ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2019.0271

PRIVACY PRESERVATION OF MICRO DATA PUBLISHING USING FRAGMENTATION

V. Arul[1], C. Vairavel[2], M. Prakash[3] and N.V. Kousik[4]

[1,2]Department of Computer Science and Engineering, Anna University, Chennai, India
[3]Department of Computer Science and Engineering, SRM Institute of Science and Technology, India
[4]Department of Computing Science and Engineering, Galgotias University, India

Abstract

Organization such as hospitals, publish detailed data or micro data about individuals for research or statistical purposes. Many applications that employ data mining techniques involve mining data that include private and sensitive information about the subjects. When releasing the micro data, it is necessary to prevent the sensitive information of the individuals from being disclosed. Several existing privacy-preserving approaches focus on anonymization techniques such as generalization and bucketization. Recent work has shown that generalization loses considerable amount of information for high dimensional data, the bucketization does not prevent membership disclosure and does not make clear separation between quasi-identifying attributes and sensitive attributes. In this work a novel technique called Fragmentation is proposed for publishing sensitive data with preventing the sensitive information of the individual. Here first the vertical Fragmentation is applied to attributes. In vertical Fragmentation, attributes are segmented into columns. Each column contains a subset of attributes. Secondly, the horizontal Fragmentation is applied to tuples. In this, tuples are segmented into buckets. Each bucket contains a subset of tuples. Finally the real dataset is used for experiments and the results show that this Fragmentation technique preserves better utility while protecting privacy threats and prevents the membership disclosure.

Keywords:

Privacy, Privacy Preservation, Data Anonymization, Data Publishing, Data Security

1. INTRODUCTION

The practice of mining worthwhile, interesting and earlier unknown information from big data sets is known as data mining. Accessibility of high quality data and effective information sharing is the success factors of data mining. The collection of digital information by administrations, companies and persons has produced an atmosphere that eases important data mining and data analysis. Sharing of data that contains sensitive information about individuals leads to violate the privacy of an individual. This creates serious issues on privacy of individual's sensitive information. There are policies and guidelines to protect the privacy while publishing individual's sensitive information. Policies and guidelines cannot guarantee the privacy protection. Privacy Preserving Data Publishing is a method of data publishing in which the individual privacy is preserved while publishing data. The micro data consist of set of attributes that are partitioned into three classes: a) Explicit attributes are set of attributes that clearly identifies individual such as Social Security Number (SSN) or Name; b) Quasi-Identifiers are set of attributes that could potentially identifies individual when taken together such as date of birth, sex and zip code; c) Sensitive attributes are set of attributes which contains sensitive information of individual such as salary and disease. Anonymization is the process of hiding the individuality or privacy information of record owners.

Generalization and Bucketization are some anonymization techniques which are designed to protect privacy of micro data release. First the explicit identifiers are removed from records and then records are divided into buckets in both generalization and bucketization. Secondly, QI values are converted into less specific but consistent semantine values in each bucket in a general manner to avoid the duplicates which are characterized by their QI values. The generalization technique for k-anonymity leads to loss of information.

In bucketization, the QI values and SA values are separated and then SA values in each bucket are randomly permutated. Bucketization publishes their QI values in their original form, so it does not prevent membership disclosure. Bucketization needs clear separation between quasi-identifiers and sensitive attributes. But it is difficult to separate the quasi-identifiers and sensitive attributes in many datasets. While separating the quasi-identifiers and sensitive attributes, bucketization breaks the attribute correlation between quasi-identifiers and sensitive attributes.

The novel data anonymization scheme fragmentation is proposed for publishing sensitive data by preventing the sensitive information of the individual. Here first the vertical fragmentation is applied to attributes. In vertical fragmentation, attributes are segmented into columns. Each column contains a subset of attributes. Secondly, the horizontal fragmentation is applied to tuples. In this, tuples are segmented into buckets. Each bucket contains a subset of tuples. Finally the real dataset is used for experiments and the results show that this fragmentation technique preserves better utility while protecting privacy and prevents the membership disclosure.

2. GENERALIZATION FOR *k*-ANONYMITY

Definition 1 (*k*-Anonymity): If a table satisfies *k*-anonymity for some value *k*, the Quasi-Identifier values of one record can match with Quasi-Identifier values of at least *k*-1 records in the table. The local recoding technique used in generalization to achieve *k*-anonymity. In that the tuples are grouped in to buckets and in each bucket all values of one attribute is replaced with common value.

Table.1. Original Micro Data Table

Pin Code	Age	Sex	Disease
635207	30	M	Cancer
635307	35	F	Heart Disease
635109	42	F	Heart Disease
635605	49	F	Flu
638106	55	M	Heart Disease
638183	59	M	Cancer
638194	61	M	Cancer
638125	65	F	Flu

For example, Table.1 shows that original micro data table. The explicit attributes are removed from this table.

Table.2. Generalized Table

Pin Code	Age	Sex	Disease
635*	[30-50]	*	Cancer
635*	[30-50]	*	Heart Disease
635*	[30-50]	*	Heart Disease
635*	[30-50]	*	Flu
6381*	[55-65]	*	Heart Disease
6381*	[55-65]	*	Cancer
6381*	[55-65]	*	Cancer
6381*	[55-65]	*	Flu

The Table.2 show the generalized table, values of Quasi-Identifiers in each bucket are generalized and made equal.

Observation 1: Generalization for k-anonymity losses considerable amount of data.

Observation 2: Generalization for k-anonymity does not protect information against homogeneity attack and background knowledge attack.

3. BUCKETIZATION FOR *L*-DIVERSITY

The *L*-diversity is used to address the issues of *k*-anonymity.

Definition 2 (*l*-Diversity)*:* If a bucket in a table contains at least *l* “well represented” values for the sensitive attributes, then that bucket is *l*-diverse. All the buckets in table are *l*-diverse, and then the table is *l*-diverse.

The *l*-diversity is achieved through bucketization technique. In that first the Quasi-Identifiers are grouped in one column and Sensitive Attribute values in another column. Then the Sensitive Attribute values are randomly sorted to achieve *l*-diversity.

The Table.3 shows the bucketized version of the Table.1. Suppose the person match the records in the bucketized table with publicly available voter list table, can easily identify the individual, because the bucketized table contains the original values for Quasi-Identifiers.

Table.3. Bucketized Table

Pin Code	Age	Sex	Disease
635207	30	M	Heart Disease
635307	35	F	Cancer
635109	42	F	Flu
635605	49	F	Heart Disease
638106	55	M	Flu
638183	59	M	Heart Disease
638194	61	M	Cancer
638125	65	F	Cancer

Observation 1: Bucketization does not prevent membership disclosure.

Observation 2: There is no clear separation between quasi-identifiers and sensitive attributes.

4. FRAGEMENTATION

The novel data anonymization scheme fragmentation is proposed for publishing sensitive data by preventing the sensitive information of the individual. Here first the vertical fragmentation is applied to attributes. In vertical fragmentation, attributes are segmented into columns. Each column contains a subset of attributes. Secondly, the horizontal fragmentation is applied to tuples. In this, tuples are segmented into buckets. Each bucket contains a subset of tuples. The fragmentation consists of following phases.

1) Attribute Grouping
2) Tuple Partitioning
3) Column Generalization

4.1.1 Attribute Grouping:

In attribute grouping, the attributes are grouped depends on their correlation. Attributes with high correlation are grouped together and form a column. Grouping of less correlated attributes lead to easy identification because uncorrelated values associated much frequently. But grouping of highly correlated attributes protects privacy.

4.1.2 Calculating Correlation:

To group the highly correlated attributes, first we have to measure the correlation between the attributes. Here we use the mean-square contingency coefficient for calculating correlation.

The above correlation measurement is only used for categorical attributes. For continuous attributes we have to perform discretization before measuring correlation.

4.1.3 Attribute Clustering:

Each and every attributes are points in the cluster space. We use clustering for attribute grouping. In cluster space, the distance between attributes is in between 0 and 1 and calculated as $d(A_1,A_2) = 1\text{-}\Phi_2(A_1,A_2)$. If the distance is very small, then that attribute are highly correlated. Here *k*-medoids method is used for attribute clustering. In that CLARA algorithm is used instead of PAM algorithm. PAM algorithm is only used for small data sets. It is not efficient for large data sets. But CLARA is used for large data sets with high number of attribute

4.2 TUPLE PARTITIONING

Tuples are divided into buckets during the multiple partitioning period. We change the tuple partition Mondrian algorithm. No generalization is applied to the tuples; unlike Mondrian K-anonymity, we use Mondrian to divide tuples in buckets.

Algorithm Tuple-Partition (*T*,*l*)

```
Step 1: $Q = \{T\}$; $SB = \emptyset$.
Step 2: While $Q$ is not empty
Step 3: Remove the first bucket $B$ from $Q$; $Q = Q$-$\{B\}$.
Step 4: Split $B$ into two bucket $B_1$ and $B_2$, as in Mondrian.
Step 5: If diversity-check ($T$, $Q \cup \{B_{1,}B_2\} \cup SB$, $l$)
Step 6: $Q = Q \cup \{B_{1,}B_2\}$.
Step 7: Else $SB = SB \cup \{B\}$.
Step 8: Return $SB$
```

The algorithm maintains two data structures: 1) a queue of buckets *Q* and 2) a set of sliced buckets *SB*. Initially, *Q* contains only one bucket which includes all tuples and *SB* is empty. In each iteration, the algorithm removes a bucket from *Q* and splits the bucket into two buckets. If the sliced table after the split satisfies *l*-diversity, then the algorithm puts the two buckets at the end of the queue *Q*. Otherwise, we cannot split the bucket anymore and the algorithm puts the bucket into *SB*. When *Q* becomes empty, we have computed the sliced table. The set of sliced buckets is *SB*. The main part of the tuple-partition algorithm is to check whether a sliced table satisfies *l*-diversity.

4.3 COLUMN GENERALIZATION

In the second phase, tuples are generalized to satisfy some minimal frequency requirement. We want to point out that column generalization is not an indispensable phase in our algorithm. As shown by Xiao and Tao, bucketization provides the same level of privacy protection as generalization, with respect to attribute disclosure. Although column generalization is not a required phase, it can be useful in several aspects. First, column generalization may be required for identity or membership disclosure protection.

If a column value is unique in a column (i.e., the column value appears only once in the column), a tuple with this unique column value can only have one matching bucket. This is not good for privacy protection, as in the case of generalization or bucketization where each tuple can belong to only one equivalence-class or bucket. The main problem is that this unique column value can be identifying.

In this case, it would be useful to apply column generalization to ensure that each column value appears with at least some frequency. Second, when column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller. While column generalization may result in information loss, smaller bucket-sizes allow better data utility. Therefore, there is a trade-off between column generalization and tuple partitioning.

5. EXPERIMENTAL ANALYSIS

Experiment is conducted on slicing technique for evaluating the performance of fragmentation on privacy preservation of micro data. The proposed privacy preservation of micro data publishing using fragmentation is evaluated with different performance metrics and compare it with existing privacy preserving data publishing.

By comparing existing privacy preserving data publishing, the experimental results show that proposed privacy preservation of micro data publishing using fragmentation technique, not only have satisfactory performance, but also achieve stronger slicing technique.

The performance of privacy preservation of micro data publishing using fragmentation is evaluated by the following metrics

- Column Size
- Classification Accuracy
- Tuple Partition Rate

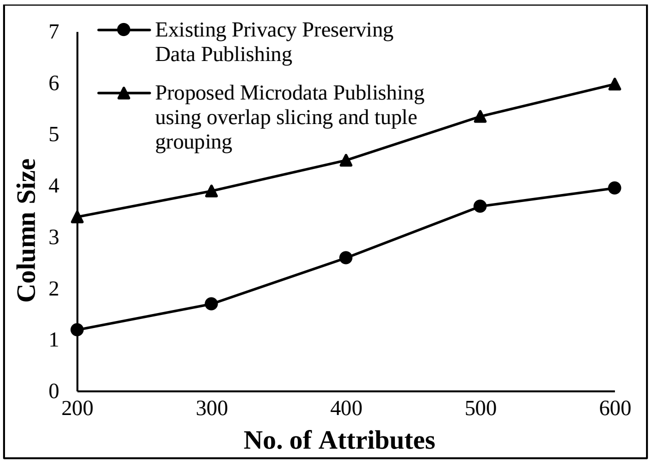

Fig.1. Column Size

The Fig.1 demonstrates the column size, *x*-axis represents the number of attributes whereas *y*-axis denotes the column size using both the privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation. When the number of attributes increased, column size also gets increases accordingly. The size of column is illustrated using the existing privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation.

The Fig.2 shows better performance of proposed privacy preservation of micro data publishing using fragmentation in terms of attributes than existing privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation. Privacy preservation of micro data publishing using fragmentation achieve 15-23% less packet processing delay variation when compared with existing system.

The Fig.2 demonstrates the classification accuracy, where *x*-axis represents number of attribute density whereas *y*-axis denotes the classification accuracy using both the privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation. When the number of attribute density increased classification accuracy also gets increased.

The Fig.3 shows the effectiveness of classification accuracy over different number of attribute density than existing privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation. Privacy preservation of micro data publishing using fragmentation achieve 20-35% more classification accuracy when compared with existing schemes.

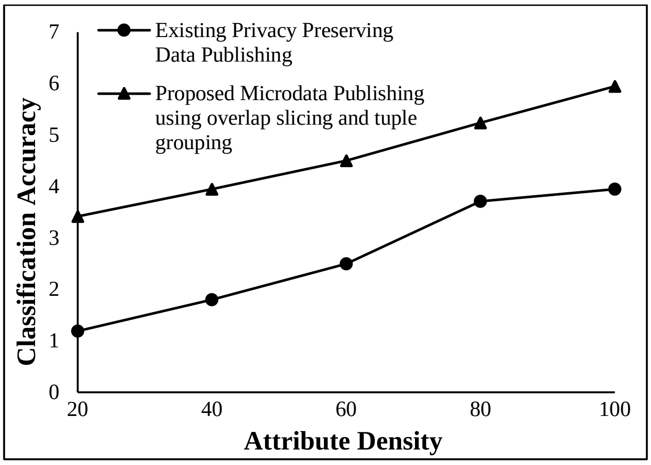

Fig.2. Classification Accuracy

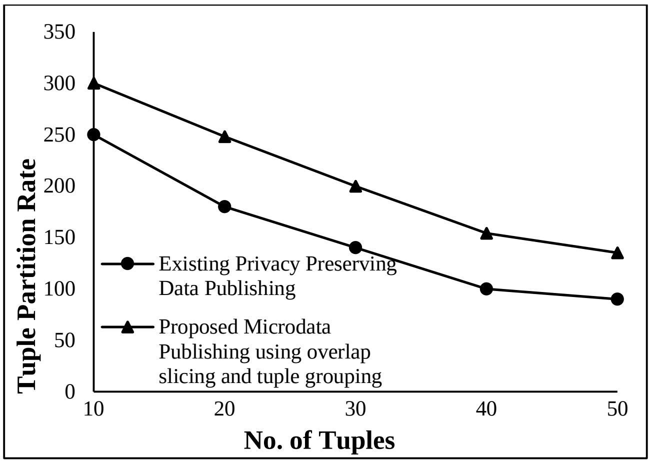

Fig.3. Tuple Partition Rate

The Fig.3 demonstrates the tuple partition rate, *x*-axis represents the number of tuple whereas *y*-axis denotes the tuple partition rate using both the privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation. When the number of tuple increased, tuple partition rate gets decreases accordingly. The rate of tuple partition is illustrated using the existing privacy preserving data publishing and our proposed privacy preservation of micro data publishing using fragmentation.

6. CONCLUSION

This paper describes a new approach called fragmentation to privacy-preserving micro data publishing. Fragmentation overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. We illustrate how to use fragmentation to prevent attribute disclosure and membership disclosure. The proposed scheme presents a fragmentation with tuple grouping to preserve privacy in micro data publishing. Fragmentation duplicates an attribute in more than one column. Present tuple grouping algorithm to minimize complexity in random grouping of micro data publishing.

REFERENCES

[1] Tiancheng Li, Nninghui Li, Jian Zhang and Ian Molloy, "Slicing: A New Approach for Privacy Preserving Data Publishing", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 24, No. 3, pp. 561-574, 2012.

[2] C. Aggarwal, "On K-Anonymity and the Curse of Dimensionality", *Proceedings of 31st International Conference on Very Large Data Bases*, pp. 901-909, 2005.

[3] J. Brickell and V. Shmatikov, "The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing", *Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 70-78, 2008.

[4] D.R. Kumar Raja and S. Pushpa, "Diversifying Personalized Mobile Multimedia Application Recommendations through the Latent Dirichlet Allocation and Clustering Optimization", *Multimedia Tools and Applications*, pp. 1-20, 2019.

[5] N. Koudas, D. Srivastava, T. Yu, and Q. Zhang, "Aggregate Query Answering on Anonymized Tables", *Proceedings of International Conference on Data Engineering*, pp. 116-125, 2007.

[6] K. LeFevre, D. DeWitt and R. Ramakrishnan, "Mondrian Multidimensional K-Anonymity", *Proceedings of International Conference on Data Engineering*, pp. 20-25, 2006.

[7] K. Raja and S. Pushpa, "Novelty-Driven Recommendation by using Integrated Matrix Factorization and Temporal-Aware Clustering Optimization", *International Journal of Communication Systems*, pp. 1-16, 2018.

[8] N. Li, T. Li and S. Venkatasubramanian, "T-Closeness: Privacy Beyond k-Anonymity and '-Diversity", *Proceedings of International Conference on Data Engineering*, pp. 106-115, 2007.

[9] T. Li and N. Li, "On the Tradeoff between Privacy and Utility in Data Publishing", *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 517-526, 2009.

[10] D.J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke and J.Y. Halpern, "Worst-Case Background Knowledge for Privacy- Preserving Data Publishing", *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 126-135, 2007.

[11] U. Selvi and S. Puspha, "A Review of Big Data an Anonymization Algorithms", *International Journal of Applied Engineering Research*, Vol. 10, No, 17, pp. 13125-13130, 2015.

[12] L. Sweeney, "Achieving K-Anonymity Privacy Protection using Generalization and Suppression", *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 6, pp. 571-588, 2002.

[13] M. Terrovitis, N. Mamoulis, and P. Kalnis, "Privacy-Preserving Anonymization of Set-Valued Data", *Proceedings of 31st International Conference on Very Large Data Bases*, pp. 115-125, 2008.

[14] R.C.W. Wong, J. Li, A.W.C. Fu and K. Wang, "(α, k)-Anonymity: An Enhanced k-Anonymity Model for Privacy Preserving Data Publishing", *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 754-759, 2006.

[15] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation", *Proceedings of 31st International Conference on Very Large Data Bases*, pp. 139-150, 2006.

[16] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.C. Fu, "Utility- Based Anonymization Using Local Recoding", *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 785-790, 2006.

[17] Benjamin C.M. Fung, Ke Wang, Ada Wai-Chee Fu, and Philip S. Yu, "*Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*", CRC Press, 2011.