymity. ACM Int. Conf. on Management of Data (2005)
15. Wang, K., Yu, P. S., Chakraborty, S.: Bottom-up generalization: A data mining solution to privacy protection.4th IEEE Int. Conf. on Data Mining (2004)
16. Fung, B. C. M., Wang, K., Yu, P. S.: Top-down specialization for information and privacy preservation. In 21st IEEE Int. Conf. on Data Engineering (ICDE) pp 205–216 (2005)
17. Lefevre, K., Dewitt, D. J., Ramakrichnon, R.: Mondrian multidimensional k-anonymity. In 22nd IEEE Int. Conference on Data Engineering (ICDE) (2006)
18. Lefevre, K., Dewitt, D. J., Ramakrichnon, R.: Workload-aware anonymization. In 12th ACM SIGKDD (2006)
19. Ben Fredj F., Lammari, N., Comyn-Wattiau, I.: Characterizing Generalization Algorithms- First Guidelines for Data Publishers. In 6th International Conference on Knowledge Management and Information Sharing (KMIS) (2014)
20. Benjamin, C. M., Fung, Wang, K., Chen, R., Yu, P. S.: Privacy-Preserving Data Publishing: A Survey on Recent Developments. ACM Computing Surveys, Vol. 42(4) (2010)
21. Wing, J. M.: Computational Thinking and Thinking About Computing. Philosophical Transactions of the Royal Society, vol. 366, pp. 3717-3725 (2008)
22. Návrat, P., Filkorn, R.: A Note on the Role of Abstraction and Generality in Software Development. Journal of Computer Science, Vol 1(1), pp 98-102 (2005)