A (*k*, *p*)-anonymity Framework to Sanitize Transactional Database with Personalized Sensitivity

Binbin Zhang[1,2], Jerry Chun-Wei Lin[3,4], Qiankun Liu[3], Philippe Fournier-Viger[5], Youcef Djenouri[6]

[1] Department of Biochemistry and Molecular Biology, Shenzhen University Health Science Center, China
[2] Center for Anti-aging and Regenerative Medicine, Shenzhen University Health Science Center, China
[3] School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China
[4] Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, Norway
[5] School of Humanities and Social Sciences, Harbin Institute of Technology (Shenzhen), China
[6] Department of Computer Science, NTNU, Norway
zhangbb@szu.edu.cn, jerrylin@ieee.org, mcqueenqkliu@gmail.com, philfv@hitsz.edu.cn, djenouri@imada.sdu.dk

Abstract

In recent years, analyzing transactional data has become an important data analytic task since it can discover important information in several domains, for recommendation, prediction, and personalization. Nonetheless, transactional data sometimes contains sensitive and confidential information such as personal identifiers, information aboutsexual orientations, medical diseases, and religious beliefs. Such information can be analyzed using various data mining algorithms, which may cause security threats to individuals. Several algorithms were proposed to hide sensitive information in databases but most of them assume that sensitive information is the same for all users, which is an unrealistic assumption. Hence, this paper presents a (*k*, *p*)-anonymity framework to hide personal sensitive information. The developed ANonymity for Transactional database (ANT) algorithm can hide multiple pieces of sensitive information in transactions. Besides, it let users assign sensitivity values to indicate how sensitive each piece of information is. The designed anonymity algorithm ensures that the percentage of anonymized data does not exceed a predefined maximum sensitivity threshold. Results of several experiments indicate that the proposed algorithm outperforms the-state-of-the-art PTA and Gray-TSP algorithms in terms of information loss and runtime.

Keywords: Anonymization, Cluster, Multiple sensitive information, Hierarchical attributes

1 Introduction

Data mining techniques are often used to uncover hidden relationships between items in transactional data [1, 8-9, 12-13, 15-16]. For specific applications such as analyzing medical records, each record may contain sensitive/confidential information, such as personal identifiers, information about sexual orientation, and information about medical diseases. The presence of such information in transactional data can lead to the disclosure of sensitive private information. This may result in serious security threats if the information falls into the wrong hands. Besides, some attackers may infer private and sensitive information from a database if the non-sensitive information in each record is not identical.

Privacy-preserving data mining [20, 24, 27] has thus become a key research area of data analytics, which consists of sanitizing a database to protect private information and ensure security against privacy threats. The *k*-anonymity concept [20] has been proposed to ensure that at least (*k*-1) transactions have the same values for sensitive attributes, so that at least (*k*-1) transactions are indistinguishable from each other. Nonetheless, even by applying this concept, a disclosure of private information can occur when transactions from a same equivalence class contain similar sensitive information. The concept of *l*-diversity [19] was introduced to ensure that the number of transactions in each equivalence class is at least (*l*). Hence, the probability that an attacker infers a user's sensitive information is less than 1/*l*.

However, *l*-diversity does not provide a solution to the attribute disclosure problem of *k*-anonymity. For example, when sensitive information of equivalence classes consists of multiple attributes, sensitive information can still be inferred under *l*-diversity model. The most existing algorithms treat all the attributes having the same sensitivity for anonymity. However, different users may have different needs in terms of privacy. Each attribute may be treated with varied sensitivity to different users. Thus, the anonymity progress should achieve not only (*k*-1) transactions become indistinguishable but also the

*Corresponding Author: Jerry Chun-Wei Lin; E-mail: jerrylin@ieee.org
DOI: 10.3966/160792642019052003013

personalized sensitive information could be successfully hidden under the maximum (p) sensitivity threshold. Motivated by the above problems, this paper introduces a novel (k, p)-anonymity framework and an effective anonymity algorithm called ANonymity for Transactional database (ANT) algorithm for transactional data, which can be used to solve both the problems of k-anonymity and l-diversity. In the designed ANT algorithm, users can define varied sensitive degrees of the attributes in transactional database, which is more applicable to real-life situation. Substantial experiments indicate that the better performance of the proposed algorithm can be obtained compared to the state-of-the-art anonymity algorithms in terms of information loss and runtime.

2 Literature Review

The k-anonymity concept is widely used to protect personal privacy [20] against malicious attacks [22-23] in relational databases. It utilizes quasi-identifier attributes such as birth date and postcode to define a generalized hierarchy in a specific domain and generate equivalence classes. This approach ensures that at least (k-1) records become indistinguishable and that sensitive information is thus hidden. Many techniques such as generalization [28], suppression [17, 25], clustering [3, 21] and perturbation [24] were studied as methods to provide k-anonymity. Wang et al. [26] proposed a novel utility metric and designed a k-anonymity framework for graph models. The proposed framework uses a variety of utility metrics to achieve k-anonymity for a given social network and reduce utility loss. Doka et al. [5] defined the k-anonymity problem of maximal-utility and generalized it as a network flow problem. An algorithm was then presented to provide privacy protection for syntactic data and fully exploit the potential of heterogeneity.

Most anonymity algorithms are difficult to apply to transactional data since each transactional record sometimes contains unstructured data. Xu et al. [29] proposed the concept of (h, k, p)-coherence for the anonymization of transactional data. The (h, k, p)-coherence algorithm adopts a greedy approach to protect sensitive user information. Hsu et al. [10] proposed a k-anonymity algorithm for multi-patterns. It uses a combinational approach to protect user data from being re-identified and ensure that each frequent pattern meets the requirement of k-anonymity. Xue et al. [30] presented a k-anonymity algorithm that transforms transactional data into a binary format. Ghinita et al. [7] proposed an approach that takes the relevance of customer purchased goods into account to ensure that sensitive items cannot be inferred from non-sensitive information. Lin et al. [14] presented an effective and efficient algorithm to anonymize transactional data.

Nevertheless, using the above algorithms based on the k-anonymity concept can still result in privacy leaks since users can still be identified. Although (k-1) records (transactions) become indistinguishable in an equivalence class, sensitive information of these users can still be inferred. Machanavajjhala et al. [19] proposed two simple attacks and pointed out that k-anonymity does not protect against these attacks. Zhou and Pei [31] extended the k-anonymity and l-diversity concepts to social network data for protecting against attacks from neighbors. Li et al. [11] then developed a novel t-closeness privacy protection technique for anonymity. It requires that the distribution of sensitive attributes in any equivalence class is close to that of the distribution of the attribute in the entire dataset. Thus, the distance between any two distributions cannot exceed a given threshold t, and anonymity with respect to the sensitive information is achieved.

3 Preliminaries and Problem Statement

A transactional dataset with n transactions is denoted as $D = \{T_1, T_2, \ldots, T_n\}$, where each transaction in D is denoted as T_q. Items in D are denoted as $I = \{I_1, I_2, \ldots, I_r\}$. A transaction T_q is a collection of a finite number of items from I. A personalized transactional dataset is shown in Table 1 where personalized sensitive information is considered, that is where each item in a transaction has a sensitivity degree indicating how sensitive this item is for the user. For example, a male user may consider that items representing the marriage status are more sensitive (90%) than a female user does (30%). In this example, the maximum sensitivity threshold initially is set to 0.8.

Table 1. A personalized transactional database

TID	Items with their sensitiviety degrees
1	(1, 0.9); (3, 0.6); (4, 0.1); (5, 0.3); (7, 0.3); (11, 0.4)
2	(3, 0.3); (4, 0.2); (7, 0.7); (8, 0.2); (12, 0.7)
3	(3, 0.6); (4, 0.2); (8, 0.5); (9, 0.1)
4	(2, 0.3); (4, 0.2); (5, 0.6); (11, 0.8); (12, 0.9)
5	(2, 0.4); (5, 0.1); (7, 0.3); (10, 0.3); (12, 0.9)
6	(1, 0.1); (5, 0.4); (6, 0.5); (11, 0.8)
7	(2, 0.4); (4, 0.7); (5, 0.1); (7, 0.2); (11, 0.8)
8	(2, 0.2); (5, 0.2); (9, 0.5); (10, 0.1)
9	(1, 0.3); (3, 0.7); (4, 0.6); (6, 0.4); (8, 0.2); (12, 0.8)
10	(1, 0.2); (5, 0.2); (6, 0.3); (8, 0.5)
11	(3, 0.1); (5, 0.3); (12, 0.8)
12	(1, 0.2); (5, 0.3); (9, 0.4)

In Table 1, each item is denoted by an integer and is associated with a sensitivity degree in the [0,1] interval. An item is considered as a sensitive item that needs to be hidden if its sensitivity is no less than a given maximum sensitivity threshold. The traditional l-diversity constraint requires to generate valid equivalence classes each containing at least (l) different sensitive items to avoid the disclosure problem. But this constraint may be difficult to satisfy

in some situations. For example, when sensitive information is very sparse, it is impossible to guarantee that enough pieces of sensitive information can be successfully assigned to each equivalence class. Moreover, it still may lead to the disclosure of sensitive information if a transaction consists of multiple pieces of sensitive information (in that case the content of transactions or the identities of users may still be inferred). To address the above problems, we propose a novel (*k*, *p*)-anonymity framework with the ANT algorithm as follows.

Definition 1 (Equivalence class). Given a database *D*, and each transaction consists of several attributes, an equivalence class is the collection of the transactions, and each transaction consists of the same values.

For example, two transactions have the same attributes with the quantities as (*A*:1, *B*:1, *C*:3); they are identical to each other, and (*A*:1, *B*:1, *C*:3) is the equivalence class of those two transactions.

Definition 2. A database *D* is a *k*-anonymity database if any of (*k*-1) transactions are identical.

For example, if a database is 3-anonymity database, any of (3-1) transactions are identical to each other.

Definition 3 ((*k*, *p*)-anonymity). The (*k*, *p*)-anonymity framework requires that at least (*k*-1) transactions are identical in the anonymized database, and that in each valid equivalence class, sensitivity degree of the attributes should less than a given maximum sensitivity threshold (*p*).

For example, the maximum sensitivity value (*p*) is set as 0.8, and (*k*) is set as 3. In a 3-anonymity database, at least (3-1) transactions are identical, and the sensitivity of the attributes in the transaction is less than 0.8.

During the anonymity progress, a database must be modified to achieve (*k*, *p*)-anonymity. This process may cause information loss (*IL*). If the *IL* is low, it indicates that the anonymized database has higher similarity to the original database.

Definition 4 (Information loss, *IL*). The information loss (*IL*) is the number of different items between the original dataset *D* and the anonymized dataset *D*'.

Problem Statement: The proposed (*k*, *p*)-anonymity framework requires to sanitize a database to obtain (*k*-1) identical (indistinguishable) transactions under a maximum sensitivity threshold (*p*). Besides, *IL* should be as small as possible when comparing a sanitized database with its corresponding original database.

4 The Proposed (*k*, *p*)-anonymity Framework

To solve the above problem, a (*k*, *p*)-anonymity framework is designed. The developed framework consists of three steps, which are (1) **data pre-processing**; (2) **clustering**; and (3) **anonymization**. Details of three steps are given as follows.

4.1 Pre-processing Step

To handle the transactional database into the anonymity problem, the database is then first processed, and the attributes are organized under hierarchical attributes, as shown in Table 2. In the running example, each hierarchical attribute is denoted by an upper-case letter, and represents a generalization of one or more items. For example, "milk" and "coffee" are two different products that can be generalized as a hierarchical attribute "drink". The reason for using hierarchical attributes is that the attackers usually do not have details about the purchased items but are only interested in general attributes.

Table 2. Hierarchical attributes with their items

Hierarchical attribute	Items
A	1, 7, 11
B	2, 9
C	3, 6
D	4, 10
E	5, 8
F	12

Notice that the generalized attributes can be defined by users' preference. Hence, a transactional database can be transformed into a mapped dataset, as shown in Table 3. In that table, only the items having a sensitivity degree no less than the maximum sensitivity threshold (0.8, in this example) will be used by the anonymization process. A mapped database is shown in Table 3.

Table 3. A mapped dataset under hierarchical attribute

TID	Non-sensitive attributes	Sensitive attribute, item, sensitivity
1	*A*:3; *C*:1; *D*:1; *E*:1	*A*:1:0.9
2	*A*:1; *C*:1; *D*:1; *E*:1; *F*:1	-
3	*B*:1; *C*:1; *D*:1; *E*:1	-
4	*B*:1; *D*:1; *E*:1	*A*:11:0.8; *F*:12:0.9
5	*A*:1; *B*:1; *D*:1; *E*:1	*F*:12:0.9
6	*A*:2; *C*:1; *E*:1	*A*:11:0.8
7	*A*:1; *B*:1; *D*:1; *E*:1	*A*:11:0.8
8	*B*:2; *D*:1; *E*:1	-
9	*A*:1; *C*:2; *D*:1; *E*:1	*F*:12:0.8
10	*A*:1; *C*:1; *E*:2	-
11	*C*:1; *E*:1	*F*:12:0.8
12	*A*:1; *B*:1; *E*:1	-

In Table 3, each transaction contains non-sensitive hierarchical attributes and sensitive items of the original dataset, in which each non-sensitive hierarchical attribute is associated with an occurrence frequency of the attribute in each transaction, and sensitive information consists of the mapped attributes, original items, and the sensitivity values.

For example of T_1 in Table 1, the attribute (1) is then transformed as the attribute (*A*) according to the generalized table shown in Table 2. For transaction T_1

in Table 1, three attributes such as (1), (7) and (11) are generalized as (*A*:3) but in T_1, the sensitivity of (1) was defined as (0.9), thus only one attribute (*A*) is required to be hidden since the maximum sensitivity threshold is set as (0.8).

To group transactions with high similarity for the purpose of anonymization, the mapped table is then transformed into a matrix representation. For example, Table 3 is transformed into Table 4.

Table 4. Matrix representation of the mapped database

TID	Non-sensitive attribute				Sensitive attribute			
	A	*B*	*C*	*D*	*E*	*F*	*A*	*F*
1	2	0	1	1	1	0	1	0
2	1	0	1	1	1	1	0	0
3	0	1	1	1	0	0	0	0
4	0	1	0	1	1	0	1	1
5	1	1	0	1	1	0	0	1
6	2	0	1	0	1	0	1	0
7	1	1	0	1	1	0	1	0
8	0	2	0	1	1	0	0	0
9	1	0	2	1	1	0	0	1
10	1	0	1	0	2	0	0	0
11	0	0	1	0	1	0	0	1
12	1	1	0	0	1	0	0	0

Thus, the transactional database is then transformed into a relational one, where each attribute is associated with a value for later similarity calculation. The clustering step is then performed using that transformed matrix for anonymization.

4.2 Clustering Step

The purpose of the designed framework is to ensure that transactions become indistinguishable from each other, thus hiding sensitive information and protecting user privacy. Reducing information loss (*IL*) is the main difficulty for anonymizing a database. Thus, non-sensitive transactions with high similarity are assigned to the same cluster, which can be used to reduce the number of modifications of the original dataset.

In this step, the *k*-means [18] algorithm is applied to group highly similar transactions into the same cluster. For example, we assume that the *k* is initially set as 2. We randomly select transactions as the initial cluster center, for example, T_1 and T_3 are respectively defined as the centers of two clusters. For the transaction T_2, the similarity of T_2 is then calculated between T_1 and T_3. In this example, T_1 is encoded as (201110), T_3 $ is encoded as (011110), and T_2 is encoded as (101111). The distances between T_2 to T_1 and to T_3 are respectively calculated as: (|1-2| + |0-0| + |1-1| + |1-1| + |1-1| + |1-0|)(= 2), and (|1-0| + |0-1| + |1-1| + |1-1| + |1-1| + |1-0|)(= 3); T_2 is more similar to T_1 and will be put into the same cluster with T_1. The resting transactions are processed as the same progress. This iteration is recursively processed until the cluster center is found and unchanged. The results are shown in Table 5.

Table 5. Generated clusters for the running example

Cluster ID	TIDs	Cluster center
C_1	1, 2, 6, 9, 10, 11	1 0 1 1 1 0
C_2	3, 4, 5, 7, 8, 12	1 1 0 1 1 0

In Table 5, six transactions such as T_1, T_2, T_6, T_9, T_{10}, and T_{11} are grouped into the same cluster, and the other six transactions such as T_3, T_4, T_5, T_7, T_8, T_{12} are then put into another cluster. By averaging each transaction within the same cluster, the cluster centers of two groups can be also calculated as (1 0 1 1 0) and (1 1 0 1 1 0), respectively.

4.3 Anonymization Step

In the clustering step, the dataset is divided into several clusters such that transactions in a same cluster are highly similar to each other. To achieve (*k*, *p*)-anonymity and minimize information loss (*IL*) in a sanitized database, it is necessary to assign the most similar transactions to the same group and generate valid equivalence classes. Transactions in each equivalence class are then anonymized. To avoid the security issues of *k*-anonymity and traditional *l*-diversity, it is necessary to consider the distribution of sensitive items when generating equivalence classes. The pseudo-code the proposed ANT is provided in Algorithm 1.

Algorithm 1. Proposed ANT algorithm

INPUT: c_k_set, a set of *k* clusters; *L*, the anonymity degree; δ, maximum sensitivity threshold.
OUTPUT: *EC_set*, a set of equivalence class.

```
1.  set q := 1;
2.  while |c_k_set| > L do
3.    for each c_i in c_k_set do
4.      tmp.center := argmin{distance(c_k_set.center, c_i)};
5.      EC_q ← EC_q ∪ tmp.center;
6.      remove tmp.center from c_k_set;
7.    sort c_i ∈ c_k_set;                    /*distance to tmp.center */
8.      Judge (EC_q, c_i, δ);
9.    end for
10. end while
11. Judge (EC_q, c_i, δ) {
12. tmp_set := EC_q ∪ c_i ;
13. if tmp_set.sensitivity >δ do    /*calculate sensitivity*/
14.   q++;
15.   EC_q ← EC_q ∪ c_i;
16. end if
17. }
18. return EC_set;                          /*the set of EC_q*/
```

First, the closest transaction to the non-sensitive part of the cluster center is found (Line 4), and it is set as the equivalence class (Line 5). After that, this transaction is removed from the cluster (Line 6). The remaining transactions of the cluster are then sorted by ascending order of distance to the cluster center (Line 7). The closest transaction to the cluster center is then examined to decide whether it can be assigned to the equivalence class (Line 8, Lines 11 to 17). After that, the (*k*, *p*)-anonymity requirement is attained. The running example of the designed algorithm is shown in Table 6.

Table 6. The equivalence classes generated by the anonymity step

Equivalence class	TIDs	Non-sensitive cluster center
EC_1	T_1, T_2, T_9	1 0 1 1 1 0
EC_2	T_6, T_{10}, T_{11}	1 0 1 0 1 0
EC_3	T_3, T_4, T_8	0 1 0 1 1 0
EC_4	T_5, T_7, T_{12}	1 1 0 1 1 0

For example, assumes that (3, 60%)-anonymity is desired. Consider cluster C_1 in Table 5. The distances of T_1, T_2, T_6, T_9, T_{10}, and T_{11} to the cluster center (1 0 1 1 1 0) are respectively (1 1 2 2 2 2). The transaction T_1 is then set as the cluster center of the equivalence class EC_1, and T_1 is removed from C_1. The distances of the remaining five transactions to T_1 are respectively (2 1 2 3 3). Hence, the transactions in C_1 are sorted as T_6, T_2, T_9, T_{10}, and T_{11} using the ascending order of distance to T_1. In this example, T_6 and T_1 have the same sensitive hierarchical attribute of (*A*). If T_6 is added to EC_1, the sensitivity of the sensitive attribute (*A*) for a 3-diversity equivalence class is calculated as 66%, which exceeds the given maximum sensitivity threshold of 60%. Thus, T_6 is not assigned to EC_1. After that, T_2 and T_6 are grouped in EC_1, and a new cluster center is generated by considering the non-sensitive data by averaging the values of the transactions. The final result is the anonymized dataset shown in Table 7, where each equivalence class satisfies the (3, 60%)-anonymity condition.

Table 7. The final anonymized results of the running example

TID	Attribute with quantity	Attribute, item, sensitivity
1	*A*:1; *C*:1; *D*:1; *E*:1	*A*:1:0.9
2	*A*:1; *C*:1; *D*:1; *E*:1	-
3	*B*:1; *D*:1; *E*:1	-
4	*B*:1; *D*:1; *E*:1	*A*:11:0.8; *F*:12:0.9
5	*A*:1; *B*:1; *D*:1; *E*:1	*F*:12:0.9
6	*A*:2; *C*:1; *E*:1	*A*:11:0.8
7	*A*:1; *B*:1; *D*:1; *E*:1	*A*:11:0.8
8	*B*:2; *D*:1; *E*:1	-
9	*A*:1; *C*:2; *D*:1; *E*:1	*F*:12:0.8
10	*A*:1; *C*:1; *E*:2	-
11	*A*:1; *C*:1; *E*:1	*F*:12:0.8
12	*A*:1; *B*:1; *D*:1; *E*:1	-

5 Experimental Evaluation

In this section, extensive experiments are described to compare the effectiveness and efficiency of the proposed algorithm with the traditional Gray-TSP [30] and PTA [14] algorithms. It is important to notice that no prior work has considered anonymization using personalized sensitivity degree in transactional database. To assess the efficiency of the designed clustering approach, the proposed ANT algorithm without *k*-means clustering is called ANT-. Five real-life datasets (chess, mushroom, pumsb, connect and accidents) [6] and a synthetic dataset (T10I4D100K) [2] were used in the experiments to evaluate the performance of the compared algorithms.

5.1 Information Loss (IL)

In this section, information loss (*IL*) is compared for the four algorithms. The *IL* ratio is used to evaluate item differences between an original dataset and an anonymized dataset. The *IL* of four algorithms for various *k* values and a fixed cluster number are compared in Figure 2. Since the Gray-TSP and the PTA algorithms do not perform clustering, the number of segments used by those two algorithms is considered instead of the cluster number. The number of clusters for the datasets were respectively set to 60, 150, 800, 1250, 6,000, and 1,000 for chess, mushroom, pumsb, connect, accidents, and T10I4D100K, respectively. The maximum sensitivity threshold was set to 75% for all datasets.

It can be observed in Figure 2 that information loss of the four algorithms increases as the *k* value is increased for all datasets. The proposed ANT algorithm clearly outperforms the others since it produces less *IL*. ANT- algorithm provides better results than ANT in some cases such as in Figure 2(a), Figure 2(b), and Figure 2(d). The reason is that the anonymization process may face the problem of generating more "equivalence classes" of the ANT-. As *k* values are increased, more equivalence classes are generated. Therefore, it can be found that the *IL* of ANT decreases but that of ANT- increases as the cluster number is increased for those three datasets. For example, when *k* is greater than 20, the proposed ANT algorithm always outperforms ANT-. To summarize, we can see that if the size and cluster number is small, ANT- may outperform ANT.

5.2 Runtime

In this section, the runtime of the four algorithms are compared. The runtime of the developed ANT algorithm includes the pre-processing time, clustering time, and anonymization time, while the ANT- includes only the pre-processing and anonymization time. Similarity, the Gray-TSP and the PTA algorithms do not perform clustering. Thus, the number of segment

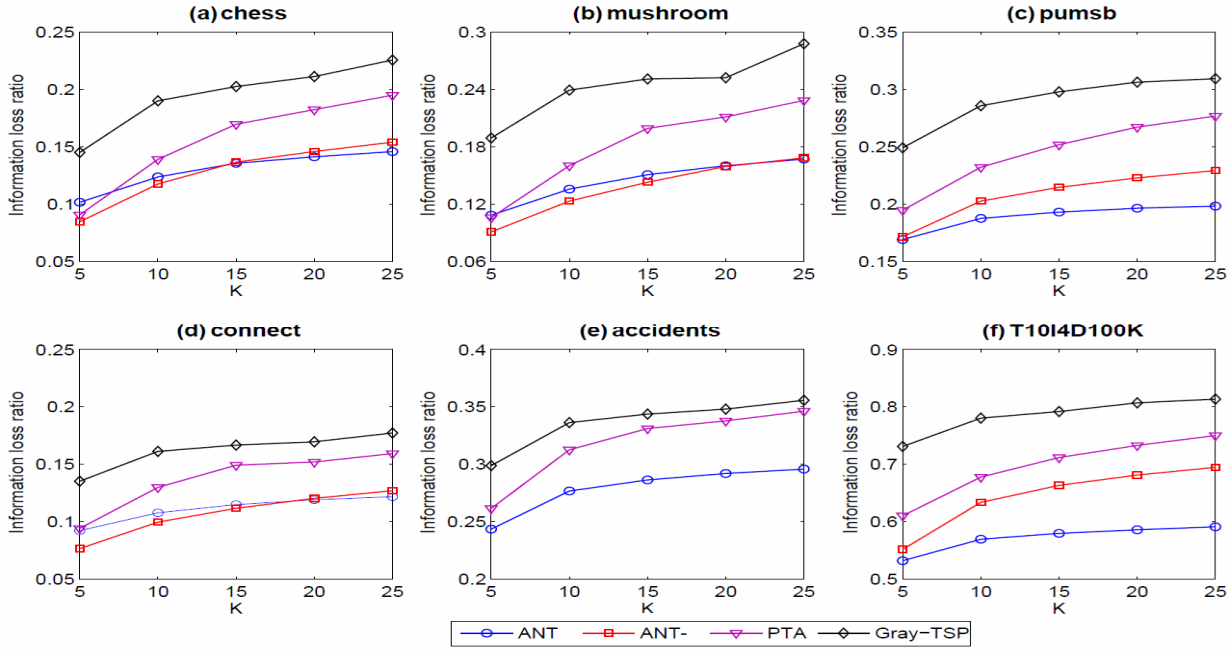

Figure 2. *IL* for a fixed cluster (segment) number and various *k* values

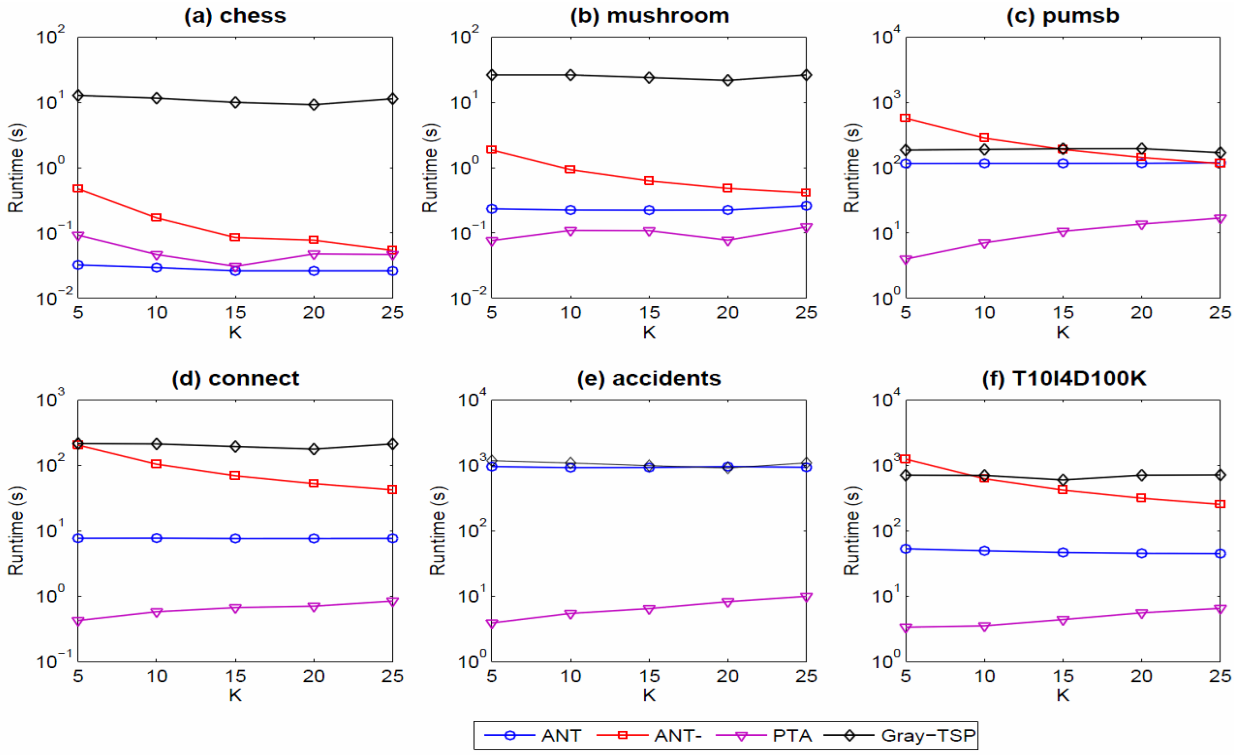

Figure 3. Runtime for a fixed *k* and various number of clusters (segments)

is used instead of the number of clusters for the designed algorithms. The results for a fixed cluster number are shown in Figure 3.

In Figure 3, it can be seen that the proposed ANT algorithm is always faster than Gray-TSP and ANT-. The runtime difference between the developed ANT algorithm and Gray-TSP is not obvious since those two algorithms utilize a shortest path process and a clustering process for anonymization. On the contrary, the runtime of ANT- obviously decreases as *k* is increased. This is because the cost of transaction similarity calculations decreases as *k* is increased. Since the number of segments is fixed, the runtime of PTA is stable, and its runtime increases along with *k*. Besides, if we only consider the search process of the shortest path and the anonymization process, PTA is only slightly faster than the proposed algorithms in some cases. However, the proposed ANT algorithm

achieves less IL, which is the major criteria to evaluate the effectiveness of the anonymity approach.

6 Conclusion

In this paper, we presented the (*k*, *p*)-anonymity framework with the ANonymity for Transactional database (ANT) algorithm to sanitize a database while considering different privacy needs of different users for multiple pieces of sensitive information. The (*k*, *p*)-anonymity framework first transforms an original dataset into a matrix representation. Then, the *k*-means clustering technique is applied to cluster highly similar data into the same clusters by considering their non-sensitive items. After that, the ANT algorithm is then performed, and the equivalence classes are generated using the transactions that satisfy the (*k*, *p*)-anonymity condition. The proposed (*k*, *p*)-anonymity framework solves the disclosure problem of the traditional *k*-anonymity and *l*-diversity requirements.

Acknowledgments

This research was partially supported by the Shenzhen Technical Project under JCYJ201703071517 33005 and KQJSCX20170726103424709.

References

[1] R. Agrawal, R. Srikant, Fast algorithms for Mining Association Rules in Large Databases, *The International Conference on Very Large Data Bases*, Santiago, Chile, 1994, pp. 487-499.

[2] R. Agrawal, R. Srikant, *Quest Synthetic Data Generator*, http://www.Almaden.ibm.com/cs/quest/syndata.html, 1994.

[3] O. Abul, F. Bonchi, M. Nanni, Anonymization of Moving Objects Databases by Clustering and Perturbation, *Information Systems*, Vol. 35, No. 8, pp. 884-910, December, 2010.

[4] M. S. Chen, J. S. Park, P. S. Yu, Efficient Data Mining for Path Traversal Patterns, *IEEE Transactions of Knowledge and Data Engineering*, Vol. 10, No. 2, pp. 209-221, April, 1998.

[5] K. Doka, M. Xue, D. Tsoumakos, P. Karras, *k*-anonymization by Freeform Generalization, *ACM Symposium on Information, Computer and Communications Security*, Singapore, 2015, pp. 519-530.

[6] P. Fournier-Viger, J. C. W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, H. T. Lam, The SPMF Open-source Data Mining Library Version 2, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Riva del Garda, Italy, 2016, pp. 36-40.

[7] G. Ghinita, P. Kalnis, Y. Tao, Anonymous Publication of Sensitive Transactional Data, *IEEE Transactions on Knowledge & Data Engineering*, Vol. 23, No. 2, pp. 161-174, February, 2011.

[8] J. Han, J. Pei, Y. Yin, R. Mao, Mining Frequent Patterns without Candidate Generation: A Frequent-pattern Tree Approach, *Data Mining and Knowledge Discovery*, Vol. 8, No. 1, pp. 53-87, January, 2004.

[9] T. P. Hong, C. W. Lin, Y. L. Wu, Incrementally Fast Updated Frequent Pattern Trees, *Expert Systems with Applications*, Vol. 34, No. 4, pp. 2424-2435, May, 2008.

[10] C. H. Hsu, H. P. Tsai, KAMP: Preserving *k*-anonymity for Combinations of Patterns, *IEEE International Conference on Mobile Data Management*, Milan, Italy, 2013, pp. 97-102.

[11] N. Li, T. Li, S. Venkatasubramanian, *t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity, *IEEE International Conference on Data Engineering*, Istanbul, Turkey, 2007, pp. 106-115.

[12] C. W. Lin, T. P. Hong, W. H. Lu, The Pre-FUFP Algorithm for Incremental Mining, *Expert Systems with Applications*, Vol. 36, No. 5, pp. 9498-9505, July, 2009.

[13] J. C. W. Lin, T. P. Hong, G. C. Lan, Updating the Sequential Patterns in Dynamic Databases for Customer Sequences Deletion, *Journal of Internet Technology*, Vol. 16, No. 3, pp. 369-377, May, 2015.

[14] C. W. Lin, Q. Liu, P. Fournier-Viger, T. P. Hong, PTA: An Efficient System for Transaction Database Anonymization, *IEEE Access*, Vol. 4, pp. 6467-6479, August, 2016.

[15] J. C. W. Lin, S. Ren, P. Fournier-Viger, T. P. Hong, J. H. Su, B. Vo, A Fast Algorithm for Mining High Average Utility Itemsets, *Applied Intelligence*, Vol. 47, No. 2, pp. 331-346, September, 2017.

[16] J. C. W. Lin, S. Ren, P. Fournier-Viger, T. P. Hong, EHAUPM: Efficient High Average-utility Pattern Mining with Tighter upper Bounds, *IEEE Access*, Vol. 5, pp. 12927-12940, June, 2017.

[17] S. Kisilevich, L. Rokach, Y. Elovici, B. Shapira, Efficient Multidimensional Suppression for K-anonymity, *IEEE Transactions on Knowledge & Data Engineering*, Vol. 22, No. 3, pp. 334-347, March, 2010.

[18] J. B. Macqueen, Some Methods for Classification and Analysis of Multivariate Observations, *The Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281-297.

[19] A. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, *L*-diversity: Privacy beyond *K*-Anonymity, *ACM Transactions on Knowledge Discovery from Data*, Vol. 1, No. 1, Article No. 3, March, 2007.

[20] P. Samarati, L. Sweeney, Generalizing Data to Provide Anonymity when Disclosing Information, *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Seattle, Washington, USA, 1998, pp. 188-201.

[21] P. Samarati, Protecting Respondents Identities in Microdata Release, *IEEE Transactions on Knowledge & Data Engineering*, Vol. 13, No. 6, pp. 1010-1027, November/ December, 2001.

[22] L. Sweeney, *k*-anonymity: A Model for Protecting Privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 557-570, October, 2002.

[23] L. Sweeney, Achieving *k*-anonymity Privacy Protection Using Generalization and Suppression, *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 571-588, October, 2002.

[24] M. Terrovitis, N. Mamoulis, P. Kalnis, Privacy Preserving Anonymization of Set-valued Data, *VLDB Endowment*, Vol. 1, No. 1, pp. 115-125, August, 2008.

[25] S. L. Wang, Y. C. Tsai, H. Y. Kao, T. P. Hong, Extending Suppression for Anonymization on Set-valued Data, *International Journal of Innovative Computing Information & Control*, Vol. 7, No. 12, pp. 6849-6863, December, 2011.

[26] Y. Wang, L. Xie, B. Zheng, K. C. K. Lee, High Utility K-anonymization for Social Network Publishing, *Knowledge & Information Systems*, Vol. 41, No. 3, pp. 697-725, December, 2014.

[27] J. C. W. Lin, J. M. T. Wu, P. Fournier-Viger, Y. Djenouri, C. H. Chen, Y. Zhang, A Sanitization Approach to Secure Shared Data in an IoT Environment, *IEEE Access*, Vol. 7, pp. 25359-25368, February 2019.

[28] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, W. C. Fu, Utility-based Anonymization Using Local Recoding, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, 2006, pp. 785-790.

[29] Y. Xu, K. Wang, W. C. Fu, P. S. Yu, Anonymizing Transaction Databases for Publication, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Nevada, 2008, pp. 767-775.

[30] M. Xue, P. Karras, C. Rassi, J. Vaidya, K. L. Tan, Anonymizing Set-valued data by Nonreciprocal Recoding, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 2012, pp. 1050-1058.

[31] B. Zhou, J. Pei, The *k*-anonymity and *l*-diversity Approaches for Privacy Preservation in social Networks against Neighborhood Attacks, *Knowledge & Information Systems*, Vol. 28, No. 1, pp. 47-77, July, 2011.

Biographies

Binbin Zhang received Ph.D. degree in Dept. of Biological Sciences in 2012 from National University of Singapore. She is currently working as an assistant professor in Shenzhen University Health Science Center, Shenzhen, China. Her interests include neuroscience, system biology, sequence analysis and bioinformatics.

Jerry Chun-Wei Lin (Ph.D.) is working as the Associate Professor at Department of Computing, Mathematics, and Physics, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 200 peer-reviewed papers. His research interests include big data analytics, privacy-preserving and security, soft computing, and machine learning.

Qiankun Liu is working as the Master student in Harbin Institute of Technology (Shenzhen), China. His research interests include data mining, soft computing and data security.

Philippe Fournier-Viger (Ph.D) is a full professor at the Harbin Institute of Technology (Shenzhen). He has published more than 160 research papers, which have received 1,000 citations in the last two years. He is the founder of the popular SPMF data mining library (http://www.philippe-fournier-viger.com/spmf/).

Youcef Djenouri obtained the PhD in Computer Engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2014. He has published over 20 refereed conference papers and 12 international journal articles in the areas of data mining, parallel computing and artificial intelligence.