Modeling and Evaluation of k-anonymization Metrics

Clémence Mauger*[1], Gaël Le Mahec[1], and Gilles Dequen[1]

[1]Laboratoire MIS - Université de Picardie Jules Verne , UFR des Sciences, 33 rue Saint-Leu - 80000 AMIENS - France

Abstract

The k-anonymity concept introduced in (Samarati and Sweeney 1998) proposes a good trade-off between the privacy and the utility of the data published for exploitation. However, minimizing the loss of information throughout the *k-anonymization* of a database is known to be NP-Hard (Meyerson and Williams 2004). Several previous works defined metrics to measure and to optimize this process according to different priorities or different ways of looking at things. In this paper, we first present a unified modeling of the optimization metrics for the *k-anonymization* of a database. Then, we propose different new metrics for this optimization problem. Finally, we evaluate three metrics of the literature and our new metrics using a greedy algorithm along the anonymization process for 21 values of k.

1 Introduction

The amount of daily collected data about individuals is considerable. These data can arise from social networks, commercial or public databases, various sensors or connected devices activated by individuals (e.g activity trackers, medical sensors). The use of these data is a major concern in many scopes and it is now clear that big data analysis is revolutionizing the way scientists, analysts, salespersons and even physicians are working. Moreover, the general public is, legitimately, more and more worried about the exploitation of their private data. Some recent and massive data breaches have increased the mistrust of general public towards companies that hold their private data. Furthermore, lawmakers over the world started to enforce data protection rules. Therefore, using individuals data implies to provide guarantees about the users privacy.

A first approach to protect users privacy is to anonymize or *pseudonymize* the data (by removing or replacing identifiers). However, it has been shown that this is not a sufficient guarantee: one can link attributes in the data to external sources to re-identify the individual (confers *linkage attack* in (Sweeney 2002a)). In (Sweeney 2002a), Sweeney developed the concept of k-anonymity to refine the simple concept of "anonymity". Instead of ensuring that identifier data are removed, k-anonymity ensures that each record is indistinguishable from at least $k-1$ other records in the data with respect to a set of attributes.

This principle is really efficient for privacy but it also alters data in such a way that they could be unusable for further analyses using data mining or machine learning algorithms. Moreover, k-anonymity has been shown to have flaws and other privacy principles emerged (Liu, Giannella, and Kargupta 2008). Other approaches have been developed to improve the guarantees of k-anonymization such as in (Holohan et al. 2017). There are many possible k-anonymous versions of a database (possibly exponential in the number of lines). Among them, some are more data-preserving than the others. Several studies proposed to optimize the data utility by limiting the amount of knowledge lost during the anonymization process. (Byun et al. 2007), (Xu et al. 2006) and (Li et al. 2006) proposed different quality metrics and/or k-anonymization "cost" measures. These metrics can be used during the anonymization operation to lead the data alteration and permit to limit its impact on the global utility of the data.

We will focus on this particular task: trying to keep as much information as possible in the anonymized data set. Therefore, we consider that a "good" anonymization minimizes the information loss and so maximizes the utility of the data. In this paper, we propose a data generalization modeling that eases evaluation, modification and design of information loss metrics depending on the usage of the anonymized data. Then, we define four new metrics and confront them with three information loss metrics of the literature in a systematic comparison. We present the experimental results obtained by making k-anonymity using these metrics. We show that our proposed metrics outperform the metrics of the literature.

The next section briefly presents the k-anonymity concept and the generalization processes that are used to achieve it. Section 3 presents a new modeling of the information loss metrics. Section 4 presents different metrics and a unified way to formalize them as cost functions. It also presents our new metrics. Section 5 presents the comparisons of the different metrics used as local optimization function of a greedy algorithm performed on a public real database.

*Corresponding author

2 k-anonymity and data generalization

PPDP (Privacy Preserving Data Publishing) (Fung et al. 2010) is a field of research aimed at giving individuals confidence that their data will be protected when the database is published. There are two main kinds of approaches in PPDP: partition-based approaches (Sweeney 2002a) and the differential privacy (Dwork 2011). We focus here on the former, particularly the k-anonymity concept.

Considering a flat database (named *table* in the following) of n lines and m columns of attributes, we distinguish three types of attributes: identifier attributes, quasi-identifier attributes and sensible data. The identifier attributes are uniquely linked to an individual (e.g. Social Security Number, driver licence number, ...). To achieve any anonymization of the table, these data must be deleted. The quasi-identifier attributes (Dalenius 1986) (also named *QID*) do not reveal information on their own but, associated with each other or linked to external sources of information, one may be able to link individuals to records (the linkage attack). They are sufficiently distinctive to identify someone. Finally, the sensible values can not be used to identify individuals and are generally the reason of being of the table (examples for a medical table: disease, vital parameters...)

The k-anonymization process consists in generalizing the values of the table, such that the combination of values in the quasi-identifier attributes can be found at least k times in the table. Considering the subset of attributes Q of the table exclusively composed of the quasi-identifiers, the set of lines where all values of Q are identical is an *Equivalence Class*.

By choosing k, we specify the anonymity level that we can provide: the bigger is k, the harder it is to find an individual in the table. The probability to link an individual with a record is then $\frac{1}{k}$.

Each attribute has a domain of definition (for example {Male, Female}, {cat, lion, tiger, dog, ...}, {1, 2, 3, ...}). We define a *generalization* of an attribute's value as the substitution of the original value with a subset of the domain of definition that contains this value. Thus, we can build a hierarchy of generalizations from the single possible values to the complete domain, with intermediate generalization steps with growing subsets of the domain. Figure 1 presents an example of such a generalization hierarchy.

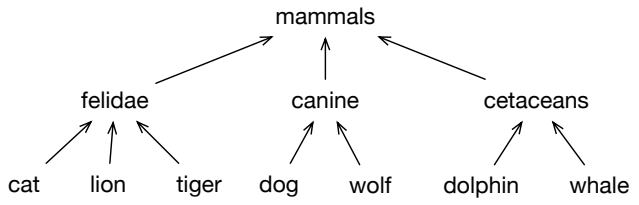

Figure 1: Example of generalization hierarchy

Using such hierarchies for all the quasi-identifiers, from one table, we can compute many k-anonymized table versions. Among these possibilities, some are quite data-preserving while others have lost almost all the utility of the original table. Thus, we need to minimize the information loss to preserve data utility.

Even considering hierarchies of height 1 as the unique way to generalize data (from single value to complete domain), to find an optimal anonymization algorithm is a NP-hard problem (Meyerson and Williams 2004). Many works have been done these years to improve the k-anonymity partition-based approaches. Sweeney and Samarati proposed $MinGen$ (Sweeney 2002b) an optimal algorithm that produced a k-anonymous version of a table (not feasible because of the amount of computation). In (LeFevre, DeWitt, and Ramakrishnan 2005), LeFevre et al. presented a practical framework for single dimensional global recoding called $Incognito$. Even though it was faster than previous algorithms (as (Xu et al. 2006)), it generates too much information loss on the data. With $Mondrian$ (LeFevre, DeWitt, and Ramakrishnan 2006), LeFevre and her team improved the results of $Incognito$ by doing multi-dimensional global recoding. Nevertheless, it only deals with numerical attributes. After that, we notice the emergence of clustering-based anonymization algorithms inspired by what is done in the clustering research field. We can cite $MDAV$ (Domingo-Ferrer and Torra 2005), $k-member$ (Byun et al. 2007) or OKA (Lin and Wei 2008).

To evaluate the quality of their solutions, researchers also studied data quality metrics or information loss metrics. In 2002, Iyengar proposed the *General Information Loss* metric (Iyengar 2002). With the *Discernability metric* (Bayardo and Agrawal 2005), Bayardo and Agrawal wanted to minimize the sizes of the *equivalence classes*. LeFevre used C_{AVG} in (LeFevre, DeWitt, and Ramakrishnan 2006) to measure the distance between the anonymized table and a solution with all *equivalence classes* of size k. There also exists metrics from clustering optimization for numerical attributes such as the Euclidian metric.

3 k-anonymity and our metrics modeling

To build a k-anonymous table, we need to merge the different *equivalence classes* until they have all a size of at least k lines. Merging two classes simply consists in generalizing their respective attributes values until they are the same in the two classes.

Let $\mathcal{G}$ be the generalization graph of an attribute of the table. $\mathcal{G}$ is a directed tree (or hierarchy) of nodes from the single possible values of the attribute (no generalization) to the root representing the complete set of possible values (maximum generalization). The leaves of $\mathcal{G}$ are the possible values of the attribute. The internal nodes are generalization of their children nodes (i.e. a subset of the domain of definition). The *level* of a node is the maximum distance in term of number of edges, from a leaf to the node, assuming the level of a leaf is 0.

Figure 2 shows an example of an attribute generalization tree. The attribute have 6 possible values (a, b, c, d, e and f). a, b and c are generalized in A, d and e are generalized in B and f is generalized in $*$ (all the possible values). Finally, A and B are generalized in $*$.

In order to favor some generalizations or to avoid some others, we can label the edges of the graph $\mathcal{G}$ with "costs".

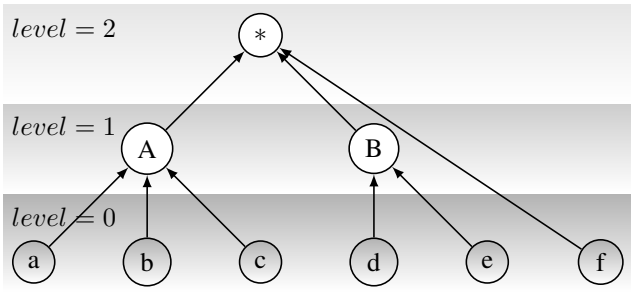

Figure 2: The generalization hierarchy of a table attribute Q

For n a node in the hierarchy and m its father, we denote by $\omega(n, m)$ the cost of the edge (n, m). These weights can be arbitrarily chosen or deduced from the graph itself. Figure 3 presents an example of a weighted generalization hierarchy.

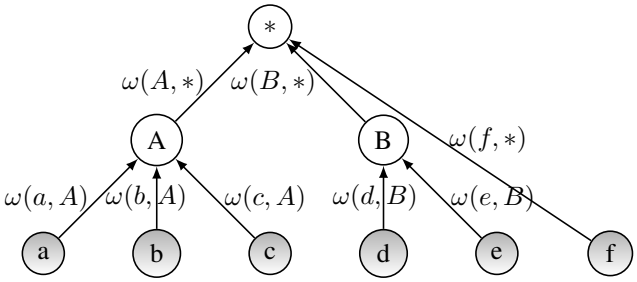

Figure 3: Generalization hierarchy of attribute Q

Using these hierarchies, one for each attribute Q, we can compute cost matrices $Cost_Q$ for generalizations in the hierarchy $\mathcal{G}_Q$ as follows.

We denote by $w_{n\to m}$ the weight of the path from n to m. $w_{n\to m} = \sum_{(i,j)} \omega(i, j)$ with (i, j) every edges on the path from n to m. $w_{n\to m}$ is not defined if there is no path from n to m in the directed graph $\mathcal{G}_Q$. We denote by $LCA(n, m)$ the *Lowest Common Ancestor* (Aho, Hopcroft, and Ullman 1976) of n and m in $\mathcal{G}_Q$.

Then, we define $Cost_Q(n, m) = w_{n\to LCA(n,m)}$, the elementary cost of generalizing n to a set that also contains m. Indeed, to generalize two attribute values such that they are in the same subset consists in finding the smallest subset that contains the both values. In $\mathcal{G}_Q$, this subset is located on the Lowest Common Ancestor node. So, when m is the immediate neighbor of n, $Cost_Q(n, m) = \omega(n, m)$; when n belongs to a higher level m in $\mathcal{G}_Q$, $Cost_Q(n, m) = w_{n\to n} = 0$. Figure 4 presents the different cases in our example.

Figure 4 presents different generalizations costs: for the merge of a and $*$ (in red), $LCA(a, *) = *$, $Cost_Q(a, *) = w_{a\to *} = \omega(a, A) + \omega(A, *)$ and $Cost_Q(*, a) = \omega(*, *) = 0$; for the merge of a and c (in green), $LCA(a, c) = A$, $Cost_Q(a, c) = w_{a\to A} = \omega(a, A)$ and $Cost_Q(c, a) = w_{c\to A} = \omega(c, A)$; for the merge of e and B (in blue), $LCA(B, e) = B$, $Cost_Q(e, B) = \omega(e, B)$

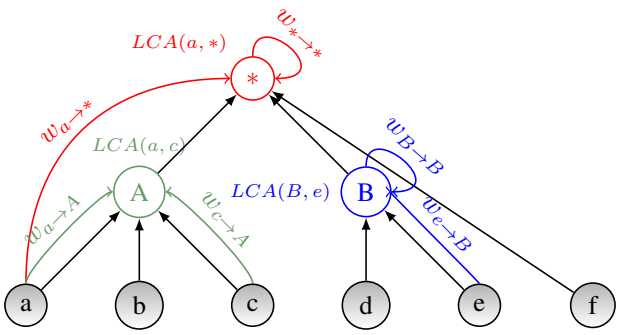

Figure 4: Costs computation examples for the attribute Q

and $Cost_Q(B, e) = \omega(B, B) = 0$.

The cost matrix of our previous example $Cost_Q$ is then:

$$
\begin{array}{c}
 \\
* \\
A \\
B \\
a \\
b \\
c \\
d \\
e \\
f
\end{array}
\begin{array}{c}
\begin{array}{ccccccccc} * & A & B & a & b & c & d & e & f \end{array} \\
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
w_{A\to *} & 0 & w_{A\to *} & 0 & 0 & 0 & w_{A\to *} & w_{A\to *} & w_{A\to *} \\
w_{B\to *} & w_{B\to *} & 0 & w_{B\to *} & w_{B\to *} & w_{B\to *} & 0 & 0 & w_{B\to *} \\
w_{a\to *} & w_{a\to A} & w_{a\to *} & 0 & w_{A\to *} & w_{A\to *} & w_{a\to *} & w_{a\to *} & w_{a\to *} \\
w_{b\to *} & w_{b\to A} & w_{b\to *} & w_{b\to A} & 0 & w_{b\to A} & w_{b\to *} & w_{b\to *} & w_{b\to *} \\
w_{c\to *} & w_{c\to A} & w_{c\to *} & w_{c\to A} & w_{c\to A} & 0 & w_{c\to *} & w_{c\to *} & w_{c\to *} \\
w_{d\to *} & w_{d\to *} & w_{d\to B} & w_{d\to *} & w_{d\to *} & w_{d\to *} & 0 & w_{d\to B} & w_{d\to *} \\
w_{e\to *} & w_{e\to *} & w_{e\to B} & w_{e\to *} & w_{e\to *} & w_{e\to *} & w_{e\to B} & 0 & w_{e\to *} \\
w_{f\to *} & w_{f\to *} & w_{f\to *} & w_{f\to *} & w_{f\to *} & w_{f\to *} & w_{f\to *} & w_{f\to *} & 0
\end{pmatrix}
\end{array}
$$

Using the cost matrices of the different attributes of the table, we can define a *metric* $\mathcal{MC}$ for the merge of two equivalence classes by simply summing the different generalization costs of the attributes of the two classes.

Let $\mathcal{Q} = \{Q_1, ..., Q_m\}$ be the set of quasi-identifiers attributes. Let C_1 and C_2 two equivalence classes containing respectively $|C_1|$ and $|C_2|$ lines ($[x_{Q_1}, x_{Q_2}, ...x_{Q_m}]$ and $[y_{Q_1}, y_{Q_2}, ...y_{Q_m}]$).

$$\mathcal{MC}(C_1, C_2) = \sum_{i=1}^{m} (Cost_{Q_i}(x_{Q_i}, y_{Q_i}) \times |C_1| + Cost_{Q_i}(y_{Q_i}, x_{Q_i}) \times |C_2|)$$

Indeed, to merge C_1 and C_2, we have to generalize the $|C_1|$ lines of the C_1 equivalence class and the $|C_2|$ lines of the C_2 equivalence class such that the two classes have their quasi-idenitifiers identical.

4 Information loss metrics

In this section, we will introduce the metrics chosen for our study. First of all, we will present three metrics of the literature in the light of our new modeling: *Distortion* (Li et al. 2006), *NCP* (Xu et al. 2006) and *Total* (Byun et al. 2007). Then, we will expose four new metrics: *Lost Leaves Metric* (LLM), *Normalized Lost Leaves Metric* ($NLLM$), *Wid Lost Leaves Metric* ($WLLM$) and *Wid Normalized Lost Leaves Metric* ($WNLLM$). For a sake a clarity, we give some notations, valid for all the rest of the section.

Let T be a table. Let $\mathcal{Q} = \{Q_1, ..., Q_m\}$ be the set of quasi-identifier attributes of T. For all $i \in \{1, ..., m\}$, we denote by $\mathcal{G}_i$ the generalization hierarchy of Q_i. Let h_i be

the number of nodes on the longest path in $\mathcal{G}_i$ (the highest level of the hierarchy plus 1). We denote h_{max}, the maximum size of the hierarchies of the different attributes $h_{max} = \max_{i=1,...,m}(h_i)$ (e.g. $h_{max} = 3$ in the example of Figure 2).

Let wid (the QID weights from (Pramanik, Lau, and Zhang 2016)) be a map from $\mathcal{Q}$ to $\mathbb{R}$ defined by $wid(Q_i) = 1 - \frac{(h_i-1)^m}{\sum_{j=1}^{m}(h_j-1)^m}$ for all $i \in \{1, ..., m\}$. If $|\mathcal{Q}| = 1$, we set $wid(Q_1) = 1$.

Let Q_i belong to $\mathcal{Q}$. Let v_i be a value of $\mathcal{G}_i$ and l_{v_i} be the level of v_i in $\mathcal{G}_i$. We define $nl(v_i)$ as the number of leaves in the subgraph whose root is v_i. $nl(\mathcal{G}_i)$ stands for the number of leaves in $\mathcal{G}_i$ (i.e. nl(root of $\mathcal{G}_i$)).

Let $C = [v_1, ..., v_m]$ be an equivalence class of T. We will explicit the cost of C for each metric in the set $\{Distortion, NCP, Total, LLM, NLLM, WLLM, WNLLM\}$.

To make the link with our modeling, we take n_1 and n_2 two nodes of the generalization hierarchy $\mathcal{G}_i$ of the attribute Q_i such that there exists a directed edge from n_1 to n_2 (i.e. $l_{n_1} < l_{n_2}$).

Three metrics of the literature

The metric $Distortion$ (Li et al. 2006) takes into account the level of the generalized value in the generalization hierarchy and does not deal with the number of possible values of the attribute (i.e. with the width of the hierarchy). For each attribute, it assigns weights to the transitions in the generalization hierarchy and makes sum of them from the original value to the generalized value. We use the $wids$ as in (Pramanik, Lau, and Zhang 2016) to obtain an improved version of the metric $Distortion$. These factors aims to give a penalty to attributes with generalization hierarchies of small heights. The distortion of C is:

$$Distortion(C) = |C| \sum_{\substack{i=1 \\ l_{v_i}>0}}^{m} \frac{\sum_{j=1}^{l_{v_i}} c_{j,j-1}}{\sum_{j=1}^{h_i-1} c_{j,j-1}} \times wid(Q_i).$$

The coefficient $c_{j,j-1}$ adds a weight on the transition from level j to level $j-1$ in the generalization hierarchy. In order to take into account the different heights of the hierarchies, we can choose $c_{j,j-1} = \frac{1}{h_i - j}$ for each $j \in \{1, ..., h_i - 1\}$ and each $i \in \{1, ..., m\}$.

The weights of the generalization hierarchy from which we compute the cost matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:

$$\triangleright\, \omega(n_1, n_2) = \frac{\sum_{j=l_{n_1}+1}^{l_{n_2}} c_{j,j-1}}{\sum_{j=1}^{h_i-1} c_{j,j-1}} \times wid(Q_i).$$

Exposed in (Xu et al. 2006), *Normalized Certainty Penalty* (NCP) is an extension of (Iyengar 2002). For a value v_i of an attribute Q_i, NCP calculates the number of leaves which are generalized in v_i. Then, it makes a normalization by dividing by the total number of leaves in the generalization hierarchy. However, it does not use factors on the attributes to equalize the heights of the generalization hierarchies. The weights of the generalization hierarchy of Q_i from which we compute the cost matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:

$\triangleright\, \omega(n_1, n_2) = \frac{nl(n_2) - nl(n_1)}{nl(\mathcal{G}_i)}$.

As for $Distortion$, the metric $Total$ (Byun et al. 2007) takes into account the level of the generalized value in the hierarchy : it divides the height of the generalized value in the hierarchy by the height of the hierarchy minus 1. Thereby, the higher the value in the hierarchy, the more expensive it will be.

The weights of the generalization hierarchy from which we compute the cost matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:

$\triangleright\, \omega(n_1, n_2) = \frac{l_{n_2} - l_{n_1}}{(h_i - 1)}$.

Our contribution

We now propose our metric, LLM, and three variants of it. The idea is to have an overview of the impacts of normalization on the costs and of the use of the attributes' factors. The cost of C for the metric LLM is given by following equation:

$$LLM(C) = |C| \sum_{i=1}^{m} nl(v_i) \times \frac{h_{max}}{h_i}.$$

The factor $\frac{h_{max}}{h_i}$ aims to give a penalty to attributes with generalization trees of small heights. With this metric, we would like to put more weights on the values that induce the most lost leaves and on the attributes with the lowest heights. Here, we do not make a first normalization on the specific cost of a value as for the previous metrics.

The weights of the generalization hierarchy from which we compute the costs matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:

$\triangleright\, \omega(n_1, n_2) = (nl(n_2) - nl(n_1)) \times \frac{h_{max}}{h_i}$.

We now introduce combinations of the previous metrics. We mix LLM, NCP and the factors put on the attributes to obtain three new metrics.

For the first one, we combine NCP and the factor in LLM to get the *Normalized LLM*.

$$NLLM(C) = |C| \sum_{i=1}^{m} \frac{nl(v_i)}{nl(\mathcal{G}_i)} \times \frac{h_{max}}{h_i}.$$

With $NLLM$, we would like to know if the factors $\frac{h_{max}}{h_i}$ are effective in preserving information in the table.

The weights of the generalization hierarchy from which we compute the cost matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:

$\triangleright\, \omega(n_1, n_2) = \frac{nl(n_2) - nl(n_1)}{nl(\mathcal{G}_i)} \times \frac{h_{max}}{h_i}$.

For the *Wid LLM*, we change the factors in LLM by the $wids$ used with *Distortion* in order to find out which one is the most interesting in terms of information retention.

$$WLLM(C) = |C| \sum_{i=1}^{m} nl(v_i) \times wid(Q_i).$$

The weights of the generalization hierarchy from which we compute the cost matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:
$\triangleright\ \omega(n_1, n_2) = (nl(n_2) - nl(n_1)) \times wid(Q_i)$.

Finally, in the *Wid Normalized LLM*, we use NCP with the wid.

$$WNLLM(C) = |C| \sum_{i=1}^{m} \frac{nl(v_i)}{nl(\mathcal{G}_i)} \times wid(Q_i).$$

The weights of the generalization hierarchy from which we compute the cost matrix are, for (n_1, n_2) in the edges of $\mathcal{G}_i$:
$\triangleright\ \omega(n_1, n_2) = \frac{nl(n_2) - nl(n_1)}{nl(\mathcal{G}_i)} \times wid(Q_i)$.

LLM, $NLLM$, $WLLM$ and $WNLLM$ give a range of combinations that we could compare with the metrics of the literature. Thanks to our modeling, we could express the different formulas in the same homogeneous way.

5 Experiments and metrics comparisons

For our experiments, we choose to work on the *Adult Data Set*, from the UC Irvine Machine Learning Repository (UCIrvine 1987). This data set is available online and it is commonly used to test anonymization algorithms (LeFevre, DeWitt, and Ramakrishnan 2005; Li et al. 2006; Byun et al. 2007). We retain nine quasi-identifier attributes: *Age, Gender, Race, Marital status, Education, Native country, Work class, Occupation* and *Salary*. The description of the attributes is presented in Figure 5. We suppress the tuples with unknown values. We obtain 30162 lines, grouped by 19502 pre-existing equivalence classes.

Attribute	Number of values	Generalization (graph's height)
Age	74	5-, 10-, 20-year intervals (5)
Gender	2	Suppression (2)
Race	5	Suppression (2)
Marital status	7	Hierarchy (3)
Education	16	Hierarchy (4)
Native country	41	Hierarchy (3)
Work class	7	Hierarchy (3)
Occupation	14	Hierarchy (3)
Salary	2	Suppression (2)

Figure 5: Description of the attributes of the *Adult Data Set*

We apply on the data set an anonymization algorithm that produces a k-anonymous table and in which a metric is to be optimized to reduce the information loss. It is similar to what is done in (Li et al. 2006). Let $\mathcal{C}(T)$ be the set of the equivalence classes of a table T to be k-anonymized. We denote by $\mathcal{C}mk(T) \subset \mathcal{C}(T)$ the set of equivalence classes such that their sizes are strictly less than k. Let $\mathcal{MC} : \mathcal{C}(T) \times \mathcal{C}(T) \to \mathbb{R}$ be a metric.

In (Meyerson and Williams 2004), the authors proposed a polynomial approximation algorithm for the optimal k-anonymization problem. But this algorithm is polynomial in the number of lines and exponential in k. Considering large data sets and k as a fraction of the number of lines (implies that k is large), the algorithm is impracticable for our experiments. To perform our experiments, we used Algorithm 1 as a simple greedy algorithm to minimize the cost of the k-anonymization.

Algorithm 1 Anonymization algorithm

```
1: procedure K-ANONYMIZATION(T)
2:     while Cmk(T) is not empty do
3:         Choose arbitrarily a class C_small in Cmk(T)
4:         Find a class C in Cmk(T)\C_small such that
   MC(C_small, C) is minimal
5:         Merge C_small and C
6:         Update Cmk(T)
7:     end while
8: end procedure
```

At each step, we make the best movement relative to the metric c. We decide to search the best merge of equivalence classes in $\mathcal{C}mk(T)$ instead of $\mathcal{C}(T)$ in order to reduce the computation time. Indeed, for some values of k, we have less equivalence classes to treat. The choice of C_{small} is determined by the way the set $\mathcal{C}(T)$ is computed: we simply use the first element of $\mathcal{C}(T)$.

From now on, we make a distinction between *optimization* and *metric*. The optimizations, in $\mathcal{O}$ ={*Distortion, NCP, Total, LLM, NLLM, WLLM, WNLLM*}, are used during the anonymization process as the function to optimize in the algorithm. The metrics, in $\mathcal{M}$ ={*distortion, ncp, total, llm, nllm, wllm, wnllm*}, are used to evaluate the quality of an anonymized table. We make experiments for 21 values of k between 2 and 15000. We call *requested* k these values of k because they are the minimal equivalence classes' sizes we want to reach when we run the algorithm on the data set (the *effective* k value can be different). For each requested k, we begin from the original data set (i.e. we do not restart from the anonymized table obtained for a smaller k value).

When we run the algorithm with an optimization $o \in \mathcal{O}$, we obtain a k-anonymized table. We denote by $T_{o,k}$ this table. The tables in Figures 6, 7 and 8 summarize the cost of each $T_{o,k}$ with $o \in \mathcal{O}$ for each metric $m \in \mathcal{M}$. That means, for a row m (representing a metric) and a column o (representing an optimization), the value in (m, o) is the percentage of alteration of $T_{o,k}$ for the metric m. The percentage of alteration is $\frac{m(T_o,k)}{m(T^*)} \times 100$ where T^* is the table in which all the rows are generalized with the highest levels. For example, in Figure 7, the value in $(distortion, NLLM)$ means that the percentage of alteration for the metric $distortion$ of the 100-anonymous table generated by the algorithm fitted

thanks to $NLLM$ is 24%. Then, we make a ranking respect to each metric: the optimization that produces the smallest percentage of alteration with respect to a metric m is at rank 1 for m. The rank is given in brackets on the tables of Figures 6, 7 and 8.

For a requested k equal to 2, LLM is a little worse than the other metric but the others are on the same range. From a certain value of the requested k, some optimizations do not produce the best result considering the same metric to evaluate the final database: for example, for a requested k equal to 100 and for the metric ncp, NCP is at rank 4 and $NLLM$ is at rank 1. The table in Figure 7 shows that, for a requested k equal to 100, LLM and $WLLM$ produce a high average percentages of alteration while $Distortion$ and $NLLM$ have the best scores. For a requested k equal to 1500 (table in Figure 8), the optimization $NLLM$ is at rank 1 for 5 out of 7 metrics. These results can be viewed in the plot of the Figure 10.

Optimization Metric	Distortion	NCP	Total	LLM	NLLM	WLLM	WNLLM
distortion	2 (1)	2 (1)	3 (5)	5 (7)	2 (1)	4 (6)	2 (1)
ncp	4 (3)	3 (1)	4 (3)	5 (6)	3 (1)	5 (6)	4 (3)
total	5 (1)	5 (1)	5 (1)	7 (7)	5 (1)	6 (5)	6 (5)
llm	8 (6)	5 (2)	5 (2)	3 (1)	6 (5)	5 (2)	8 (6)
nllm	3 (2)	3 (2)	3 (2)	5 (7)	2 (1)	4 (6)	3 (2)
wllm	4 (3)	4 (3)	5 (7)	3 (1)	4 (3)	3 (1)	4 (3)
wnllm	2 (1)	3 (4)	4 (5)	5 (7)	2 (1)	4 (5)	2 (1)
Average	4 (2)	4 (2)	4 (2)	5 (7)	3 (1)	4 (2)	4 (2)

Figure 6: Percentage of alteration for a requested $k = 2$

Optimization Metric	Distortion	NCP	Total	LLM	NLLM	WLLM	WNLLM
distortion	24 (1)	34 (5)	30 (4)	56 (7)	27 (2)	47 (6)	28 (3)
ncp	35 (2)	38 (4)	38 (4)	52 (7)	34 (1)	50 (6)	37 (3)
total	35 (1)	42 (5)	39 (3)	58 (7)	37 (2)	54 (6)	39 (3)
llm	54 (6)	48 (3)	52 (4)	30 (1)	53 (5)	43 (2)	57 (1)
nllm	28 (2)	33 (5)	32 (4)	57 (7)	27 (1)	53 (6)	30 (3)
wllm	39 (3)	45 (7)	43 (4)	32 (7)	43 (4)	26 (1)	43 (4)
wnllm	28 (1)	35 (5)	33 (4)	56 (7)	28 (1)	47 (6)	30 (3)
Average	35 (1)	39 (5)	38 (3)	49 (7)	36 (2)	46 (6)	38 (3)

Figure 7: Percentage of alteration for a requested $k = 100$

Optimization Metric	Distortion	NCP	Total	LLM	NLLM	WLLM	WNLLM
distortion	72 (5)	68 (4)	61 (2)	86 (7)	58 (1)	73 (2)	65 (3)
ncp	67 (2)	82 (7)	67 (2)	81 (6)	63 (1)	78 (5)	70 (4)
total	67 (2)	72 (4)	73 (5)	86 (7)	63 (1)	79 (2)	70 (3)
llm	83 (4)	86 (7)	80 (3)	59 (1)	83 (4)	65 (2)	85 (6)
nllm	60 (3)	65 (5)	59 (1)	85 (7)	59 (1)	80 (2)	63 (4)
wllm	79 (4)	85 (7)	75 (3)	58 (2)	80 (5)	52 (1)	82 (2)
wnllm	63 (3)	69 (4)	62 (2)	85 (7)	59 (1)	75 (5)	75 (5)
Average	70 (3)	75 (6)	68 (2)	77 (7)	66 (1)	72 (4)	73 (5)

Figure 8: Percentage of alteration for a requested $k = 1500$

The plot in Figure 9 represents the minimum k plotted against the requested k. The *minimum* k is the size of the smallest equivalence class of the anonymized table obtained for a requested k. We see that, until a requested k equal to 1000, all the $T_{o,1000}$ for $o \in \mathcal{O}$ have a minimum k equal or very close to the requested k (i.e. for a requested k equal to 2, the smallest equivalence class of the $T_{o,2}$ is of size 2). For a requested k between 1500 and 4000, the minimum k remains relatively close to the requested k and all the optimizations have the same behavior. From a requested k equal to 5000, the minimum k strongly increases compared to the requested k. For instance, for a requested k equal to 7500, the minimum k of $T_{LLM,7500}$ is 12307. For a requested k equal to 10^4, $T_{Distortion,10^4}$ and $T_{Total,10^4}$ only have one equivalence class so the minimum k is 30162. Finally, for a requested k equal to 15000, all the optimizations produce tables with a minimum k equal to 30162.

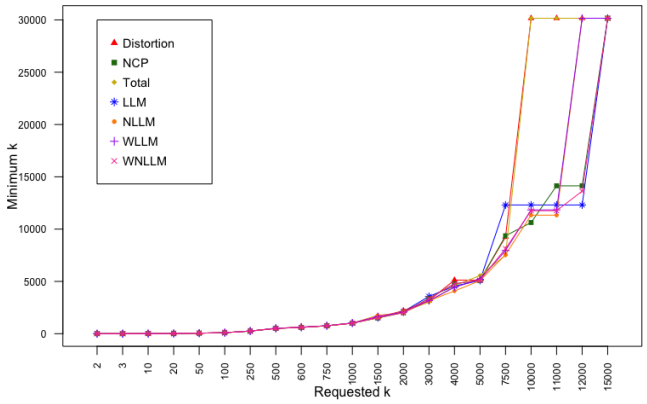

Figure 9: Minimum k plotted against the requested k

For the average percentages of alteration and the percentages of modified values (Figures 10 and 11), the behavior of the optimizations is quite similar. We call *modified value* a value in the k-anonymous table that is different from the value in the original table, whatever the level of generalization applied to it. For a requested k between 2 and 750, $Distortion$ and $NLLM$ have the lowest percentages. LLM and $WLLM$ alter much the data set and modify more values than the others. They are joined by NCP for the average percentage of alteration from a requested k equal to 500. For a requested k between 1000 and 7500, the optimization $NLLM$ is the one with the least average percentage of alteration (we find these results in Figure 8) and percentage of modified values. Finally, for a requested k more than 10^4, the behavior is more chaotic because the number of equivalence classes is small and it needs a lot of huge generalizations.

Figures 12 and 13 present the amount of deleted values as a function of k. A *deleted value* is a value generalized at the highest level of the generalization hierarchy of its attribute. Figure 12 shows the percentage of deleted values considering the total number of values in the table (271458). We can see that using LLM or $WLLM$ produced more deleted values than the other optimizations while using $NLLM$ or $WNLLM$ better preserved the table data than the others.

Figure 13 shows the part of deleted values among the modified ones. For a requested k between 2 and 50, NCP, $NLLM$ and $Total$ stand out from the rest of the optimiza-

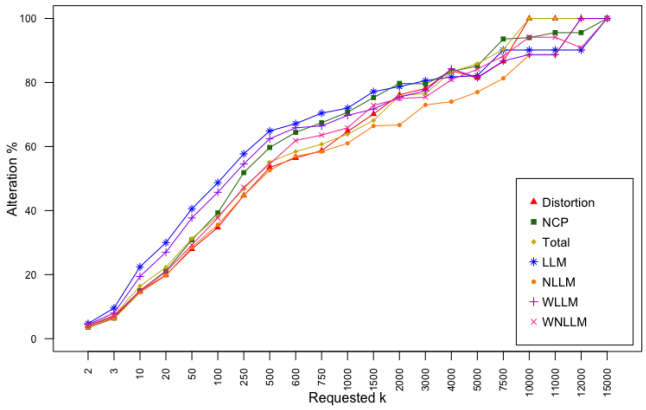

Figure 10: Average percentage of alteration according to k

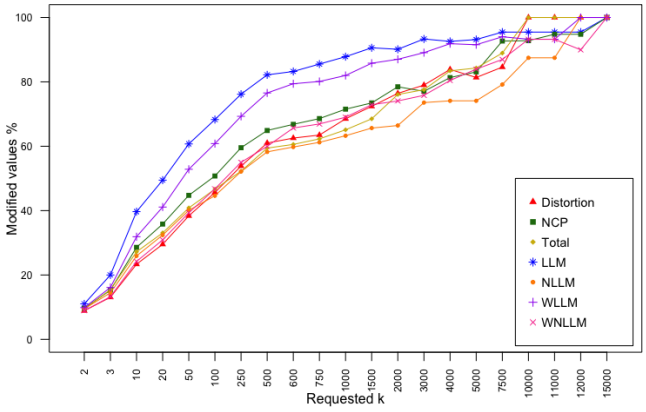

Figure 11: Percentage of modified values according to k

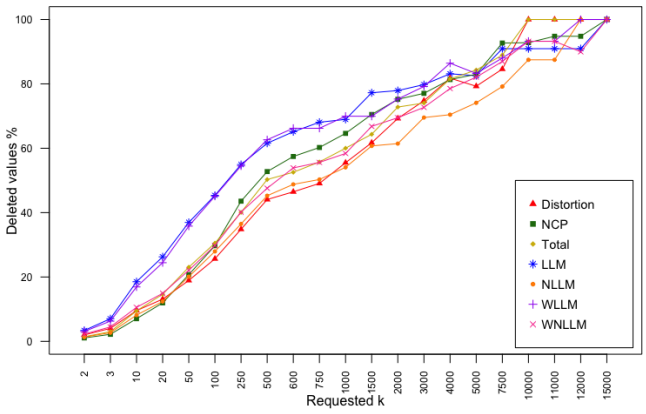

Figure 12: Percentage of deleted values on the total number of values according to k

tions (these optimizations cause less deletion among the generalizations). For a requested k between 100 and 750, $Distortion$ is "better" for this aspect and the other optimizations are in the same range, LLM and $WLLM$ start to decrease the part of deleted value. Finally, for a requested k more than 1000, LLM and $WLLM$ are less deleting data and the others reach fastly 100%, i.e. all the modified values are deleted values.

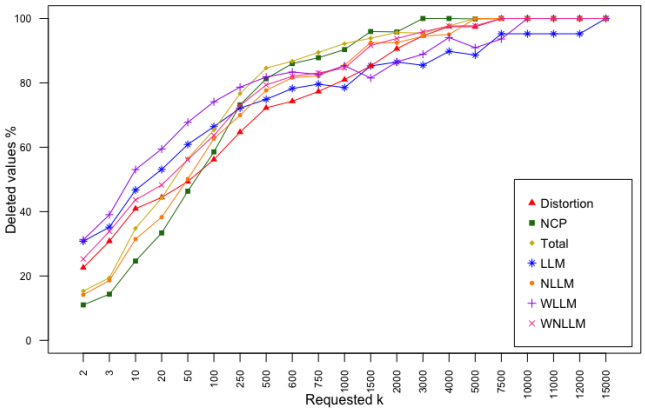

Figure 13: Percentage of deleted values on the number of modified values according to k

6 Conclusion and future works

In this paper, we propose a formalization of information loss metrics for k-anonymization with matrices of costs. We then compare seven metrics, from the literature and from our researches, applying an algorithm on a data set and varying the optimization. We focus on the difference of average percentages of alteration, percentages of modified values and percentages of deleted values. At the end, we can say that the optimization $NLLM$ is the best trade-off to make k-anonymization with a good quality. For a small requested k compared to the size of the data set, the metrics are equivalent. However, when the requested k increases, $NLLM$ manages to the information loss, until a point where all the optimizations have a chaotic behavior (when the requested k is too big compared to the size of the data set).

As a perspective, we will consider more suitable strategies in the anonymization algorithm (i.e. change the selection processes for C_{small} and C) because we know that the k-anonymous tables are not the optimal. We could also make the study on various data sets to clear features more or less favorable for the optimizations.

Acknowledgments

This research was inspired by the Smart Angel project. The project Smart Angel is sponsored by BPI France as part of "Programme d'Investissements d'Avenir" (PIA) within the PSPC funding scheme. All the experiments was conducted using the MATRICS computing platform of Université de Picardie Jules Verne.

References

Aho, A. V.; Hopcroft, J. E.; and Ullman, J. D. 1976. On finding lowest common ancestors in trees. *SIAM Journal on computing* 5(1):115–132.

Bayardo, R. J., and Agrawal, R. 2005. Data privacy through optimal k-anonymization. In *21st International Conference on Data Engineering (ICDE'05)*, 217–228.

Byun, J.-W.; Kamra, A.; Bertino, E.; and Li, N. 2007. Efficient k-anonymization using clustering techniques. In Kotagiri, R.; Krishna, P. R.; Mohania, M.; and Nantajeewarawat, E., eds., *Advances in Databases: Concepts, Systems and Applications*, 188–200. Berlin, Heidelberg: Springer Berlin Heidelberg.

Dalenius, T. 1986. Finding a needle in a haystack or identifying anonymous census records. *Journal of official statistics* 2(3):329.

Domingo-Ferrer, J., and Torra, V. 2005. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery* 11(2):195–212.

Dwork, C. 2011. *Differential Privacy*. Boston, MA: Springer US. 338–340.

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

Holohan, N.; Antonatos, S.; Braghin, S.; and Aonghusa, P. M. 2017. (k,ϵ)-anonymity: k-anonymity with ϵ-differential privacy.

Iyengar, V. S. 2002. Transforming data to satisfy privacy constraints. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, 279–288. New York, NY, USA: ACM.

LeFevre, K.; DeWitt, D. J.; and Ramakrishnan, R. 2005. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, 49–60. New York, NY, USA: ACM.

LeFevre, K.; DeWitt, D. J.; and Ramakrishnan, R. 2006. Mondrian multidimensional k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, 25–25.

Li, J.; Wong, R. C.-W.; Fu, A. W.-C.; and Pei, J. 2006. Achieving k-anonymity by clustering in attribute hierarchical structures˙ *Data Warehousing and Knowledge Discovery* 405–416.

Lin, J.-L., and Wei, M.-C. 2008. An efficient clustering method for k-anonymization. In *Proceedings of the 2008 International Workshop on Privacy and Anonymity in Information Society*, PAIS '08, 46–50. New York, NY, USA: ACM.

Liu, K.; Giannella, C.; and Kargupta, H. 2008. A survey of attack techniques on privacy-preserving data perturbation methods. In *Privacy-Preserving Data Mining*. Springer. 359–381.

Meyerson, A., and Williams, R. 2004. On the complexity of optimal k-anonymity. In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, 223–228. New York, NY, USA: ACM.

Pramanik, M. I.; Lau, R. Y. K.; and Zhang, W. 2016. K-anonymity through the enhanced clustering method. In *2016 IEEE 13th International Conference on e-Business Engineering (ICEBE)*, 85–91.

Samarati, P., and Sweeney, L. 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, technical report, SRI International.

Sweeney, L. 2002a. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05):557–570.

Sweeney, L. 2002b. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10(05):571–588.

UCIrvine. 1987. Machine Learning Repository. [Online; accessed on June 2019] https://archive.ics.uci.edu/ml/index.php.

Xu, J.; Wang, W.; Pei, J.; Wang, X.; Shi, B.; and Fu, A. W.-C. 2006. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, 785–790. New York, NY, USA: ACM.