International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

IJIEMR Transactions, online available on 20th Jan 2021. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=ISSUE-01

DOI: 10.48047/IJIEMR/V10/I01/32

Title: **Privacy preservation for Abstracting Anonymization Techniques using Generalization Algorithm**

Volume 10, Issue 01, Pages: 157-161.

Paper Authors

B Md Irfan, Dr. K.Bhavana Raj, Deepu J.J.Lazarus

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Privacy preservation for Abstracting Anonymization Techniques using Generalization Algorithm

[1]B Md Irfan, [2]Dr. K.Bhavana Raj, [3]Deepu J.J.Lazarus
[1]Research Scholar, Department of Computer Science & Technology, Sri Krishnadevaraya University, Anantapuramu, A.P. India.
[2]Assistant Professor, Institute of Public Enterprise, Shamirpet, Hyderabad.
[3]Research Scholar, Department of Computer Science & Technology, Sri Krishnadevaraya University, Anantapuramu, A.P. India.
bhavana_raj_83@yahoo.com

Abstract— The ongoing improvements around open information featured the significant issue of anonymization with regards to information distributing. Many research endeavors were given to the meaning of strategies performing such an anonymization. Anyway the determination of the most applicable method and the satisfactory calculation is perplexing. Effective choice relies right off the bat upon the capacity of information distributers to comprehend the anonymization systems and their related calculations. In this paper, we center around the decision of a calculation among the various ones executing one of the anonymization systems, to be specific speculation. Through a reflection procedure displayed in this paper, we give information distributers improved depictions for the speculation system and its calculations. These depictions encourage the comprehension of the calculations by information distributers having low programming aptitudes. We present additionally some other use instances of these deliberations just as an experimentation directed to approve them.

Keywords : Anonymization, Programming aptitude.

INTRODUCTION

The open information activity is an open door for making an immense measure of data available to end clients. In any case, information distributers are confronting the issue of discharging helpful information without trading off protection. The eventual fate of their business relies upon their capacity both to offer helpful information to open, to examiners, to scientists, and so on and to pick up the trust of information proprietors. The last requires executing forms that forestall the abuse of their delicate data. Thusly, information on anonymization methods gets essential for information distributers. In specific, through this information, they should pick the suitable calculation given their unique circumstance.

Scrambling calculations are various. They are commonly portrayed gratitude to their launch in a given setting. Papers portraying these calculations center around the experimentations prompting execution assessment. In addition, a few reviews proposed in the writing are utilization arranged. They for the most part break down various anonymization procedures featuring their focal points and downsides so as to propose explore headings. Others are method situated. Their correlations of calculations are for the most part devoted to scientists wishing to take a shot at information anonymization and in this manner are not open to information distributers having commonly low aptitudes in information anonymization. Moreover, existing devices are obscure. Regardless of whether they propose a few procedures, they execute, more often than not, just a single calculation for every strategy without portraying it. Additionally, the vast majority of these apparatuses don't give direction to the selection of calculations or procedures.

Notwithstanding, this assistance comprises in exhibiting an assessment of the anonymized informational collection coming about because of the client's unique informational collection. On the off chance that the client is unsatisfied, the instrument offers him/her the likelihood to change the information parameters of the calculation. In this way, its utilization requires a lot of aptitude. At long last, the extent that we know, there exists neither one of the knowledges bases where information distributers could look for the missing data nor ways to deal with direct them in the anonymization procedure. Our exploration questions are: How can an information distributer pick an anonymization system and, in the arrangement of calculations actualizing this procedure, an anonymization calculation? An initial move toward responding to these inquiries is to furnish information distributers with improved depictions of the methods and the calculations, permitting them to get them, even without the necessary programming abilities. We are persuaded that this progression is important for basic leadership. These disentangled depictions are acquired through a reflection procedure comprising in removing and formalizing information inserted in the writing so as to make it accessible through data assets. Confronting the wealth of the writing, in this paper, we focus our exertion on the speculation system for relational databases.

2 Preliminaries

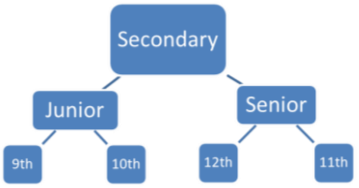
Privacy is one of the major concerns when publishing or sharing data. It refers to different forms of disclosureregarding the type of published or shared content. Identity disclosure, for instance, can occur when publishing orsharing personnel data. In our research, we focus on microdata (atomic data elements describing the individualobjects) contained in relational databases. Each tuple has a value (microdata) for each relational attribute. The lattercan be an explicit identifier, a quasi-identifier, a sensitive attribute or a non-sensitive one. An explicit identifier (EI) A quasi-identifier (QI) is an attribute set which, when linked to external information, enables the re-identificationof individuals whose identifiers were removed ({sex, zip code, and birthdate} is a well-known quasi-identifier inmany data sets). A sensitive attribute (SA) represents data that individuals don't want to divulgate, such as medicalinformation. Non-sensitive attributes (NSA) are all attributes that are not included in previous categories. Forinstance, at Fig. 1a representing the original data set to be anonymized, the attributes "Age" and "Education" mayconstitute a QI. The attribute "Disease" is a sensitive attribute (SA).

Explicit identifier	Quasi Identifier		Sensitive attribute
Name	Age	Education	Disease
Ajay	20	10th	Flu
Chand	20	9th	Flu
Ram	28	9th	Diabetes
Krishna	30	9th	Cancer
Sudhir	23	11th	Cancer
Basha	23	11th	Cancer

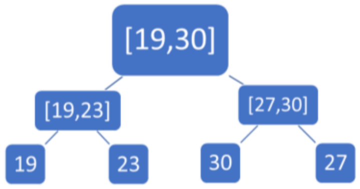
(a)

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

Companies may implement specific techniques to protect their data from disclosure risk. Most of them are knownas privacy preserving data publishing (PPDP) or mining (PPDM) techniques. To our knowledge, the mostfamiliar techniques for microdata anonymization are: data swapping,adding noise, micro-aggregationandgeneralization. In this paper, we focus on the generalization technique and its algorithms. The generalization isapplied on a QI. Each QI attribute could be either continuous or categorical. A continuous attribute is numerical andmay take an infinite number of different real values (e.g. "Age" in Fig. 1a). A categorical attribute takes a value in alimited set and arithmetic operations on it do not make sense (e.g. "Education" in Fig. 1a). Generalization techniquerequires the definition of a hierarchy for each attribute of the QI. Each hierarchy contains at least two levels. Theroot is the most general value. It represents the highest level. The leaves correspond to the original data values andconstitute 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 ofindividuals, several privacy models have been proposed: k-anonymity, l-diversity, t-closeness, etc. In this paper, weplace special emphasis on k-anonymity since all generalization algorithms are based on it. Let k be an integer. Atable satisfies k-anonymity if each release of data is such that every combination of values of QI can be indistinctlymatched to at least k individuals11. 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:-argus, Datafly, Samarati's algorithm, Incognito, Bottom Up Generalization, Top Down Specialization,Median Mondrian, Infogain Mondrian and LSD Mondrian. In our previous work, we have compared thesealgorithms in terms of process models, having four main constituents: pre-requisites, inputs, process logic, andoutputs. Regarding the inputs, all generalization algorithms require at least: (i) to set the value of k (corresponding tok-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 ofthem are bottom up processes i.e. small groups of tuples are constituted and then iteratively merged until eachgroup contains at least k rows (k-anonymity satisfaction). Others are top down processesi.e. they start from agroup 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 datasets[20]. Others are based on heuristics and thus do not guarantee the optimality. Finally, the algorithms perform threedifferent generalizations that we define as: full-domain, sub-tree and multidimensional generalization. Full-domainmeans that, for a given QI column, all the values in the output table belong to the same level of the generalizationhierarchy. Sub-tree means that values sharing the same direct parent in the hierarchy are necessarily generalized atthe same level, taking the value of one of their common ancestors. Finally, in multidimensional generalizations, twoidentical values in the original table may lead to different generalized values (i.e. are not always generalized at thesame level). In terms of usage scenario, let us note that bottom up generalization, top down specialization andInfoGain Mondrian produce data for classification tasks. LSD Mondrian is used when regression must be performedon data.

3 Abstraction approach

We aim at providing data publishers with a deep knowledge of generalization algorithms' behavior. Hence, weperformed an abstraction process of all these algorithms, allowing us to map them into a common frame. Ashighlighted by Wing: "The abstraction process introduces layers. In computer science, we work simultaneouslywith at least two, usually more, layers of abstraction: the layer of interest and the layer below; or the layer of interestand the layer above". In our case, using an example artifact of the algorithm (layer below), we defined anabstraction of the algorithm artifact (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, throughthis abstraction process, we aim to describe the different steps of the generalization algorithms as simply as possibleto facilitate their appropriation and adoption. Second, we wish to highlight the requirements of each algorithm inorder to introduce them as meta-data. Taking into account these two motivations, we built an abstraction byparameterization. As defined by Navrat et al., “Abstraction by parameterization extracts an essential core of somecomputational elements and reifies them as a named element of its own, leaving parameters to be filled in when theabstraction is instantiated”. Moreover, in order to reach an overall understanding of these algorithms, we conductedan inductive process, also called generalization in Navrat et al that presents all these algorithms using a commondescription. 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 intoa program. Usually, they are usually partially instantiated using an example of a table to be anonymized. Their basicprinciples are textually described. Therefore they are dedicated to computer scientists or to professionals having aprogramming background. To produce a more abstract description of the generalization algorithms, we have filteredaway all irrelevant information and sometimes added information in order to facilitate the extraction of content.Since an algorithm is a dynamic artifact, we chose to represent it via a flowchart. The latter is quite helpful inunderstanding the logic of complex problems. For lack of space, we present the results of our abstraction process foronly three algorithms: Datafly, Top Down specialization (TDS) and Median Mondrian algorithms. Datafly was thefirst algorithm able to meet the k-anonymity requirement for a big set of real data. It combines generalization of dataand suppression of tuples in order to avoid an excessive generalization which would reduce data usefulness. At eachiteration, DataFly (a) generalizes the attributes having the highest number of distinct values, (b) and checks whetherthe resulting table complies with the k-anonymity. If the number of tuples which do not satisfy k-anonymity is equalor lower than k, then these tuples are removed and the algorithm stops. Otherwise, the algorithm performs anotheriteration of generalization. In the description of Fig. 2a, let PT represent the table to be anonymized, k the kanonymityconstraint threshold, DGHAi the generalization hierarchy of the attribute Ai, and MGT the resulting Fig. 2(a) focuses more on the implementation of Datafly than on its functioning principle.

```
INPUT: Private Table $PT$; quasi-identifier $QI = (Ai, ….., An)$, $k$ constraint; hierarchies $DGH_{Ai}$, where
       $i = 1, … , n$.

Output: $MGT$, a generalization of $PT[QI]$ with respect to $k$

Assumes: $|PT| \geq k$

Method:

  1.   freq ← a frequency list contains distinct sequences of values of PT[QI], along with the number
            of occurrences of each sequence.
  2.   While there exists sequences in freq occurring less than $k$ times that account for more than
            $k$ tuples do
  2.1.  Let $A_i$ be attribute in freq having the most number of distinct values
  2.2.  freq ← generalize the values of $A_i$ in freq
  3.   freq ← suppress sequences in freq occurring less than $k$ times.
  4.   freq ← enforce $k$ requirement on suppressed tuples in freq
  5.   Return MGT ← construct table from freq
```

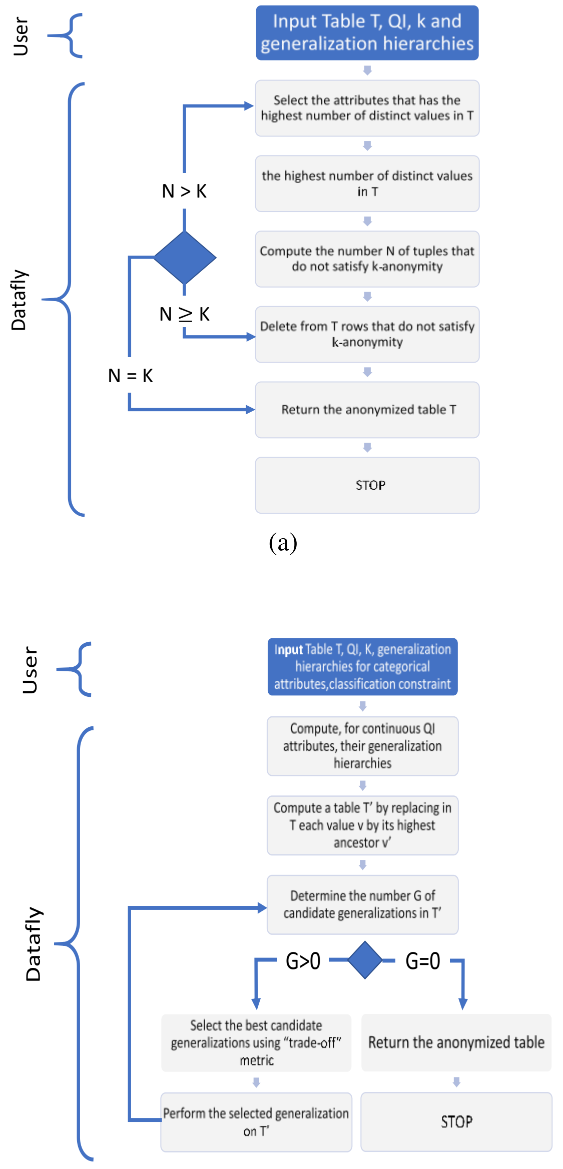
(a)

```
1.  Algorithm TDS
2.  Initialize every value in T to the top most value.
3.  Initialize $Cut_i$to include the top most value.
4.  while some $x \in \cup Cut_i$is valid and beneficial do
5.     Find the Best specialization from $\cup Cut_i$.
6.     Perform Best on T and update $\cup Cut_i$.
7.     Update $Score(x)$ and validity for $x \in \cup Cut_i$.
8.  end while
9.  return Generalized T and $\cup Cut_i$.
```

(b)

Fig. 2. (a) Datafly Algorithm13; (b) Top-Down Specialization Algorithm (TDS)

The abstractpresentation 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 thebest score (the highest number of distinct values in the current table). The execution stops when the k-anonymity isreached. The second example describes the TDS algorithm extracted from Fung et al.16 (Fig. 2b). It browses thegeneralization hierarchy


User
Datafly
Input Table T, QI, k and generalization hierarchies
Select the attributes that has the highest number of distinct values in T
the highest number of distinct values in T
Compute the number N of tuples that do not satisfy k-anonymity
Delete from T rows that do not satisfy k-anonymity
Return the anonymized table T
STOP
N > K
N ≥ K
N = K
User
Datafly
Input Table T, QI, K, generalization hierarchies for categorical attributes,classification constraint
Compute, for continuous QI attributes, their generalization hierarchies
Compute a table T' by replacing in T each value v by its highest ancestor v'
Determine the number G of candidate generalizations in T'
G>0
G=0
Select the best candidate generalizations using "trade-off" metric
Return the anonymized table
Perform the selected generalization on T'
STOP

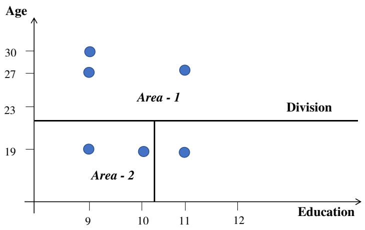
(a)

(b)

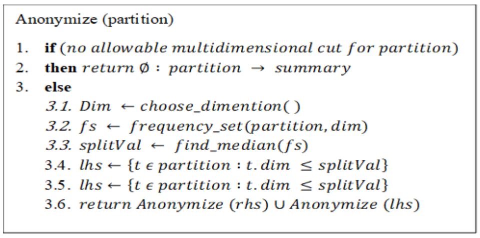
Fig. 3. (a) Abstraction of Datafly algorithm; (b) Abstraction of TDS algorithm

from top to bottom. A high generalization of all the values of the original table preserves k-anonymitybut can negatively impact the quality of the resulting table in terms of classification. Therefore TDSperforms iterations to find the best specializations i.e. those that not only satisfy k-anonymity but also generate lessanonymity 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 generalizationhierarchies from top to down). At the initialization step (line 3) ∪CUTi is a single set, denoted CUTi in thealgorithm. Valid and beneficial specializations (denoted x) are specializations that do not violate the k-anonymityand that respect the classification constraint (TDS is dedicated to data mining usage, and especially to classificationalgorithms). 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-offmetric20. 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 ofgeneralization, giving priority to security at the expense of quality. Then in order to find the best compromisebetween quality and security (the trade-off metric serves this purpose), TDS tries to get closer to the actual values ofT.

The last example of generalization algorithm is Median Mondrian[17]. Its principle is to divide the set ofindividuals (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. Moreprecisely, individuals (tuples of the original table) are represented, thanks to the values of their QI, in amultidimensional space where each dimension corresponds to an attribute of the QI (Fig. 4a). The splitting of thespace into areas generates the groups. It is performed using the median value of the attribute.



(a)



```
Anonymize (partition)
if ($no\ allowable\ multidimensional\ cut\ for\ partition$)
then $return\ \emptyset : partition \rightarrow summary$
else
    $Dim \leftarrow choose\_dimention(\,)$
    $fs \leftarrow frequency\_set(partition, dim)$
    $splitVal \leftarrow find\_median(fs)$
    $lhs \leftarrow \{t \in partition : t.dim \leq splitVal\}$
    $lhs \leftarrow \{t \in partition : t.dim \leq splitVal\}$
    $return\ Anonymize\ (rhs) \cup Anonymize\ (lhs)$
```

(b)

Fig. 4. Principle of Median Mondrian and its algorithm

At each iteration, the algorithm chooses a dimension and checks the possibility of splitting a group into twogroups (i.e. splitting the area on the median value of this dimension). A group can be divided into two groups if eachresulting group contains at least k individuals. If the division is not possible, the corresponding group is marked. Thesplitting process switches to another dimension when all groups are marked for the current dimension. It stops whenall dimensions have been explored. Then the algorithm performs the possible generalizations, replacing the differentvalues in the same area with the value of their first common parent in the generalization hierarchy (recodingprocess). As shown at Fig. 4b, the algorithm is recursive. At the first iteration, “partition” contains all the tuples ofthe table to anonymize. The function “choose_dimension()” (resp. “frequency_set(partition, dim)”) returns thechosen 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 samepartition. 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.

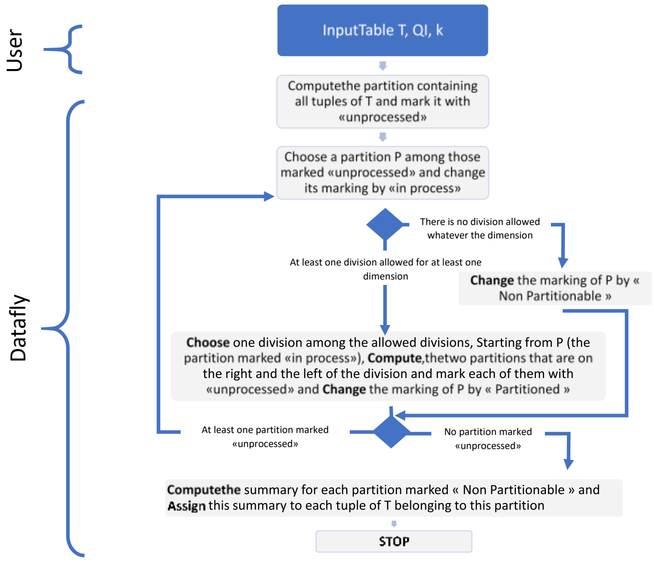

User
Datafly
InputTable T, QI, k
Computethe partition containing all tuples of T and mark it with «unprocessed»
Choose a partition P among those marked «unprocessed» and change its marking by «in process»
There is no division allowed whatever the dimension
At least one division allowed for at least one dimension
Change the marking of P by « Non Partitionable »
Choose one division among the allowed divisions, Starting from P (the partition marked «in process»), Compute,thetwo partitions that are on the right and the left of the division and mark each of them with «unprocessed» and Change the marking of P by « Partitioned »
At least one partition marked «unprocessed»
No partition marked «unprocessed»
Computethe summary for each partition marked « Non Partitionable » and Assign this summary to each tuple of T belonging to this partition
STOP

Fig. 5. Abstraction of Median Mondrian

In order to validate our contribution, we conducted an experiment. The objective was to evaluate theunderstandability of our abstraction by comparing it with those found in the literature. We performed thecomparison 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 familiarwith algorithmic and programming techniques but not aware of anonymization algorithms. To avoid any biasedinterpretation of the results, we have provided the participants with the same types of representations: ourabstractions have been transformed into textual representation (algorithms).

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

Following the experiment, we started analyzing the collected information. 12 participants executed threealgorithms each. We rejected three illegible executions out of the 36. For each algorithm, the legible executions havebeen grouped into three classes. The first (resp. the second one) gathered all the correct executions (resp. all thepartial executions). The last class contained all the erroneous executions. For the partial executions we have deducedfrom 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 MedianMondrian (Fig. 4b). Table 1 summarizes the results. For each algorithm and each profile, the percentages oferroneous, correct and partially correct executions are mentioned.

	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%

Table 1. Synthesis of data collected from the experiment

All the participants who have executed our abstractions didn’t face blocking difficulties. Moreover, only thosethat had ancient knowledge obtained erroneous executions and they are few. Only those who first performed theexecution of Datafly abstraction have then proposed a correct execution of the original Datafly. The sameobservation emerged for Median Mondrian. They unanimously mentioned in the post questionnaire that ourabstraction helped them understanding the original algorithms. Moreover they have ranked the original algorithmsas 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 participantswhatever their programming skills. Threats to validity must be mentioned due to the limited size of ourexperimentation group and the restriction of the study to only two algorithms. However, this first analysisencourages us to persevere in this abstraction effort. We wanted to check if our algorithm representation was easierto understand than the classical one. Therefore, our pool of testers was composed of persons with programmingskills. Thus, they were able to understand both representations. We evaluated both perceived usefulness andobjective usefulness. Perceived usefulness was captured through direct queries regarding how they could understandthe underlying logic of each algorithm. Objective usefulness was measured through the correctness of the resultobtained by participants. Thus, if users with programming skills prefer better perform our abstraction, how muchmore 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 processled us to the identification of three categories using the parameterization and the generalization process describedabove. 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 anabstract representation. Hence we defined a one-to-many mapping between this representation and the differentalgorithms belonging to this category.

We have first grouped the algorithms into categories regarding their basic principles and their type of resultinggeneralization. The first category called MR Recoding1 groups together the algorithms generating multidimensionalgeneralizations (Median Mondrian, InfoGain Mondrian and LSD Mondrian). Their basic specificity is to considertogether all the attributes of the QI.

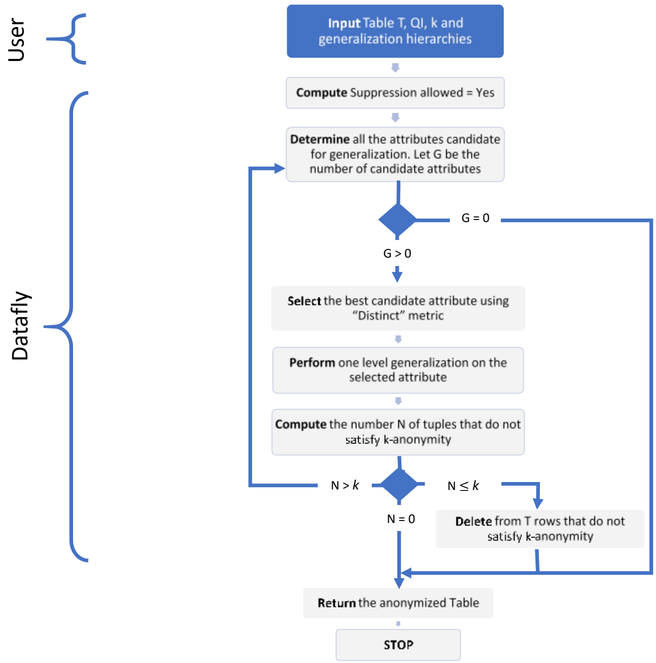

User
Datafly
Input Table T, QI, k and generalization hierarchies
Compute Suppression allowed = Yes
Determine all the attributes candidate for generalization. Let G be the number of candidate attributes
G = 0
G > 0
Select the best candidate attribute using "Distinct" metric
Perform one level generalization on the selected attribute
Compute the number N of tuples that do not satisfy k-anonymity
N > k
N ≤ k
N = 0
Delete from T rows that do not satisfy k-anonymity
Return the anonymized Table
STOP

Fig. 6. Homogenized abstraction of Datafly algorithm

They iteratively divide the set of tuples into groups such that each group satisfiesk-anonymity. Conversely, in the two other categories (LR Recoding and TR Recoding), the generalizations are notmultidimensional. Moreover, at each iteration, they only deal with one attribute of the QI. LR Recoding gathersalgorithms based on a lattice structure representing all the possible generalizations of the original table. Samaratiand Incognito algorithms belong to this category. Finally, TR Recoding includes Datafly, μ-argus, bottom-upgeneralization and top down specialization algorithms. Their specificity is to build directly and iteratively ananonymized table. In order to provide each category with an abstraction, we had to homogenize the nineabstractions. For instance, we had to transform TR Recoding abstractions since some algorithms of this category aretop down processes and others are bottom up, some of them include local or global suppressions and others onlyperform generalizations. Thus, we have introduced a parameter allowing or disallowing suppressions. For example,Datafly allows suppressions (Fig. 6).

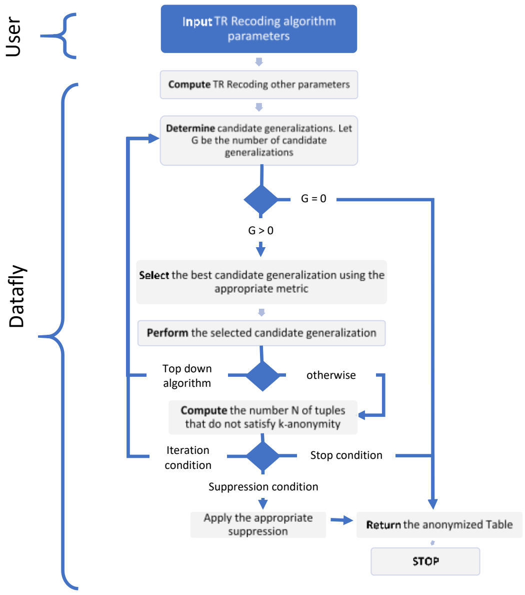

User
Datafly
Input TR Recoding algorithm parameters
Compute TR Recoding other parameters
Determine candidate generalizations. Let G be the number of candidate generalizations
G = 0
G > 0
Select the best candidate generalization using the appropriate metric
Perform the selected candidate generalization
Top down algorithm
otherwise
Compute the number N of tuples that do not satisfy k-anonymity
Iteration condition
Stop condition
Suppression condition
Apply the appropriate suppression
Return the anonymized Table
STOP

Fig. 7. Abstraction of TR Recoding algorithms

We also rephrased some instructions by defining new concepts in order to reach a unique abstraction for eachcategory of algorithms and thus to allow parameterization. For instance, in order to homogenize the concepts “candidate generalization” and “best candidate generalization” with Datafly concepts, we introduced in the latter theconcepts of “candidate attribute” and “best candidate attribute”. As another example of standardization, since eachalgorithm 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 “Tradeoffmetric” in TDS. Thus, the abstraction of Fig. 7 may be instantiated into the four algorithms thanks to a correctparameterization. The same process allowed us to obtain an abstraction for all nine algorithms and for the threerecoding categories. For space reasons, we cannot provide the reader with all the abstractions. Following thisbottom-up logic, we have performed the same abstraction effort to homogenize the three types of recoding. Wegenerated the abstraction of Fig. 8a which in fact represents the generalization technique. Figure 8b instantiates thisprocess for Recoding TR. Step numbers refer to Fig. 7. Thus our recurring abstraction process resulted in ataxonomy of generalization algorithms (Fig. 8). We have defined an abstraction by parameterization usingflowcharts for all the nodes of this taxonomy.

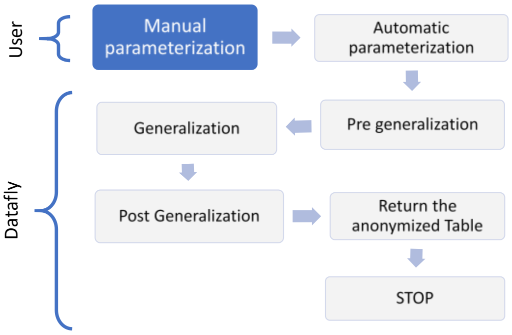


Fig. 8. (a) Abstraction of the generalization technique;

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

Fig 8 (b) An instantiation of the generalization technique

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

4. Conclusion

An extensive attention has been paid to privacy protection by statistics and computer science communities overthese past years. A large body of research deals with anonymization techniques and algorithms. Our research is notdevoted to new algorithms or new techniques. We want to help data publishers with low programming skills inunderstanding the existing techniques and algorithms. This help is made possible thanks to simplifiedrepresentations associated to each technique and algorithm. In this paper, we focused on the generalizationtechnique and on nine of its algorithms, since it is one of the most used techniques for tabular data. The simplifiedrepresentations have been obtained through an abstraction process described in the paper. The abstraction effortallowed us to detect similar behaviors between the nine algorithms and thus to define three main categories ofgeneralization algorithms. We also defined an abstract model for each category and finally an abstraction of thegeneralization technique. Finally, thanks to our categorization, we built a taxonomy of these generalizationalgorithms, helping a novice to understand how all these anonymization algorithms may be differentiated. For spacereasons, the paper only contains some abstraction models. We conducted a controlled experiment, leading toencouraging results since participants found that the proposed abstractions were very useful to understand thealgorithms whatever their programming skills. These abstractions constitute a first step towards the design ofpatterns that will be part of a catalog. The latter will be made available through a guidance approach we plan todesign. 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 previousAnother use case of our abstractions lies in the context of e-learning. Our abstractions may serve as a basis for thedevelopment of tutorials whose purpose is to assist data stewards, students and researchers in learning how to useanonymization algorithms and then to provide them with a well-informed usage of existing anonymization tools. Toensure a better transfer of knowledge, these tutorials could be contextualized according to the level of expertise oftheir potential users. Moreover, we are convinced that our taxonomy can be a starting point for the design of anontology of anonymization techniques and algorithms. This ontology will exhibit conflicts between techniques andthen will contribute to a definition of combined anonymization scenarios for a whole database. Finally, we believethat the availability of such knowledge should facilitate the adoption of new anonymization techniques andminimize the loss of competencies that arise when an employee leaves the company.

Finally, we also expect to develop ontology of techniques andalgorithms. This ontology could be the main component for a decision support system helping a data publisher inselecting 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 ComputingSurveys (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 Scienceand 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 onApplied 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 andAdvanced Engineering, Vol 3(6) (2013)

5. Vinogradov, S., Pastsyak, A.: Evaluation of Data Anonymization Tools. The 4th Int. Conference on Advances in Databases,

Knowledge, andData Applications DBKDA (2012)

6. Poulis G., Gkoulalas-Divanis A., Loukides G., Skiadopoulos S., Tryfonopoulos C.:SECRETA: A System for Evaluating and ComparingRElational 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 andAnalysis 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 □-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 DataMining (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 DataEngineering (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 6thInternational 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. ACMComputing 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)