Privacy Preservation In Public Graph By Clustering Techniques

1. R. VALARMATHY , 2. Dr. U. DURAI

1. Research Scholar, Dept. Of Computer Science, Marudupandiayar College of Arts and Science, Thanjavur.

2. Research Advisor, Dept. Of Computer Science, Marudupandiayar College of Arts and Science, Thanjavur.

(Affiliated to Bharathidasan University, Tiruchirappalli)

Abstract

Graph data created through users sharing for corporate analytics and social science research provides a unique challenge for privacy protection in the current context. Medical records, business data, and so on all contain analytical values, the results of which are free of definite qualities and as clean as possible. Data sanitization, privacy protection, and data publication are only a few of the methods proposed in the data mining sector. Recent years have seen an increase in study of k-anonymization methods. The approaches must be able to provide data anonymization while also limiting the information loss brought on by changes to the data. This work proposes clustering as a means to guarantee high data quality with little loss of information. The primary goal of storage is to maintain records of data that are very comparable to each other and should belong to the same equivalence class. Using this method, we may express an issue in the form of a K-member clustering problem, where the members of the cluster are known in advance.

Keywords: Privacy Preservation, public graph, Clustering Techniques, k-anonymization, k-member clustering.

Introduction

The most up-to-date method for protecting user information is called k-anonymity. These methods protect the confidentiality of publicly available information by excluding from the collection of quasi-identifier characteristics any record that is identical to (K-1) other records. K-anonymity is a method for solving the K-anonymity problem with minimal computer complexity. Our approach primarily aims to solve a new K-anonymization issue that characterizes the constraints. Our method centers on solving the K-anonymization issue, which is analogous to the clustering problem. A natural transformation into clustering occurs when the need for K-anonymity is considered. The goal here is to identify a collection of clusters, each of which has the potential to hold K+ records. On the other hand, the cluster's records are identical to each other in order to raise both data quality and quasi-identifier value. The need
for distortion is reduced when cluster records are updated. This is to communicate the k-member clustering issue, a particular type of clustering problem. The greedy technique used here takes time complexity O(n2) to solve, placing this issue in the category of non-deterministic polynomial-time hardness (NP). While this method does not rely on trust or generalization orders, it can take use of the fact that there may be underlying relationships between the variables in a domain to arrive at better outcomes. Therefore, quality criteria for representing information loss through hierarchy-free generalization have been proposed but have not been deployed. Therefore, the hierarchy-free generalization makes use of the data quality meter, also called the information loss metric, to assess the data's overall quality.

**Concepts of Anonymity model**

In order to ensure that no part of the microdata is omitted or generalized, the anonymity model is often considered to be the best option. K-anonymity models are based on the idea that no one should be able to create strong connections between the data in a given table and the corresponding entities. Any one record that is similar to the other (K-1) records associated with the set of characteristics is called a Quasi-identifier, and it is necessary for the K-anonymity model to use them all to achieve the aim. An equivalence class is defined as the set of records that are equivalent to one another.

**Table.1 Raw Medical Data Set**

<table>
<thead>
<tr>
<th>I</th>
<th>QI</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
</tr>
<tr>
<td>Bill</td>
<td>M</td>
<td>20</td>
</tr>
<tr>
<td>Ken</td>
<td>M</td>
<td>24</td>
</tr>
<tr>
<td>Linda</td>
<td>F</td>
<td>26</td>
</tr>
<tr>
<td>Mary</td>
<td>F</td>
<td>28</td>
</tr>
</tbody>
</table>

**Table. 2 A 2 Anonymous Data Set of Table.1**

<table>
<thead>
<tr>
<th>I</th>
<th>QI</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Sex</td>
<td>Age</td>
</tr>
<tr>
<td>Bill</td>
<td>M</td>
<td>[20,24]</td>
</tr>
<tr>
<td>Ken</td>
<td>M</td>
<td>[20,24]</td>
</tr>
</tbody>
</table>
Private data on the population is included in Table.1. There are several ways to categories the data in the table:

1. Identifiers (I),
2. Quasi-Identifiers (QI),
3. Sensitive Attribute (SA)

Age, ZIP code, and sex attribute data from QI make up the raw medical data collection (Table.1). Patient records 1 and 2 are easily re-identifiable due to the fact that each patient's age, zip code, and gender are recorded in a separate field in Table 1 of the A 2 anonymous dataset (Table.2). Two patients are less likely to be mistakenly matched up because to these tables [2].

**Related works**

When working with quasi-identification characteristics in large data, anonymization techniques are used to ensure confidentiality. This work presents a novel k-anonymity method that does not require a predetermined value for k as a threshold [3]. Concerns about privacy associated with quasi-identifiers and sensitive features are often addressed by the application of anonymization methods. Researchers have recently identified some methods as unsuitable because to the presence of quasi-identifying features in the dataset, including sex, age, and date of birth [4]. By "quasi-identifier," we mean a group of characteristics that, when combined with other data, may be used to re-identify objects [5]. The concept of k-anonymity states that if a tuple in a publicly available dataset is identical to k-1 other tuples in the dataset, then it should be considered anonymous. Therefore, an intruder who discovers the values of a person's quasi-identifier but is unable to differentiate them from (k-1) other data [6]. To protect privacy, K-anonymity makes use of techniques like suppression and generalization [7]. In [8], Zhen Tu et al. present a generalization strategy that guarantees minimal Spatio-temporal granularity loss while maintaining k-anonymity, t-closeness, and l-diversity of tracks. A utility-driven adaptive clustering approach has been presented with comparable best data quality to partition tuples [9], and its anonymization method is able to keep personal privacy requirements intact. This study looked into how k-anonymity might affect data mining's predictive power. Using both anonymous and non-anonymous data, a Nave Bayes classifier is computed, with increased anonymity corresponding to a proportionate decrease in classifier performance [10]. Hadoop's MapReduce programming paradigm was utilized to implement a parallel k-anonymity method, and both the background knowledge attack and the homogeneity attack models were studied.

The suggested approach may provide anonymized large data, which can then be used to back up privacy-preserving data mining [11]. Preventing information leaks related to k-anonymity techniques is the primary goal of this research. We'll use decision tree classification to evaluate the anonymized data set in terms of privacy and mining quality, and we'll compare the results.
to those obtained using other data mining methods (support vector machine and logistic regression) [12]. K-anonymization is used to protect against these kinds of assaults by making minor adjustments to microdata that might result in a larger dataset [13]. The author protects sensitive information from being inferred by strong implications [15] by extending the k-anonymity model to the (, k)-anonymity model [14]. Data generalization over several dimensions simultaneously constitutes the idea of multi-dimensional k-anonymity [17][16], and [18] expands the multi-dimensional generalization to anonymize data for a specific goal, such as grouping.

**Anonymization Algorithm**

Potentially the best approach to solving the clustering problem may be found in k-member clustering. The following is an attempt to define the computational complexity problem, which is best described as a judgement problem.

**Theorem 1.** The k-member clustering decision problem is Nondeterministic Polynomial time (NP)-complete.

**Proof.** Initial cluster e1 is formed by randomly selecting ri record from a collection of n records. The IL (e1 rj) record becomes minimal as time passes. This continues until |e1| equals k. The clustering procedure is continued until there are less than k records remaining or |e1| = k, at which point a record is the farthest away from ri and. Following the completion of the iteration process, records are placed into each cluster so that the amount of information lost is as small as possible.

**Theorem 2.** Let’s say 'k' stands for the anonymity parameter you've set, and 'n' is the total number of records you're feeding in. As part of a greedy k-member method, each group must locate exactly k, but no more than 2k-1, records.

**Proof.** S must stand for the collection of input records. Each cluster has precisely k records, hence there must be at least k records left over. In the worst-case scenario, k 1 more records are added to a cluster that already holds k total records. As a result, 2k+1 is the largest cluster possible.

**Theorem 3.** Let represents the total numbers of input records and represents the specified anonymity parameter. The time complexity of the greedy k-member clustering algorithm is in O (n²).

**Function greedy_k_member_clustering (S, k)**

**Input:** a set of records S and a threshold value k.

**Output:** a set of clusters each of which contains at least k records.

1. if (|S| | k)
2. return S;
3. end if;
4. result = ∅; r = a randomly picked record from S;
5. while (|S| ≥k)
6. r = the furthest record from r;

http://www.webology.org
7. $S = S - \{r\}$
8. $c = \{r\}$
9. while ($|c| < k$)
10. $r = \text{find\_best\_record}(S, c)$
11. $S = S - \{r\}$
12. $c = c \cup \{r\}$
13. end while;
14. result = result $\cup \{c\}$
15. end while;
16. while ($|S| \neq 0$)
17. $r =$ a randomly picked record from $S$
18. $S = S - \{r\}$
19. $c = \text{find\_best\_cluster}(\text{result}, r)$
20. $c = c \cup \{r\}$
21. end while;
22. return result;
End;

Function \text{finds\_best\_record}(S, c)
\textbf{Input}: a set of records $S$ and a cluster $c$.
\textbf{Output}: a record $r \in S$ such that $IL(c \emptyset \{r\})$ is minimal.
1. $n = |S|; \min = \infty; \text{best} = \text{null}$
2. for($i = 1, \ldots n$)
3. $r =$ $i$-th record in $S$
4. $\text{diff} = IL(c \cup \{r\}) - IL(c)$
5. if( $\text{diff} < \min$ )
6. $\min = \text{diff}$
7. $\text{best} = r$
8. end if;
9. end for;
10. return $\text{best}$;
End;

Function \text{find\_best\_cluster}(C, r)
\textbf{Input}: a set of clusters $C$ and a record $r$.
\textbf{Output}: a cluster $c \emptyset C$ such that $IL(c \emptyset \{r\})$ is minimal.
1. $n = |C|; \min = \infty; \text{best} = \text{null}$
2. for($i = 1, \ldots n$)
3. $c =$ $i$-th cluster in $C$
4. $\text{diff} = IL(c \cup \{r\}) - IL(c)$
5. if( $\text{diff} < \min$ )
5. min = diff;
6. best = c;
7. end if;
8. end for;
9. return best;
End;

**Experimental Results**
The tests aim to examine how well the suggested method performs in terms of data quality, efficiency, and scalability. So that we may compare greedy k-member clustering with another method, specifically the median partitioning technique, and draw some conclusions about its efficacy.

**Data Quality and Efficiency**
Here we present experimental findings for the greedy k-members method, measuring its performance in terms of data quality and speed of execution. Total-IL costs for three methods (median partitioning, greedy k-member, and greedy k-member modified to decrease classification error) are shown for different values of k in Figure.1.

![Figure.1 Information Loss Metric](http://www.webology.org)

As shown in Figure.1, the total-IL is minimized using the greedy k-members approach regardless of the value of k. Total-IL cost comparison between the modified greedy k-member and the original method shows that they are almost identical. As a result of simply considering
closeness among data points with respect to one dimension at each partitioning, the median partitioning method is inferior to the presented techniques. The size of each equivalence class is taken into account by the Discernibility Metric (DM), another measure of data quality. It is intuitive that data quality would decrease as the number of similar records increases; DM does a good job of capturing this effect of the k-anonymization process by displaying the DM costs of the three techniques for different k values. The median partitioning algorithm illustrated in the picture has less performance than the two greedy k-member techniques. While due to clustering, greedy k-member algorithms consistently generate equivalence classes with sizes near the chosen k.

Conclusion

By recasting the K-anonymity problem as a K-member clustering problem, this research suggests a practical technique for K-anonymization. Cost and distance, the two most prominent criteria proposed for clustering, are correctly represented for the K-anonymization problem. The suggested method draws attention to the fact that generalization introduces distortion into the data for both the IL measure and the cost metric. It's low enough in volume that it may be used as a quality metric for K-anonymized data.

Reference


