International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Review of Slicing Approach: Data Publishing with Data Privacy and Data Utility

Vina M. Lomte[1], Hemlata B. Deorukhakar[2]

[1]Professor Computer Engineering Department, RMD Sinhgad College of Engineering, Pune, India

[2]M.E. Computer Engineering Department, RMD Sinhgad College of Engineering, Pune, India

Abstract: ***Data publishing with data privacy and data utility has been emerged to manage high dimensional data efficiently. In this paper, to deal with this advancement in data mining technology using accentuate approach of slicing. Slicing provide better data utility and data privacy than bucketization and generalization. Slicing can handle high dimensional data than generalization which loses great amount of information for high dimensional data. Slicing also prevents from membership disclosure and attributes disclosure.***

Keywords: Data Publishing, Data privacy, Generalization, Bucketization, Slicing, Data Utility, Data Anonymization.

1. Introduction

Fast growing field of Data Mining is the process of discovering interesting patterns and knowledge from large database. It is also called as KDD process i.e. Knowledge Discovery from Data. It allows data analysis while preserving data privacy. Data privacy is prevent personal confidential or private data from unnecessarily distributed or publicly known or not be misused by third person. In privacy preserving data publishing, interesting and useful information is publish with privacy of sensitive information has been preserved. There are two stages in privacy preserving data publishing first is data collection and second data publishing. In data collection, data holder stores data which is gathered by data owner. In data publishing, data can be released to data recipient by data holder and data recipient mines published secured data. This scenario of privacy preserving data publishing shown in Figure: 1.

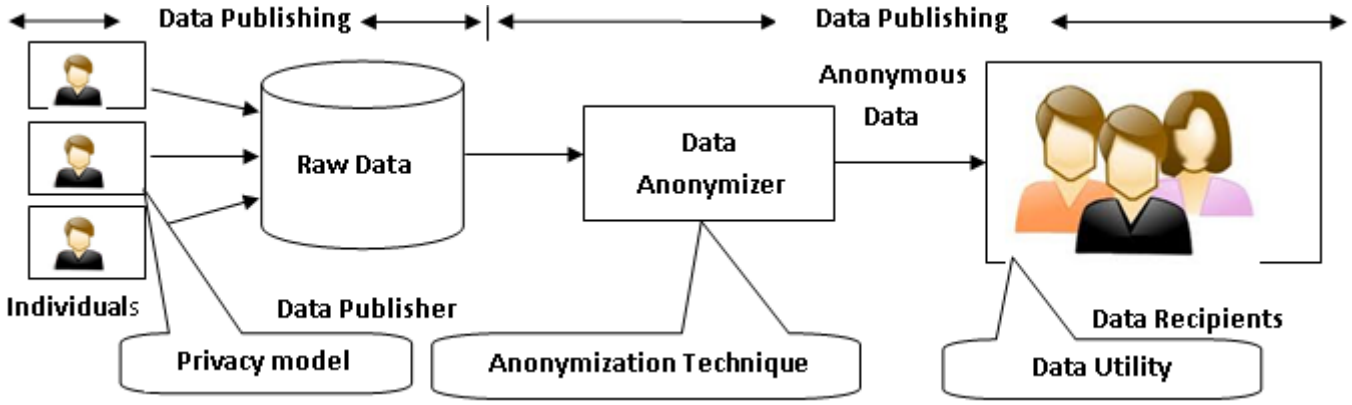

Figure: 1: Privacy preserving data publishing

Data Anonymization is a technology that convert clear text into another text which is hide the data. Up to this point, number of Anonymization techniques are introduced these are bucketization and generalization [1]. Bucketization does not provide protection for membership disclosure and it required clear separation between quasi identifier and sensitive attributes but this is not possible every time. Generalization does not handle high dimension data efficiently, it loses great amount of information [2]. So that in this paper, slicing is a new approach introduced to overcome these drawbacks of Bucketization and Generalization. Slicing provide better data utility and data privacy than bucketization and generalization. Slicing can handle high dimensional data efficiently. Slicing also prevents from membership disclosure and attributes disclosure. Slicing can do all ℓ-diversity requirements.

Microdata consist of detailed information about each individual entity such as a person, a household or an organization. There are three types of attributes in microdata: (1) Identifiers which uniquely identify an individual person or entity such as Name or security Number; (2) Quasi-Identifiers (QI) attribute which the attacker may already know (possibly from publicly available database) and which taken together to identify an individual such as Birth-date, sex and zipcode; (3) Sensitive Attribute (SA) which are unknown to the attacker and consist of sensitive value like salary, disease.

Table 1: The Microdata Table Example

TID	*ID*	*Quasi-identifiers*			*Sensitive*
	Name	Age	Sex	Zipcode	Disease
1	ALEX	22	M	47906	HIV
2	BOB	22	F	47906	FLU
3	CARL	33	F	47905	FLU
4	DEBRA	52	F	47905	TB
5	ELAIN	54	M	47302	FLU
6	FRANK	60	M	47302	HIV
7	GRAY	60	M	47304	HIV
8	JOY	64	F	47304	CANCER

2. Related Work

The popular Anonymization techniques are introduced up to this point is bucketization and generalization.

Table 2: The Original Table

Age	*Sex*	*Zipcode*	*Disease*
22	M	47906	HIV
22	F	47906	FLU
33	F	47905	FLU
52	F	47905	TB
54	M	47302	FLU
60	M	47302	HIV
60	M	47304	HIV
64	F	47304	CANCER

2.1 Bucketization

In bucketization, partition tuple in the table into buckets and separate the quasi-identifiers attributes with the sensitive attributes. Then after by randomly permuting value of sensitive attribute in each bucket. Bucketization handles high dimensional data. Bucketization provides better utility than generalization [1,3,5]. But it has some disadvantages. First, Bucketization does not prevent for membership disclosure and second, it required clear separation between quasi identifier and sensitive attributes but this is not possible every time. Bucketization for ℓ-diversity make sure each group on takes well represented sensitive values appear at most $1/\ell$ times in group. Each equivalence class has at least ℓ well-represented sensitive values. Doesn't prevent probabilistic inference attacks Sensitive attributes must be "diverse" within each quasi-identifier equivalence class [5]. 2 ℓ-diversity satisfy in following Table 3.

Table 3: The Bucketized Table

Age	*Sex*	*Zipcode*	*Disease*
22	M	47906	FLU
22	F	47906	HIV
33	F	47905	TB
52	F	47905	FLU
54	M	47302	CANCER
60	M	47302	FLU
60	M	47304	HIV
64	F	47304	HIV

2.2. Generalization

In Generalization, partitions attributes in the table into columns and then replace quasi-identifiers attributes with less specific, but semantically consistent values i.e. get k identical values. So that tuple cannot distinguished by their quasi-identifiers [1,4,5]. Every person looks identical to k-1others on quasi-identifiers. Partitioning order value into intervals. Some disadvantages of generalization. Generalization loses great amount of information, especially for high dimensional and it also reduces data utility.

Generalization for k-anonymity is the information for each person contained in the released table cannot be distinguished from at least k-1 individuals whose information also appears in the release. K- anonymity is to release data where for all possible queries at least K times result will be return. Example: you try to identify a man in the released table, but the only information you have is his birth date and gender. There are k men in the table with the same birth date and gender. Any quasi-identifier present in the released table must appear in at least k records [5]. 4 k-anonymity satisfy in following Table 4.

Disclosure means act or process of revealing or uncovering the private data. There are three types of privacy disclosure threat:

1) Membership Disclosure: attacker cannot tell that a given person in given data table.
2) Attribute Disclosure: attacker cannot tell that a given person has a certain sensitive attribute.
3) Identity Disclosure: attacker cannot tell which record corresponding to given a person.

Table 4: The Generalized Table

Age	Sex	Zipcode	Disease
[20-52]	*	4790*	HIV
[20-52]	*	4790*	FLU
[20-52]	*	4790*	FLU
[20-52]	*	4790*	TB
[54-64]	*	4730*	FLU
[54-64]	*	4730*	HIV
[54-64]	*	4730*	HIV
[54-64]	*	4730*	CANCER

3. Review of Slicing

A new accentuate technique is developed for data publishing with data privacy and data utility is called as slicing [1]. In slicing first, vertically partition attribute in the table into columns. Each column consists of a subset of attribute. So that highly correlated attributes value in same column and preserve correlation among those attribute this is good for data utility and breaking association of uncorrelated attribute it is also good for data privacy because the association between values of uncorrelated is much less frequent and thus more identifiable and noticeable to adversary. Shown in Table.5: The Sliced table contains 2 columns.

Slicing also, horizontally partition tuples in the table into buckets. Each bucket consists of a subset of tuples. Shown in Table.5: The Sliced table contains 2 buckets.

Then after within each bucket, values in each column are randomly permuted and then break correlation between values of different columns. So that the correlation between the values of two columns within one bucket is hidden from

adversary Multiset based generalization is equivalent to a trivial slicing approach where each column contains exactly one attribute.

Table 5: The Sliced table

(Age ,Sex)	*(Zipcode, Disease)*
(22, M)	(47905, FLU)
(22, F)	(47906, HIV)
(33, F)	(47905, TB)
(52, F)	(47906, FLU)
(54, M)	(47304, CANCER)
(60, M)	(47302, FLU)
(60, M)	(47302, HIV)
(64, F)	(47304, HIV)

4. Slicing Algorithm

This algorithm consists of three stages:

1. Attribute Partitioning
2. Column Generalization
3. Tuple Partitioning

4.1. Attribute Partitioning

In attribute partitioning first, vertically partition attribute in the table into columns. Each column consists of a subset of attribute. So that highly correlated attributes value in same column and preserve correlation among those attribute this is good for data utility and breaking association of uncorrelated attribute it is also good for data privacy because the association between values of uncorrelated is much less frequent and thus more identifiable and noticeable to attacker.

4.2. Column Generalization

In column generalization, main problem of unique column value can be noticeable or identifying by attacker. In this case, it ensure that each column appears with at least some frequency. Tuples are generalized to satisfy some k-anonymity and ℓ-diversity means minimal frequency requirements. It provides the same level of data privacy protection as generalization provide with respect to attribute disclosure [1]. It prevents membership/ identity disclosure. Mondrian Anonymization algorithm used for column generalization [6].it apply on sub table consist of only one attribute in one column to satisfy the anonymity requirements.

4.3. Tuple Partitioning

In this phase, horizontally partition tuples in the table into buckets. Each bucket consists of a subset of tuples. In this Mordrian with modification used for partition tuples into buckets [6]. In tuple partition algorithm, the algorithm maintains two data structures:

1) A queue of buckets Q
2) A set of sliced buckets SB

```
Algorithm tuple-partition (T, ℓ)
1. Q = {T}; SB = ∅.
2. While Q is not empty
3. Remove the first bucket B from Q; Q=Q − {B}.
4. Split B into two buckets B1 and B2, as in Mondrian.
5. if diversity-check (T, Q ∪ {B1, B2} ∪ SB, ℓ)
6. Q = Q ∪ {B1, B2}.
7. else SB = SB ∪ {B}.
8. return SB.
```

Figure 2: Tuple-Partition Algorithm

Initially, Q contains only one bucket of all tuples and SB is empty. In each loop, the algorithm removes a bucket from queue Q and splits the bucket into two buckets (the splitting criteria which is described in Mondrian [6]). If the sliced table after the split satisfies ℓ-diversity, then the algorithm puts the two buckets at the end of the queue Q. else, we cannot split the bucket anymore and the algorithm puts the bucket into SB. When queue Q becomes empty, computed the sliced table and return set of sliced buckets is SB.

In tuple-partition algorithm main part is to check whether a sliced table satisfies ℓ-diversity. In, the ℓ -diversity-check algorithm, for each tuple, the algorithm maintains a list of statistics L[t] about t's matching buckets B. Each element in the list L[t] contains statistics about one matching bucket: the matching bucket probability p (t, B) and the distribution of candidate sensitive values in bucket D (t, B).

```
Algorithm diversity-check (T, T*, ℓ)
1. for each tuple t ∈ T, L[t] = ∅.
2. for each bucket B in T*
3. Record f (v) for each column value v in bucket B.
4. for each tuple t ∈ T
5. Calculate p (t, B) and find D (t, B).
6. L[t] = L[t] ∪ {<p (t, B), D (t, B)>}.
7. for each tuple t ∈ T
8. Calculate p (t, s) for each s based on L[t].
9. if p(t, s) ≥ 1/ℓ, return false.
10. return true.
```

Figure 3: ℓ-diversity-Check Algorithm

The algorithm first takes one scan of each bucket B to record the frequency f(v) of each column value v in bucket B. Then the algorithm takes one scan of each tuple t in the table T to find out all tuples that match bucket B and record their matching bucket probability p(t,B) and the distribution of candidate sensitive values in bucket D(t,B), which are added to the list L[t]. At the end, for each tuple obtained , the list of statistics L[t] about its matching buckets B. A final scan of the tuples in will compute the p(t,s) values based on the law of total probability. By analysing the time complexity the tuple partition algorithm. Time complexity of tuple partitioning algorithm (Mondrian [6]) is O(n log n) And modification $O(n^2)$ and the total time complexity is $O(n^2 \log n)$.

5. System Architecture

System Architecture of slicing approach is explain in below in which contain several phases from which data flow through step by step. Original data contain original table where data in understandable manner. In second stage data is going to generalized due to secure manner. But due to some problem that table bucketized and it is comes in next stage and so that the data comes through various phases it's becomes secured sliced table.

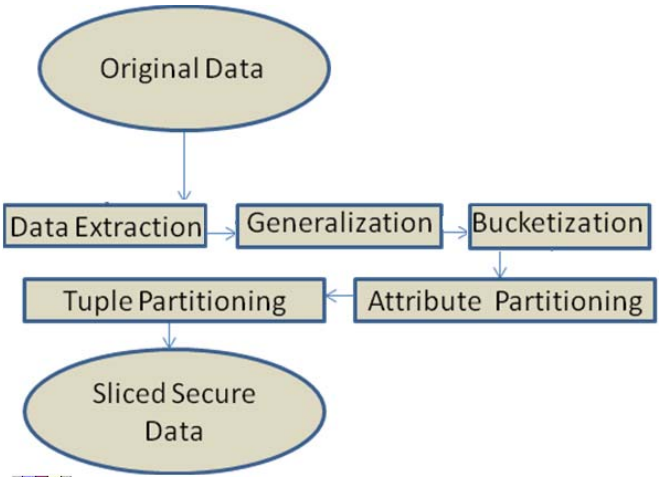

Figure 4: System Architecture

6. Conclusion and Future Work

In this paper we have discussed the Slicing approach along with generalization and bucketization and its advantages and disadvantages. This study will helpful in future to continue research in this area. In this slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Slicing can be used to prevent membership disclosure. Slicing can also handle high-dimensional data. In this work motivates various directions for future research or work. First, Overlapping slicing, this duplicates an attribute in more than one column. Second, Design more effective tuple grouping algorithms. Third, Handle big data.

References

[1] Tiancheng Li, Ninghui Li, Senior Member, IEEE, Jia Zhang,Member, IEEE, and Ian Molloy "Slicing: A New Approach for Privacy Preserving Data Publishing" Proc. IEEE Transactions On Knowledge And Data Engineering, Vol. 24, No. 3, March 2012.
[2] Aggarwal. C., "On K-Anonymity and the Curse of Dimensionality," In Conf. Very Large Databases (VLDB), pages 901-909, 2005.
[3] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In ICDE, pages 126–135, 2007.
[4] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In VLDB, pages 139–150, 2006.
[5] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-diversity. In ICDE, pages 106–115, 2007.
[6] K. LeFevre, D. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In ICDE,page 25, 2006.

Author Profile

Prof. Vina M. Lomte, received the B.E. degree in Computer Science and Engineering from Amravati University, BNCOE, Pusad. and M.E. degree in Computer Engineering from Mumbai University, MGMCET, Kamothe. Currently working as Assistant Professor of Computer Engineering Department in RMD SSOE Pune, India.

Hemlata B. Deorukhakar received the B.tech. degree in Computer Science and Engineering from IGNOU, New Delhi in 2013. Currently appearing M.E. 1st year Computer Engineering in RMD SSOE Pune.