Publishing Anonymized Set-Valued Data via Disassociation towards Analysis

Nancy Awad, Jean-François Couchot, Bechara Al Bouna, Laurent Philippe

future internet

MDPI

Article

Publishing Anonymized Set-Valued Data Via Disassociation Towards Analysis

Nancy Awad [1,2], **Jean-Francois Couchot** [1], **Bechara Al Bouna** [2] **and Laurent Philippe** [1]

[1] Femto-ST Institute, UMR 6174 CNRS, University of Bourgogne-Franche-Comte, 25000, France; jean-francois.couchot@univ-fcomte.fr; laurent.philippe@univ-fcomte.fr (L.P.)
[2] TICKET Labortary, Antonine University, Hadat-Baabda, 1003, Lebanon; bechara.albouna@ua.edu.lb (B.A.B);
* Correspondence: nancy.awad@ua.edu.lb

Received: 17 March 2020; Accepted: 15 April 2020; Published: date

Abstract: Data publishing is a challenging task for privacy preservation constraints. To ensure privacy, many anonymization techniques have been proposed. They differ in terms of the mathematical properties they verify and in terms of the functional objectives expected. Disassociation is one of the techniques that aim at anonymizing of set-valued datasets (e.g., discrete locations, search and shopping items) while guaranteeing the confidentiality property known as k^m-anonymity. Disassociation separates the items of an itemset in vertical chunks to create ambiguity in the original associations. In a previous work, we defined a new ant-based clustering algorithm for the disassociation technique to preserve some items associated together, called utility rules, throughout the anonymization process, for accurate analysis. In this paper, we examine the disassociated dataset in terms of knowledge extraction. To make data analysis easy on top of the anonymized dataset, we define neighbor datasets or in other terms datasets that are the result of a probabilistic re-association process. To assess the neighborhood notion set-valued datasets are formalized into trees and a tree edit distance (TED) is directly applied between these neighbors. Finally, we prove the faithfulness of the neighbors to knowledge extraction for future analysis, in the experiments.

Keywords: anonymization; knowledge extraction; ant colony clustering; association rules; utility; privacy; disassociation

1. Introduction

Set-valued data is one of the data formats that can be extracted from social networks, it is characterized by associating a set of values to individuals. Set-valued data is common in logs of search engines, search by hashtags, and can be also found in databases such as health databases and market basket records. This work is an extension of [1] which studies the privacy-utility trade-off of certain predefined associations in an anonymized set-valued data through the disassociation technique, defined first in [2]. The aim of this paper is to continue the investigation and reconstruct the disassociated set-valued dataset to be considerate to future data analysis. Describing, diagnosing, predicting and prescribing are the main uses of data. Set-valued data provides multiple opportunities for various data mining tasks. Using data mining techniques and learning algorithms,

machine learning infers models of what is underlying in order to predict possible futures. Extracting knowledge from data, whether it be to increase business productivity, drives effective decision making or predict trends or behaviors, is done through discovering patterns inherited in the data. To benefit from the large amount of set-valued data, association rule mining is deployed and used in many fields all the way from discovering the link between diseases to marketing and retail. It does not refer to just one single algorithm or application, but more to a set of applications that can be used to analyze the correlation between data items in order to find relationships between different items and items categories. This study is an investigation of general set-valued data, where data temporality and semantics is not examined. To illustrate the dilemma of data analysis and privacy preservation in set-valued data, let us consider an example of a mobility data, which stores the GPS location of individuals, each record corresponds to the set of visited cities by an individual. Finding a very high association rule between three cities $\{\{\text{Valetta, Bratislava}\} \Rightarrow \{\text{Salzburg}\}\}$ might indicate a high demand for mobility solutions between the cities. On the other side, publishing set-valued data raises the problem of privacy preservation. If an individual, Bob, visited {Salzburg (Austria), Valetta (Malta), Bratislava (Slovakia), Bergamo (Italy)} and an attacker knows that Bob visited both {Valetta, Bratislava}; he/she can predict with high probability that Bob also visited {Salzburg}. Publishing the dataset unrefined, fails to protect the privacy of Bob's mobility data.

From the above example, we can see that it is necessary to extract knowledge from set-valued datasets but at the same time, the privacy of the individuals should be considered wisely. Publishing raw data is a public disclosure of information collected and puts the privacy of individuals at risk. For that reason, privacy-preserving mechanisms should be applied to the data before publishing it.

Anonymization is the process that imposes a level of privacy on the data in an attempt to protect the privacy of individuals participating in the dataset. Anonymization techniques that can be deployed to protect the confidentiality of user data is a wide area of research[3–10]. Disassociation, is an anonymization technique presented in [2], particularly designed for set-valued data. Basically, it set-valued data are clustered into homogeneous groups (via horizontal partitioning), and then each cluster is split into record chunks (via vertical partitioning) where items of a set are separated into disconnected subsets to preserve the k^m-anonymity privacy constraint.

However, the anonymization approach aims are both: to provide a dataset respecting privacy and to make the published data valuable for direct analysis. A trade-off must be found between privacy preservation and the benefits of knowledge extraction from datasets.

Guided by the disassociation technique our investigation started in [1] by studying and improving the probabilistic preservation of associations in a disassociated dataset. After examination of the probabilistic preservation results, we present an optimization of disassociation for a predefined set of associations, we call utility rules. The goal is to save the utility rules from vertical partitioning as much as possible, while respecting the k^m-anonymity privacy constraint of disassociation.

A derivative of the ant-based algorithm for the disassociation has been proposed to group data whilst respecting the utility rules.

The new problematic in this extension of the work [1], is that in reality, we cannot define every association present in the original dataset as a utility rule in order to optimize its preservation for future analysis. Therefore, the study of association rules in a disassociated dataset cannot be limited to items preserved together without a break up after vertical partitioning. To run an analysis on a disassociated dataset, we must compute a huge set of pre-images of the disassociated dataset, by reconstructing all possible associations between subsets of items disconnected. This new data format, the disassociated dataset, by breaking down a set of items into disconnected subsets of items, makes the analysis over it hard or at least time consuming to achieve. Primarily, it is useless to publish a dataset that cannot be helpful for data analysis.

In this extension, the goal is to publish an anonymized set-valued dataset that falls in the neighborhood of the original one, consequently preserving its original format. In this context, the "neighbor datasets" terminology seems to fall perfectly to our intention. In an effort to make this article self-sufficient, Section 2 recalls and summarizes our previous work [1] for probabilistic preservation of utility in a disassociated dataset and its implementation as ant clustering. In Section 3, we present our first contribution where we define "neighbor datasets" as two similar datasets that fall under a certain radius of distance and present a way to asses the distance between the "neighbor datasets" by formalizing the datasets into trees. Our second contribution in this work, presented in Section 4, is a technique that generates a neighbor datasets to the original, from its disassociated result, which can be seen as a statistical-based re-association. We investigate our solution from two perspectives. First, we look at the preservation of association rules which will reflect the synthetic similarity of neighbor datasets. Second, to generalize our approach, we evaluate the distance between the original and reconstructed datasets to have an overview of the created neighborhood using the distance introduced above. Figure 1 summarizes of the whole solution depicted in this article. All experimental results that confirm the applicability of our approach are presented in Section 5. Finally, Section 6 presents conclusive remarks.

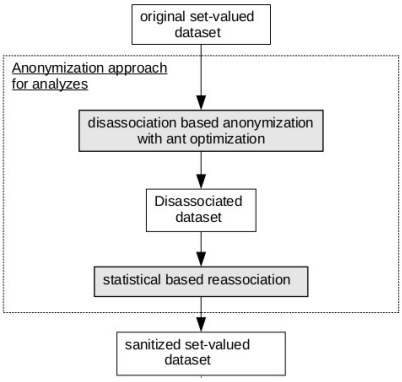


Figure 1. Synthesis of the anonymization of setvalued datasets based on disassociation optimized by ant colonies and a reconstruction respecting the statistical item appearance frequencies.

2. Disassociation and utility awareness for associations

The dissociation technique [2] ensures the constraint of k^m-anonymity by separating the elements of a record into several chunks within the same group. It thus creates association ambiguity between its separated terms, which causes a reduction of the utility for the association in question. Dissociation, as defined by Terrovitis, is based on two assumptions. The first is that no one association is more significant than another. The second is that the data should not be modified, generalized or deleted. In the next section, we provide an algorithm to preserve

a better utility for a set of predefined associations, called the utility rules, by reducing the amount of split-ups a utility rule has to endure in order to preserve k^m-anonymity [2].

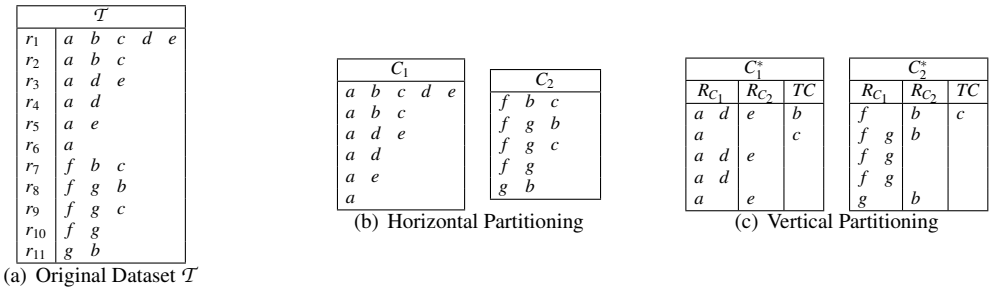
Table 1 recalls the basic notations used in this paper.

Table 1. Notations used in the paper.

$\mathcal{D}$	a set of items
$\mathcal{T}$	a table containing individuals related records
$\mathcal{T}^*$	a table anonymized using the disassociation technique
r	a record (of $\mathcal{T}$) which is set of items associated with a specific individual of a population
I	an itemset included in $\mathcal{D}$
$s(I,\mathcal{T})$	the support of itemset I i.e., the number of records in $\mathcal{T}$ that are superset of I
C	a cluster in a disassociated dataset, formed by the horizontal partitioning of $\mathcal{T}$
C^*	a vertically partitioned cluster C that results in record chunks and a term chunk
R_C	a record chunk from the vertically partitioned cluster C^*
T_C	the term chunk from the vertically partitioned cluster C^*
δ	maximum number of records allowed in a cluster, also know as the maximum cluster size

2.1. Disassociation of a set-valued data

This section is dedicated to show how disassociation works in terms of privacy and utility. We use Figure2 to illustrate an example of disassociation, applied with $k = 3$, $m = 2$ and $\delta = 6$.



(a) Original Dataset $\mathcal{T}$ (b) Horizontal Partitioning (c) Vertical Partitioning

Figure 2. Example of disassociation.

Horizontal partitioning: is the process of clustering the records of a datasets following a naive similarity function. Iteratively and until all records are clustered, from non clustered records, horizontal partitioning groups at most δ records containing the most frequent item at the current iteration. With that, horizontal partitioning fails to take into consideration the associations in the dataset, relying only on one common term to cluster the records. Figure 2(b) reflects the process of horizontal partitioning, where records containing the most frequent item, a, $s(a,\mathcal{T}) = 6$, are grouped together within cluster C_1, and all the other records within C_2. Both clusters have a size less than δ.

Vertical partitioning: is the process that verifies the k^m-anonymity privacy constraint by vertically cutting every cluster into record chunks RC. A record chunk represents sub-itemsets of the cluster's records verifying k^m-anonymity between their items. A term chunk TC, is added to the vertical cut for items with a support of less than k. This process is known as vertical partitioning and is the core of privacy preservation in disassociation. In our example, vertical partitioning is applied over C_1 and C_2. Associations in the clusters are split into different record chunks when k^m-anonymity is not verified.

Figure 2(c) represents the final result of horizontal and vertical partitioning.

To illustrate the effect of disassociation on the utility of associations, let us consider that the frequency of the association $\{b,c\}$, is valuable for future analysis. From the result of disassociation, Figure 2(c), we can note that the bond between items b and c is ambiguous. In C_1^*, items b and c are dropped in the term chunk TC having a support less than $k=3$, therefore, camouflaging their exact support and the association between them, if it exists. Similarly, the association between b and c is unclear in C_2^*, with only one advantage over C_1^*; knowing the support of item b.

Let us suppose that we can guide the horizontal partitioning process for the favor of the association $\{b,c\}$ by keeping together all records that its supersets, as in Figure 3(a). Now, association $\{b,c\}$ verifies $k^m - anonynity$ and is totally preserved associated after vertical partitioning, Figure 3(b).

C_1				
a	b	c	d	e
a	b	c		
f	b	c		

(a) Horizontal Partitioning for ur

C_1^*	
R_{C_1}	TC
b c	a
b c	d
b c	f
	e

(b) Vertical Partitioning

Figure 3. Example of a utility driven disassociation.

From this example, we deduce that preserving associations depends significantly on horizontal partitioning. When a set of associations is important for future analysis, a special treatment of horizontal partitioning must be considered. In what follows, we present a solution to preserve high utility for a set of predefined associations, we call utility rules, within the disassociated dataset.

2.2. The Privacy-utility Trade-off in Disassociation

Giving an exact general definition for the utility of the data in the domain of anonymization is irrational. In this work, a utility rule is an association which is important for accurate analysis, especially for aggregate query answering accuracy.

• Let $UR = \{ur_1, ..., ur_u\}$ be a set of predefined associations, that we refer to as utility rules and are important in future analysis.

• Let $s(ur, \mathcal{T})$ be the support of the utility rule ur in the original dataset.

• Let $s(ur, \mathcal{T}^*)$ be the support of the utility rule in the disassociated dataset.

To evaluate the probabilistic preservation of a utility rule in a disassociated dataset, the confidence of a utility rule is analyzed.

Definition (α-confidence)**.** *The α-confidence of a utility rule ur is evaluated as follows:*

$$\alpha = \frac{s(ur, \mathcal{T}^*)}{s(ur, \mathcal{T})}. \tag{1}$$

The term confidence is used to determine the strength of the association between the items of a utility rule ur after disassociation. Statistical queries are based on the support of the associations in question. The α-confidence represents a ratio of the preservation of a utility rule's support of a utility rule, reflected in the final

output of the disassociation.

In [1] the utility of an association in disassociated datasets is evaluated theoretically under the k^m-anonymity privacy model, and is proven to be:

$$\frac{1}{\delta^{|ur|}} \leq \alpha \leq 1. \tag{2}$$

From this perspective of privacy-utility trade-off, we are motivated to contribute with a more insightful horizontal partitioning process, tolerant to the predefined utility rules for future data analysis accuracy.

The next section provides a general description of the clustering problem and then the role of swarm intelligence algorithms for the improvement of data clustering.

2.3. Data Clustering

Clustering by definition is the process of grouping objects with similar characteristics together within a cluster. There exists no unified solution to solve all the clustering problems and it has been proven to be NP-hard [11,12], thus the problem consists of finding an optimal or near-optimal solution. In what follows, we present briefly general techniques used for data clustering.

Classical clustering algorithms attempt to optimize a fitness function in order to minimize the dissimilarity between items within a cluster and maximize it between the clusters. Clustering can be either fuzzy or partitional. For fuzzy clustering, data items may belong to multiple clusters with a fuzzy membership grade like the $c-means$ algorithm [13]. For partitional clustering, clusters are totally disjoint and data objects belong to exactly one cluster like the K-means algorithm [14]. In this work and in accordance with the disassociation principle, we are only interested in partitional clustering.

Bio-inspired algorithms model processes existing in nature for optimization problems. Example of bio-inspired systems used for data clustering are: ant colony system [15], particle swarm optimization [16], artificial bee colony [17]. Those algorithms draw inspiration from the collective behavior of decentralized, self-organized natural social animals. Even though particles of a swarm may have very limited individual capabilities, they can perform very complex jobs, vital for their survival, when acting as a community. Choosing the right bio-inspired algorithm to cluster data relies on the comparability of the given problem's background with the behavior of the bio-particles.

Ant clustering algorithm (ACA) is a population-based stochastic search process, modeled after the social behavior of ants searching for food, sorting larval and cleaning corpse. To pick and drop items, ants action is influenced by the similarity and the density of the data within the local neighborhood. From this behavior, researches introduced many variation of clustering algorithms applicable in a wide range of problems [18–22].

In the next section, we define a variant version of the ant-based clustering algorithm to cluster records that are supersets of the predefined utility rules, for a more utility-guided disassociation.

2.4. Framework of the Algorithm

We are motivated by the need to preserve some items associated together to increase their utility value despite disassociation and the ambiguity that it raises. We refer to those associations as utility rules. Accordingly,

Table 2. Ant colony terminology.

Ant colony	Set of utility rules UR.
Ant	Expert agent a_i working for the benefit of the utility rule ur_i.
Pheromone trail	Square matrix, A, representing the density of each utility rule in each cluster, updated through probabilistic picking-up and dropping functions. It is the shareable memory between the ants.
Food	Data records $\mathcal{T}_{\mid UR}$ from the dataset $\mathcal{T}$, relative to the utility rules such that: $\mathcal{T}_{\mid UR} = \{r \in \mathcal{T} \mid \exists ur \in UR \text{ and } ur \subseteq r\}$
The ant's load	$load(a_i)$ contains a data record from $\mathcal{T}_{\mid UR}$ that the ant a_i is transporting.
Individual ant's job	Picking-up and Dropping a load.

we transform normal horizontal partitioning for the set of records, which are supersets of at least one utility rule, into a clustering optimization problem. Before describing the algorithm, it is important to identify the challenges of the clustering problem in our context:

- A record might enclose multiple utility rules and with partitional clustering, this record should belong to exactly one cluster satisfying one utility rule.
- The common items, belonging or not to a utility rule, between the records affect the distance metrics.
- The maximum cluster size constant, δ, limits the number of records allowed in a cluster.

The proposed algorithm benefits from the studied behaviors of natural ant. Table 2 describes the environment of our clustering problem in the ant colony system terminology.

Let $\mathcal{T}_{\mid UR}$ be the set of records from $\mathcal{T}$ that are supersets of any utility rule $ur \in UR$:

$$\mathcal{T}_{\mid UR} = \{r \in \mathcal{T} \mid \exists\ ur \in UR \ and \ ur \subseteq r\}. \tag{3}$$

Cluster initialization: Every utility rule, ur, has a representative cluster and an expert ant that transports records. The algorithm starts by sending the expert ants in search for records from $\mathcal{T}_{\mid UR}$, containing their representative utility rules; recursively until $\mathcal{T}_{\mid UR}$ is empty.

Pheromone Trail: Square matrix $A = [u][u]$ represents the pheromone trail of the working ants, with u being the number of predefined utility rules, $u = |UR|$. It is the collective adaptive memory of the expert ants and is initialized by the value of the support of each utility rule ur_i in each cluster C_j, such that:

$$A[i][j] = s(ur_i, C_j). \tag{4}$$

We denote by β_{ur_i} the ratio of the records representing ur_i in cluster C_i:

$$\beta_{ur_i} = \frac{A[i][i]}{s(ur_i, \mathcal{T})}. \tag{5}$$

Individual and cooperative work of expert ants:During the clustering process, every ant a_i works for its utility rule ur_i to reach a support threshold $\beta_{predefined} \in [0,1]$ in its representative cluster C_i, such that:

$$\beta_{ur_i} \geq \beta_{predefined} \tag{6}$$

• If $\beta_{ur_i} < \beta_{predefined}$:

Let $d(ur_i, C_j)$ be the density of a utility rule ur_i in C_j:

$$d(ur_i, C_j) = \frac{A[i][j]}{|C_j|}. \tag{7}$$

Each ant a_i chooses the cluster whose density of ur_i is the highest. A record r from this one is then moved to cluster C_i, with $ur_i \subseteq r$. This process is known as the pick up job.

• If $\beta_{ur_i} \geq \beta_{predefined}$:

To speed up the convergence of the solution and prevent the ants from moving aimlessly, for the current iteration, ant a_i works for the benefit of another ant a_j that needs the most help to increase its β_{ur_j}. This process is known as the drop job. We consider that ant a_j is in need for the most help when reaching β_{ur_j} for its utility rule ur_j demands the highest number of iterations:

$$ur_j = \underset{j \in [1,\ldots,u]-i}{\arg\max} \left(s(ur_j, \mathcal{T}) * (\beta_{predefined} - \beta_{ur_j})\right). \tag{8}$$

The next section describes and comments the algorithm implementing the ant-based clustering methodology.

2.5. Utility Guided ant-based Clustering Algorithm (UGAC)

The utility-guided ant-based clustering (UGAC) algorithm, Algorithm 1, is presented and explained in details in [1]. This section recaps the optimization process elaborated for the preservation of the utility rules. UGAC creates for every utility rule $ur_i \in UR$, an expert ant a_i to pick up records that are supersets of ur_i, and drop them in the representative cluster C_i. The choice of records for pick up is guided by the pheromone trail represented with a square matrix $A = [u][u]$ (line 3) reflecting the support of each utility rule in every cluster. Actually, in the PICKUP procedure, an ant a_i chooses to move a record from the cluster that has the highest density of ur_i (line 2) to C_i (line 7-9). After every move, the pheromone matrix, A, is updated with the support of the utility rules in the clusters (lines 10–15).

All this pick up job is executed while $\beta_{ur_i} < \beta_{predefined}$ or if there is less than k records in C_i (line 16-19). Yet, if $\beta_{ur_i} \geq \beta_{predefined}$, the expert ant a_i can work for the benefit of another ant during the current iteration to converge to the optimal solution quickly (line 21-22). Function DROPLOAD , Algorithm 3, finds the utility rule that still needs the most iterations to achieve the $\beta_{predefined}$ (line 2). Then, it calls the PICKUP function, Algorithm 2, to find a record that can be transported to the corresponding cluster.

At the end of the iterations, there exist u clusters, each representing mainly one utility rule. The resulting clusters may have sizes greater than the maximum cluster size δ allowed. Every cluster is split into smaller clusters having respectively a size less or equal to δ (line 28), calling Algorithm 4, when necessary. Algorithm 1 ends by vertically partitioning the resulting clusters from UGAC (line 30) and treats all the records that are not supersets of any utility rule, via the normal processes of disassociation (line 31).

Algorithm 1 Utility guided ant-based clustering algorithm (UGAC).

Input: $\mathcal{T}$, UR, k, δ, $\beta_{predefined}$, it
Output: $\mathcal{T}^*$

$\mathcal{T}_{|UR} = \{r \mid r \in \mathcal{T}\ and \exists\ ur \in\ UR\ and\ ur \subseteq r\}$
$u = |UR|$
create u clusters, u ants and square matrix $A[u][u]$
$it\_count = 0$
while $(\mathcal{T}_{|UR} \neq \emptyset)$ **do**
    for each expert ant a_i **do**
        $C_i = C_i \cup r \mid r \in \mathcal{T}_{|UR}\ and\ ur_i \subseteq r$
        $\mathcal{T}_{|UR} = \mathcal{T}_{|UR} \setminus r$
        for $(j = 0;\ j < u;\ j++)$ **do**
            $A[i][j] = s(ur_i, C_j)$
        end for
    end for
end while
while $(it\_count < it$ **or** $jobless\_ants < u)$ **do**
    $jobless\_ants = 0$
    for each expert ant a_i **do**
        $\beta_{ur_i} = \frac{A[i][i]}{s(ur_i, \mathcal{T})}$
        if $(\beta_{ur_i} < \beta_{predefined}$ or $A[i][i] < k < s(ur_i, \mathcal{T}))$ **then**
            PICKUP$(a_i, ur_i, 0)$
        else
            $jobless\_ants++$
            DROPLOAD(a_i)
        end if
    end for
    $it\_count++$
end while
for each cluster C_i **do**
    $ClustersSet = ClustersSet \Cup$ SPLIT(C_i)
end for
VERTICALPARTITIONING$(ClustersSet)$
DISASSOCIATION$(\mathcal{T} \setminus \mathcal{T}_{|UR})$

Algorithm 3 DropLoad Procedure

procedure DROPLOAD(a_i)
  $ur_d = argmax_{\forall j!=i}(s(ur_j, \mathcal{T}) * (\beta_{predefined} - \beta_{ur_j}))$
  PICKUP($a_i, ur_d, 0$)
end procedure

Algorithm 2 PickUp Procedure

procedure PICKUP(a_i, ur_j, $Recur_{count}$)
  $C_{pu} = \arg\max(\frac{A[j][n]}{|C_n|})$, $\forall n! = j$
  if $|C_{pu}| \leq k and Recur_{count} < u$ **then**
    $Recur_{count}++$
    PICKUP(a_i, ur_j, $Recur_{count}$)
  end if
  $load(a_i) = r \mid r \in C_{pu}$ and $ur_j \subseteq r$
  $C_{pu} = C_{pu} \setminus r$
  $C_j = C_j \Cup load(a_i)$
  for ($l = 0; l < u; l++$) **do**
    if ($ur_l \subseteq load(a_i)$) **then**
      $A[l][pu]--$
      $A[l][j]++$
    end if
  end for
end procedure

Algorithm 4 SplitClusters Function

procedure SPLIT(C_i)
  if $|C_i| > \delta$ **then**
    create new cluster C_{new}
    for (int $l = 0; l < \delta; l++$) **do**
      $load(a_i) = r \mid r \in C_i$
      $C_i = C_i \setminus r$
      $C_{new} = C_{new} \Cup load(a_i)$
    end for
    $AntClusters = AntClusters \Cup C_{new}$
    SPLIT(i)
  else
    $AntClusters = AntClusters \Cup C_i$
  end if
  Return $AntClusters$
end procedure

2.6. Ant-based Clustering Effect on Associations

In [1], the efficiency of the UGAC technique in terms of preservation of the associations represented in the utility rule is evaluated alongside other experiments analyzing the privacy–utility trade-off. Results are very promising for the preservation of the utility rules. In this section, we investigate the effect of ant-based clustering for the predefined utility rules UR on the associations beyond UR.

We chose for the experiment the BMS1 dataset, which contains click-stream E-commerce data, and a set of 70 distinct utility rules UR, extracted from the dataset with different characteristics: highest frequency= 1204 representing frequent association and Lowest frequency = 2 representing a very rare association. Only records that are supersets of the utility rules (36141 records from the original set of 149639 records) are clustered and evaluated.

We compare UGAC (with $\alpha = 0.6$) to normal horizontal partitioning for disassociation and investigate the overall preservation of associations for the 2 clustering techniques using the Relative Association Error (RAE) defined as:

$$RAE = \sum_{\forall A \in \mathcal{T}_{|UR}} \frac{(s(A, \mathcal{T}_{|\mathcal{UR}}) - s(A, \mathcal{T}_{|\mathcal{UR}}{}^{*}))}{AVG(s(A, \mathcal{T}_{|\mathcal{UR}}), s(A, \mathcal{T}_{|\mathcal{UR}}{}^{*}))} \tag{9}$$

Where A is any type of associations present in $\mathcal{T}_{|\mathcal{UR}}$ and $s(A, \mathcal{T}_{|\mathcal{UR}})$, $s(A, \mathcal{T}_{|\mathcal{UR}}{}^{*})$ represent respectively the support of the association A in $\mathcal{T}_{|\mathcal{UR}}$ and its disassociated result $\mathcal{T}_{|\mathcal{UR}}{}^{*}$. In this experiment, two types of associations are evaluated: first, the set of all the couples present in the records related to the utility rules, $\forall (x,y) \in T_{|UR}$, denoted by GA and second the set of utility rules UR.

Table 3 shows that the results of UGAC for GA and UR are much better than normal horizontal partitioning. This indicates that for the records that are treated through UGAC, the associations are not deteriorated on the expense of preserving the UR. Finally, we can say that UGAC is reliable for analysis for the utility rules in question and beyond them.

Table 3. Relative association error (RAE) for GA and UR.

Technique	RAE **for** GA	RAE **for** UR
Horizontal Partitioning	90171	66
$UGAC$	26636	34

The above work, as presented, offers an optimization for disassociating interesting rules in the datasets for analysis. However, a disassociated dataset is not the most appropriate data format for knowledge extraction because it breaks down a set of items into disconnected subsets of items. For this reason, we present in the next section, an algorithm that generates a re-associated dataset from the disassociated one, that we intend to call it a neighbor dataset, by re-associating statistically partitioned subsets to restore the original format of set-valued dataset.

3. Neighbor Datasets: a Road-map for Distance Evaluation

When a dataset is anonymized it cannot hold the exact same records as the original dataset. Yet, for the anonymized dataset to be useful it has to lead analysis in the same direction of the original dataset. In the following, we define a dataset $\mathcal{T}$ and its anonymized version $\mathcal{T}'$ as two neighbor datasets if they fall under a certain radius of distance and consequently lead to close hypothesis when analyzed.

Let us first define neighboring for two datasets. To be able to describe datasets as neighbors, two questions arise. First, should the definition of neighboring datasets be contextual or general? Second, how can we mathematically assess this neighborhood degree of two datasets? In the following, we go over the above two dilemmas. From what we briefly described above, we see that the basic definition of neighbor datasets should

neither discriminate a contextual nor a general analysis. Taking into consideration both very general branches of data analysis we surely need to have a synthetic reproduction of the original dataset $\mathcal{T}$. Then, the value of the items should stay intact without generalization, to preserve the specificity and context of outliers. At the same time, the neighbor dataset should be a synthetic representation of $\mathcal{T}$ and this means that statistically, the two datasets should represent almost the same overview of the data, leading to a close hypothesis. This section first introduces how set-valued datasets are translated into trees (Sec.3.1) and further shows (Sec..3.2) how generated trees are used to compute distances between this dataset.

3.1. Dataset Formalization

This work focuses on unordered records of sets of distinct data. For instance, a summary of visited cities during last month for some travelers, sets of unique words used in web queries for collection of persons, etc. There is no interpretation of the sens of the data neither in terms of semantic nor in terms of values. Data in records are either equal or distinct. They are thus considered as unique. For instance, if the same city is visited twice by a traveler, it will only be counted once in the summary of cities visited during the period under study. Moreover, the order (between records or between innermost data) is not taken into consideration. There is thus no notion of temporality in the data.

From a mathematical point of view, the considered data are multisets of finite sets of discrete data. Multiset allows duplicated records: two distinct user may have indeed visited the same cities, and have then the same data.

Multisets of sets may be efficiently represented by a forest of trees. Figure 4 gives, for example, two trees representing the same multiset.

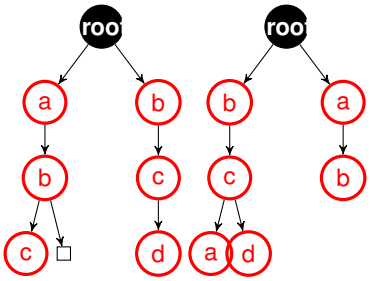


Figure 4. Two trees representing multiset $\{\{a,b,c\},\{b,c,d\},\{a,b\}\}$.

Each path from the root to a leaf defines a set, and conversely. Storing multiset of sets as is thus motivated by the objective of computing distance between two multisets as a distance between two trees.

However, as shown in Figure 4, each multiset of sets can possibly be represented in various ways. In this case, using tree edit distance (TED) will not lead to a distance because the distance between a multiset and itself will not be zero, which would contradict the separation property. The next section presents how this problem is tackled.

3.2. Tree Edit Distance for Datasets

Tree editing distance is a measure that estimates the cost of transforming one tree into another. More precisely, this cost depends on the minimum number of elementary operations (such as deletion, insertion and

renaming applied to a node of a tree) weighted by their cost, to move from one tree to another. This notion extends to trees the edit distance (or Levenshtein Distance) between character strings.

We are then left to translate in a uniform manner multiset of sets to ordered trees. Each set of items (i.e., each record) is thus translated into an ordered list of items, thanks to a lexicographic ordering. It results in a multiset of words where each of them contains at most one occurrence of each letter. Similarly, all these words are lexicographically ordered leading in a sequence of words.

Finally, the distance between the two datasets is the distance between the translated ordered trees. Since the lexical ordering based translation from a multiset of sets is bijective, and since the tree edit distance is a distance, the proposed metric is a distance on datasets. With tree edit distance it is easy to evaluate how far two sets of data are, and more specifically the original and anonymized one.

4. Probabilistic Wheel Re-association for Disassociated Datasets

This section depicts the final motive of our study: How can we publish an anonymized set-valued datasets? First, the process of building a dataset to be published, arising from the privacy constraints of disassociation is described. Then, the relationship between the result of the developed process and tree edit distance is examined from a privacy point of view and the real usefulness of the metric.

4.1. Probabilistic Wheel Re-association Algorithm

The problem of publishing an anonymized set-valued data is transformed into finding a neighbor dataset that shows true features from the original dataset while being distinct, hence faithful to data analysis. In this section, we propose an algorithm, probabilistic wheel re-association, Algorithm 5, to generate neighbor datasets, given hereafter. Roulette wheel selection is a probabilistic optimization method used for selecting potentially useful solutions for recombination. We profit from the disassociation technique to lead the re-association process. Disassociation by default creates ambiguity between associations present in different record chunks of a cluster, while preserving the accuracy for associations found in the same record chunk.Ambiguity on the level of the associations between the record chunks is our playground for the probabilistic wheel re-association. The following solution respects the result of disassociation on different levels:

- What has been disassociated in two distinct clusters should not be re-associated.
- Associations from different record chunks of the same cluster can only be re-associated.
- The associations preserved in a record chunk should not be altered or re-associated between them. They already passed the k^m-anonymity test for anonymization.

ALgorithm 5 takes as input the disassociated dataset $\mathcal{T}^*$, and starts by reconstructing a neighbor cluster C' of each cluster C from $\mathcal{T}^*$ (Line 1). C'_i is the cluster containing the result of the gradual re-association between the record chunks of C_i. At first, C'_i is loaded with the records of the first record chunk R_{C_0} (Line 2). Further, the original number of records in a cluster affects deeply the re-association where empty records show the weakness of the associations between the record chunks. To have the same size of the initial cluster, we add to C'_i a number of empty sets representing what is left from the original cluster size not represented in R_{C_0} due to k^m-anonymity (Line 3–5).

After initializing C'_i, the algorithm applies probabilistic wheel re-association between C'_i and successively every R_{C_j} in C_i (Lines 6-17). Two records r_1 and r_0 are respectively chosen from C'_i and the record chunk in question following the rules of the *SelectRecord* function (Line 9-10). Function *SelectRecord*, Algorithm 6, takes a record chunk, generates the counts for the distinct itemsets in R_C (Line 2) and then constructs the array of cumulative probabilities of the records *ACP* (Line 3). A random number between 1 and 100 is generated (Line 4); the itemset with cumulative probability equal or straight greater than the selected random number is returned. The two selected itemsets, r_0 and r_1, are merged together (Line 11) and moved to the temporary cluster *ReconstructedRecord* (Line 12), waiting for all the itemsets in R_{C_j} to be merged with other itemsets from C_i in a similar way (Line 8-15). The generated itemsets in *ReconstructedRecord* are added to the neighbor cluster C'_i for the merge with the next record chunk (Line 16).

The union of all the generated clusters form the neighbor dataset of the disassociated dataset (Line 18). In Section 5, we present a set of experiments to evaluate the result in terms of neighborhood generation.

Algorithm 5 Probabilistic wheel re-association.

Input: $\mathcal{T}^*$
Output: $\mathcal{T}'$

1: **for** each cluster C_i in $\mathcal{T}^*$ **do**
2: $C'_i = R_{0_{C_i}}$
3: **for** $k = 0, k{+}{+}$, while $|C_i| - |C'_i| > 0$ **do**
4: $C'_i = C'_i \cup \emptyset$
5: **end for**
6: **for** each record chunk $R_{j_{C_i}}$ from cluster C_i such that $j > 0$ **do**
7: $ReconstructedRecord = \emptyset$
8: **while** $\|R_{j_{C_i}}\| > 0$ **do**
9: $r_0 = \text{SELECTRECORD}(C'_i)$
10: $r_1 = \text{SELECTRECORD}(R_{j_{C_i}})$
11: $r_0 = r_0 \cup r_1$
12: $ReconstructedRecord = ReconstructedRecord \cup r_0$
13: $C'_i = C'_i \setminus r_0$
14: $R_{j_{C_i}} = R_{j_{C_i}} \setminus r_1$
15: **end while**
16: $C'_i = C'_i \cup ReconstructedRecord$
17: **end for**
18: $\mathcal{T}' = \mathcal{T}' \cup C'_i$
19: **end for**

Algorithm 6 SelectRecord Function

1: **function** SELECTRECORD(R_C)
2: Assign counts $count_r$ to the distinct records r in R_{i_C}
3: Construct array of cumulative probabilities of the records *ACP*
4: Generate a random number rn between $[1, 100]$
5: From *ACP* select the record r such that $ACP(r) \geq rn$
6: Return r
7: **end function**

4.2. Analysis of Re-associated Datasets in terms of Distance

How can a re-associated dataset be evaluated in terms of neighborhood generation? A suitable distance provides precise indicators about the similarity between objects. As presented in section 3, tree edit distance provides a solution to assess the similarity between set-valued datasets. Using this metric, it is easy to assess, first the distance between two re-associated datasets and second the distance between a re-associated dataset and the original dataset. For two datasets to be neighbors the distance between them should be small, in other words, few modifications should be made to make them similar. Neighborhood is generated when multiple re-associated datasets are close to each other. It is interesting to investigate the process of neighborhood initiation by generating re-associated datasets using the same/distinct privacy constraints. However, using the tree edit distance to evaluate the distance between the generated dataset and the original one, can cause a privacy breach in practice. Let us consider a very basic example where the distance is really low, almost zero: in this case, the public can be sure that they have the original dataset. Yet, we stress the idea that with a big dataset and domain, probabilistically it is very rare to regenerate the original dataset. Therefore, the distance between a re-associated dataset and the original one should not be leaked in any sense to the public, increasing the awareness of an attacker to the background data.

The next section presents experiments evaluating the neighborhood generation and the preservation of the data utility for analysis. Despite our realization of the privacy threat that might occur when calculating the distance with the original dataset, we are going to use the tree edit distance on the original dataset, to prove the efficiency of the probabilistic wheel re-association algorithm in terms of neighborhood generation.

5. Experiments

In what follows, a set of experiments is led to evaluate the probabilistic wheel re-association algorithm on three scales:

- The efficiency of the algorithm in neighborhood generation, using the tree edit distance.
- The data privacy effect on the neighborhood generation.
- The accuracy of knowledge discovery between the original dataset and the published dataset.

The following experiments are conducted on the same dataset, the BMS1 dataset (59,602 records), used in section 2.6.

5.1. Evaluating the Neighborhood of Set-valued Datasets

We start our investigation from the Probabilistic Wheel Re-association algorithm to see how well can it generate a neighborhood from the original dataset. Despite our awareness in terms of privacy worries arising from the calculation of the tree edit distance with the original dataset. Our main objective in these experiments is to evaluate realistically the efficiency of the proposed algorithm; and this cannot be done without the original dataset. To evaluate the neighborhood of the generated datasets, disassociation on $BMS1$ with different k, m and δ constraints is applied and then multiple re-associated datasets for each combination are generated. TED is then calculated between every re-associated dataset and the original one. The implementation of APTED, developed in [23,24], is finally used to calculate the TED.

Table 4. Tree edit distance between the re-associated datasets and the original dataset.

k	m	δ	Average TED	Standard Deviation
2	2	20	89956	425
2	2	40	94112	130
2	2	60	94959	581
3	3	20	80344	379
3	3	40	81383	215
3	3	60	81749	304
4	4	20	78631	53
4	4	40	76800	407
4	4	60	78250	570

The results for the following analysis, are represented in two formats: First, Table 4 represents the average TED calculated between multiple re-associated datasets and the original dataset and the standard deviation of TED, under the same privacy constraints (same k, m and δ).

- Are datasets generated from the same disassociated result reflecting a neighborhood?
- What is the effect of the privacy constraints on the distance?

Second, Figure 5 is a visual representation of the calculated distances for five samples of each combination of k, m and δ, to make more sense of the distances calculated. For a more concrete understanding, we plot the calculated TED relatively to the average TED by δ, without using a scale.

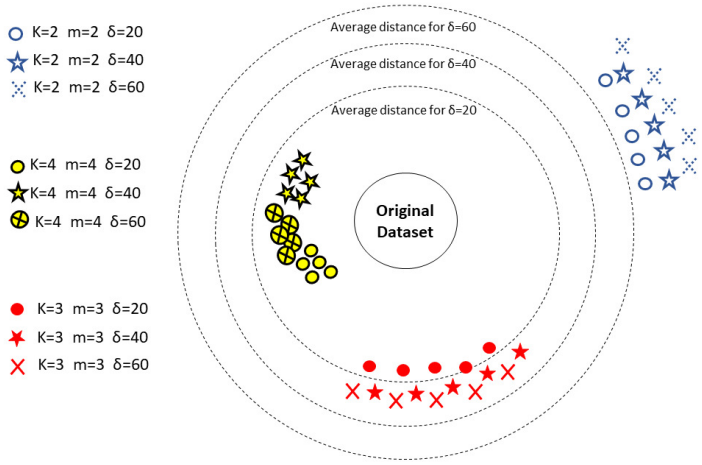


Figure 5. Tree edit distance (TED).

In the following subsections we analyze the represented results to answer respectively the following questions:

5.1.1. The Neighborhood Generation

TED evaluates the transformations that a disassociated dataset has to undergo to transform itself into a re-associated dataset. In this analysis, we are searching for datasets that are relatively in the same distant range from the original dataset. This information is enough to distinguish neighborhoods of datasets. It is notable from the data in Table 4 that datasets generated from the same privacy constraints (k, m, δ), through the probabilistic wheel re-association algorithm, fall under the same distance range, where TED is almost equivalent for the same

k, m, δ combination. This means that from the same disassociated dataset, we can generate datasets that need relatively the same amount of transformations to retrieve the starting point (i.e., the original dataset), where the standard deviation of the TED calculated is fairly small. If we overlook the data represented in Figure 5, it automatically reflects the neighborhood generation, where datasets fall at close radii from the original dataset, especially when generated from the same privacy constraints. We can deduce that the probabilistic wheel re-association algorithm through disassociation, is a reliable tool for generating neighbor datasets, assessed using the tree edit distance.

5.1.2. Data Privacy and Neighborhood Distance

In this section, we investigate the effect of the level of privacy on the neighborhood generation. From the first analysis, we recognized that the same privacy constraint can generate re-associated datasets that have a close radius to the original dataset. What is shockingly interesting is that higher privacy ($k = 4$ and $m = 4$) generate neighbor datasets closer to the original dataset than lower privacy constraints ($k = 2$ and $m = 2$) as shown in Figure 5. This can be explained by the reality that higher privacy is harder to achieve while disassociating the original dataset. In fact, imposing an extra check on associations between items in a cluster, will force dissimilar subsets to be separated into distinct record chunks, thus creating less diversity in a record chunk. Accordingly, the probabilistic wheel re-association algorithm will have less diverse subsets to choose to re-associate together between record chunks. We can notice another behavior related to the maximum cluster size δ, smaller δ generate datasets that are closer to the original dataset and this is due to the fact that when fewer records are allowed in a cluster, diversity in terms of associations is minimized. Therefore, giving less choice to probabilistic wheel re-association by imposing higher privacy (k, m), and lower number of records to re-associate (smaller δ) can lead to generating datasets that are less distant from the original dataset.

To conclude, the above experiments prove the efficiency of the Probabilistic Wheel Re-association algorithm for its ability to generate different solutions, yet taking into account the privacy level imposed in the disassociation phase. The last question we investigated in our experiments is: what can we learn from those disassociated datasets compared to the original dataset? The next section tackles this problem and evaluates the knowledge discovery on a neighbor dataset.

5.2. Neighbor Datasets and Knowledge Discovery

Creating a neighbor dataset is useless if it is not a synthetic representation of the original dataset. We dedicate this section to validate our approach for data analysis faithfulness between the original dataset and its neighbors. To do so, we examine the widely used techniques for mining set-valued data, the frequent itemsets and frequent association rules. Apriori is an algorithm for frequent itemset mining and association rule learning. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger itemsets as long as those itemsets appear sufficiently often in the database. We benefit from the implementation of the apriori algorithm in ARULES, developed in [25,26], to investigate the frequent itemsets and association rules representation in neighbor datasets. In what follows, we consider the average number of records of the

Table 5. Frequent two-itemsets (minimum support = 0.008).

Frequent 2-Itemset acronyms	**Count in $\mathcal{T}$**	**Count in $\mathcal{T}'$**	**Rank in $\mathcal{T}$**	**Rank in $\mathcal{T}'$**
Itemset1	1204	1047	1	1
Itemset2	916	765	2	2
Itemset3	877	707	3	3
Itemset4	771	643	4	4
Itemset5	738	583	5	6
Itemset6	722	603	6	5
Itemset7	631	494	7	9
Itemset8	621	517	8	7
Itemset9	615	408	9	15
Itemset10	615	517	10	8
Itemset11	590	482	11	10
Itemset12	576	453	12	12
Itemset13	571	427	13	13
Itemset14	552	458	14	11
Itemset15	509	408	15	14
Itemset16	506	375	16	17
Itemset17	496	385	17	16

above-generated neighbor datasets and consider it the size of a representative neighbor dataset $\mathcal{T}'$, $|\mathcal{T}'| = 53555$, and we use $BMS1$ as the original dataset, $\mathcal{T}$, with a size $|\mathcal{T}| = 59602$.

5.2.1. Mining Frequent Itemsets

In order to go deeper in the analysis of the neighbor datasets, we turn our attention to the most frequent itemsets. The discovery of frequent itemsets help decision-makers develop new strategies by gaining insights into which items are frequently found together. For this investigation, we use the support metric, defined as the proportion of transactions in the dataset D which contains an itemset X.

$$support(X) = \frac{|X|}{|D|}. \tag{10}$$

Table 5 presents the itemsets with support greater or equal to 0.008. We are able to extract 17 itemsets, representing the most frequent itemsets. To compare the accuracy of knowledge extraction between the original and re-associated dataset, we compared their ranks in terms of frequency. Despite the drop of the count of those itemsets, we can notice that they reappear in the re-associated dataset in the same rank range with mild changes in the ranking, yet they are still the 17 most frequent itemsets in the re-associated dataset. We further tested the three most frequent itemsets, shown in Table 6. We could not detect their presence with the same minimum support, so we dropped it from 0.008 to 0.005, which means they are less frequent in the dataset. However, we found that even for less frequent itemsets we can retrieve them in almost the same rank range. This shows that a neighbor dataset is a faithful representation of the original dataset, by preserving the most frequent itemsets in the same rank range even with the drop in the support.

Table 6. Frequent three-itemsets (minimum support = 0.005).

Frequent 3-Itemset acronyms	Count in $\mathcal{T}$	Count in $\mathcal{T}'$	Rank in $\mathcal{T}$	Rank in $\mathcal{T}'$
Itemset1	417	338	1	1
Itemset2	351	281	2	2
Itemset3	335	274	3	3
Itemset4	322	238	4	5
Itemset5	315	223	5	8
Itemset6	310	231	6	1

5.2.2. Mining Association Rules

The discovery of association rules draws another view on the neighbor datasets. In their default definition, association rules represent a correlation between itemsets. An association rule is defined as an implication of the form $X \implies Y$ where X and Y are two disjoint itemsets, $X \cap Y = \emptyset$. They illustrate a concept of correlation between the two itemsets, where the presence of itemset X shows a strong presence of itemset Y. The discovery of association rules is a more complex characterization of data that depends fundamentally on the discovery of frequent itemsets. We searched for the most frequent association rules by their support, which is relative to their count and the size of the dataset. For this experiment, we extracted the 50 most frequent association rules in the original dataset with a support greater than 0.001. Then, we calculated the average support for those association rules in the generated re-associated datasets, $\mathcal{T}'$.

From Figure 6 we can see that the support of the most frequent association rules is well-preserved through probabilistic wheel re-association. This shows that neighbor datasets are a faithful representation of the original dataset, by preserving reflecting close support. We can notice a loss in 5 Association Rules (support =0); in fact, this is not a real loss this is reflecting association rules that have a support less than the minimum threshold(0.001) after re-association. However, if the need is to preserve predefined associations, ant-driven clustering, investigated thoroughly in Section 2, can be used.

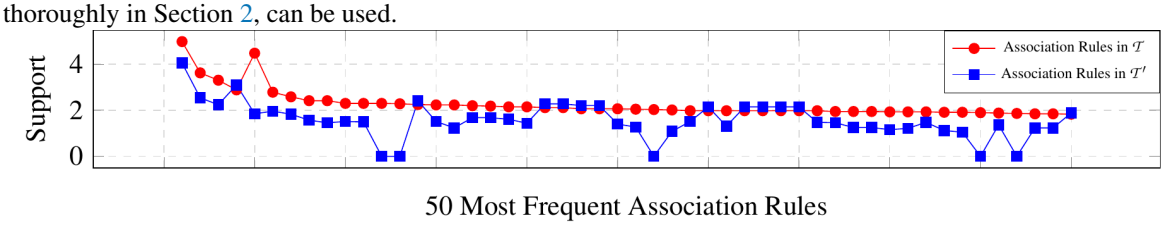


Figure 6. Frequent association rules mining (support $\geq$ 0.001).

6. Conclusion

Anonymization is challenging when data must be analyzed, studied and explored afterwards. Unstructured data complicate the anonymization process, since in a very general way data items may vary tremendously in structure and value without following a pattern. Set-valued data is a type of unstructured data representing multi-sets of sets. In this work, we are interested in publishing an anonymized set-valued dataset, ready for future accurate analysis. In this context, set-valued data is considered for isolated data, without temporality, and without semantics. This scenario is a generalization of specific data values, which can be considered for a broader investigation of set-value data. Disassociation is an anonymization technique for set-valued datasets, presented by [2]. It guarantees k^m-anonymity for associations without altering the value of the data items. It

proceeds by grouping the data records in different clusters and then vertically separate items of the records when k^m-anonymity is not verified, thus creating ambiguity on the level of associations for the items that are vertically separated in a cluster.

This paper is an extension of the work done in [1], where the utility-privacy trade-off in a disassociated dataset is studied deeply. The loss of associations for aggregate analysis is considered for the theoretical study. We came to the conclusion that the loss of the utility is directly linked to the clustering process of disassociation. Driven by this problem we propose in [1], UGAC, to drive the process of clustering for a set of records representing predefined utility rules.

As a continuity of the previous work, we tackle the problem of knowledge extraction in a disassociated dataset. We know from the aggregate analysis that it is hard to evaluate itemsets that are split on multiple record chunks. The problematic becomes: how can we re-associate the itemsets from the disassociated result while staying faithful to anonymization and knowledge extraction at the same time, in set-valued datasets?

To solve this problem, we define a general notion of similarity between datasets: neighbor datasets. Neighbor datasets are datasets that are not copies but synthetic representation, that leads to trustworthy data analysis. To standardize our notion of a neighborhood, we need a distance to assess it. Up to our knowledge, there exists no specific metric for set-valued data which is mathematically defined as a multiset of sets. Our first contribution in this work is the formalization of the datasets into trees and the use of tree edit distance that calculates the number of transformations needed to move from one tree to another. This way we are able to calculate a distance between two multisets of sets, also known as datasets. Our second contribution is an algorithm that intuitively generates neighbor datasets. We propose a probabilistic wheel re-association algorithm to generate a re-associatied dataset from the result of disassociation of the original dataset.

Finally, we test our utility guided ant-based clustering and probabilistic wheel re-association algorithms to evaluate their efficiency especially for knowledge extraction and data analysis in the context of general set-valued data. From the experiments, we can see that re-associated datasets create a neighborhood to the original dataset that depends on the privacy level imposed by disassociation. Despite the perturbation and noise added to the support of itemsets, probabilistic wheel re-association is able to generate synthetic datasets respecting the overall data representation of the itemsets and association rules in the data. This is extremely interesting for prediction and decision-making analysis, as statistical exploration will lead to a very close hypothesis. On another side, when applying the same privacy constraints for disassociating the original dataset, the probabilistic wheel re-association algorithm generates re-associated datasets almost at the same distance from the original dataset. This reflects the fact that our approach respects the privacy imposed through the process. We can conclude that probabilistic wheel re-association is a faithful algorithm for knowledge extraction and data analysis over anonymized set-valued datasets. The preservation of predefined utility rules is necessary when we want to ensure their representation over a threshold. We run a set of experiments with the utility guided ant-based clustering algorithm, UGAC, to see how well can it preserve the utility rules. UGAC is compared with the classical clustering technique, k-means, and normal horizontal disassociation for various properties of utility rules. The results show that UGAC, compared to the other two solutions, is able to decrease the information loss for the utility rules without increasing the information loss of other associations in the cluster. Combining the two solutions, we can say that

knowledge extraction and data analysis is exceptionally valid on anonymized datasets when transformed into neighbors to the original dataset.

In future work, we intend to use the Tree Edit Distance on the set of generated neighbor datasets, to decide which re-association is the most representative of the datasets for publication, being the centroid, or in other words the least distance to all the other datasets. We will investigate our approach with other anonymization techniques for publishing datasets and generalize the mathematical evaluation of the utility-privacy trade-off for future uses of the dataset in machine learning algorithms.

Author Contributions: "Conceptualization, Nancy Awad and Jean-Francois Couchot; methodology, Nancy Awad and Jean-Francois Couchot; software, Nancy Awad and Jean-Francois Couchot and Bechara Al Bouna and Laurent Philippe; validation, Jean-Francois Couchot and Bechara Al Bouna and Laurent Philippe; formal analysis, Nancy Awad and Jean-Francois Couchot; investigation, Nancy Awad and Jean-Francois Couchot; resources, Jean-Francois Couchot and Bechara Al Bouna and Laurent Philippe; writing—original draft preparation, Nancy Awad and Jean-Francois Couchot; writing—review and editing, Nancy Awad and Jean-Francois Couchot; resources, Jean-Francois Couchot and Bechara Al Bouna and Laurent Philippe; supervision, Jean-Francois Couchot and Bechara Al Bouna and Laurent Philippe. All authors have read and agreed to the published version of the manuscript."

Funding: :

This work is funded by the Labex ACTION program (contract ANR-11-LABX-01-01), by the Interreg RESponSE project, by the Hubert Curein CEDRE Programme number 40283YK, by the National Council for Scientific Research in Lebanon CNRS-L, and by the Antonine University. Computations have been performed on the supercomputer facilities of the Mésocentre de calcul de Franche-Comté.

Conflicts of Interest: "The authors declare no conflict of interest."

References

1. Awad, N.; Couchot, J.F.; Bouna, B.A.; Philippe, L. Ant-driven clustering for utility-aware disassociation of set-valued datasets. Proceedings of the 23rd International Database Applications & Engineering Symposium, Athens, Greece, 10–12 June 2019, p. 6. doi: 10.1145/3331076.3331084.
2. Terrovitis, M.; Mamoulis, N.; Liagouris, J.; Skiadopoulos, S. Privacy Preservation by Disassociation. Proc. VLDB Endow. **2012**, 5, 944–955. doi:10.14778/2336664.2336668.
3. Dwork, C. Differential privacy. Encyclopedia of Cryptography and Security **2011**, pp. 338–340, doi:10.1007/978-1-4419-5906-5_752.
4. Samarati, P. Protecting Respondents' Identities in Microdata Release. IEEE Trans. Knowl. Data Eng. **2001**, 13, 1010–1027.
5. Sweeney, L. k-Anonymity: a Model for Protecting Privacy. Int. J. Uncertain. Fuzzy. Knowl. Syst. **2002**, 10, 557–570. doi: 10.1142/S0218488502001636.
6. Xiao, X.; Tao, Y. Anatomy: Simple and effective Privacy Preservation. In Proceedings of 32nd International Conference on Very Large Data Bases (VLDB 2006), Seoul, Korea, September 12–15, 2006.
7. Li, T.; Li, N.; Zhang, J.; Molloy, I. Slicing: A New Approach for Privacy Preserving Data Publishing. IEEE Trans. Knowl. Data Eng. **2012**, 24, 561–574.
8. De Capitani di Vimercati, S.; Foresti, S.; Jajodia, S.; Livraga, G.; Paraboschi, S.; Samarati, P. Extending Loose Associations to Multiple Fragments. Data and Applications Security and Privacy; Springer-Verlag: Berlin, Heidelberg, 2013; pp. 1–16. doi:10.1007/978-3-642-39256-6_1.
9. Wang, K.; Wang, P.; Fu, A.W.; Wong, R.C.W. Generalized bucketization scheme for flexible privacy settings. Inf. Sci. **2016**, 348, 377–393.
10. Wang, J.; Deng, C.; Li, X. Two Privacy-Preserving Approaches for Publishing Transactional Data Streams. IEEE Access **2018**, 6, pp. 23648 –23658. doi:10.1109/ACCESS.2018.2814622.

11. Vattani, A. The hardness of k-means clustering in the plane. Available online: https://cseweb.ucsd.edu/~avattani/papers/kmeans_hardness.pdf (accessed on 16 April 2020).
12. Brucker, P. On the complexity of clustering problems. Optimization and Operations Research; Springer: Berlin/Heidelberg, Germany, 1978; pp. 45–54.
13. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. Comput. Geosci. **1984**, 10, 191–203.
14. MacQueen, J.; others. Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Berkeley, CA, USA, 21 June– 18 July 1967, Vol. 1, pp. 281–297.
15. Dorigo, M.; Maniezzo, V.; Colorni, A.; others. Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. man cybern. **1996**, 26, 29–41.
16. Kennedy, J. Particle swarm optimization. Encyclopedia of machine learning **2010**, pp. 760–766, doi:10.1007/978-0-387-30164-8_630.
17. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **2007**, 39, 459–471.
18. Deneubourg, J.L.; Goss, S.; Franks, N.; Sendova-Franks, A.; Detrain, C.; Chrétien, L. The dynamics of collective sorting robot-like ants and ant-like robots. Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats, Paris, France, 24–28 September 1990, pp. 356–363.
19. Sadeghi, Z.; Teshnehlab, M. Ant colony clustering by expert ants. 2008 11th International Conference on Computer and Information Technology, Khulna, Bangladesh, 24–27 December 2008, pp. 94–100.
20. Moradi, P.; Rostami, M. Integration of graph clustering with ant colony optimization for feature selection. Knowl. Syst. **2015**, 84, 144–161.
21. Labroche, N.; Monmarché, N.; Venturini, G. A new clustering algorithm based on the chemical recognition system of ants. In Proceedings of the Proceedings of the 15th Eureopean Conference on Artificial Intelligence, Lyon, France, 21–26 July 2002, pp. 345–349.
22. Dorigo, M.; Birattari, M. Ant colony optimization, IEEE Computat. Intell. Mag., **2010**, 4, 28–39. doi: 10.1109/MCI.2006.329691.
23. Pawlik, M.; Augsten, N. Efficient computation of the tree edit distance. ACM Trans. Database Syst. **2015**, 40, 3. doi: 10.1145/2699485.
24. Pawlik, M.; Augsten, N. Tree edit distance: Robust and memory-efficient. Inf. Syst. **2016**, 56, 157–173.
25. Hornik, K.; Grün, B.; Hahsler, M. arules-A computational environment for mining association rules and frequent item sets. J. Stat. Softw. **2005**, 14, 1–25.
26. Hahsler, M.; Grün, B.; Hornik, K. Introduction to arules–mining association rules and frequent item sets. SIGKDD Explor. **2007**, 2, 1–28.