Perpetuate Data Report based on the Slicing Approach

G. Sai Raghunath [1], Bhaludra Raveendranadh Singh [2], Moligi Sangeetha [3]

[1] pursuing M.Tech (CSE) , [2] Principal , [3]Associate Professor&HOD(CSE)

Visvesvaraya College of Engineering and Technology (VCET), M.P Patelguda, Ibrahimpatnam (M), Ranga Reddy (D)-501510, India

Abstract: **Anonymization is a technique preserving privacy on micro data, we have so many anonymization techniques like generalization, bucketization all these are privacy preserving on sensitive data, with these techniques there is no security for the data, generalization loses the important data and bucketization is not preventing membership disclosure and does not apply on the data for clear separation in quasi identifiers and sensitive attributes.**

In this paper we are proposing a novel technique providing privacy on sensitive data is called Slicing, this technique divides the particular data into horizontally and vertically. Here we are showing that slicing is better data utility technique compare with generalization and this can provide membership disclosure protection. Alternative major advantage of slicing is it can handle high-dimensional data. We exhibits how slicing provide membership disclosure protection and it develop an efficient algorithm for computing sliced data which are required l - diversity. Our works confirm that slicing is better preserve data utility concept compare with generalization and more effective than bucketization, our workload involves the sensitive attributes. Our experiment also described that slicing is used to prevent membership disclosure protection.

Keywords: generalization, bucketization, slicing, k-anonymization, l-diversity and attribute data.

1. INTRODUCTION

In the last few years it's extensively studied about privacy-preserving publishing of micro data, micro data contains individual records each of which having information about individual entity such as a house hold, a person or an organization, so many micro data anonymization techniques introduced, some of the important techniques are generalization k-anonymity and bucketization in l – diversity, in this two techniques attributes are divided in to 3 categories: 1) some of them are identifiers, these are identified by name or some social security numbers, 2) some of them are quasi attributes, this are like sex, age and address etc…3) some of the are Sensitive Attributes these are like age and salary.

In both techniques generalization and bucketization the one removes the identifiers from the data after those segregate rows in to buckets. This both techniques different in second step, generalization transfer the QI – values in each buckets, values so that tuples in the same bucket cannot be illustrious by their QI standards. In bucketization, one splits the SAs from the QIs by arbitrarily permuting the SA values in every bucket. The anonymized data contains of a set of buckets with permuted complex element values.

1.1 Motivation of Slicing

It is showing that generalization for k-anonymity is loses significant on the micro data. This is because of the following three reasons, k-anonymity suffers from the obscenity of dimensionality. In order for generalization to be better, all the records in the same bucket should be close to each other thus that generalizing the records not be lose much information. Though, in high-dimensional data, lot of data points has similar detachments with every one, forcing a excessive amount of generalization to fulfill k-anonymity flush for relative slight k's. Second, to accomplish data investigation or data mining jobs on generalized table, the data predictor has to do the constant distribution guess that each value in a generalized interval is similarly possible, as no other distribution theory can be vindicated. This meaningfully decreases the data utility of the generalized data. Third, because every element is generalized distinctly, associations between different elements are lost. While to study element associations on the generalized table, the data specialist has to adopt that each and every possible combination of element values is similarly possible. This is an integral problem of generalization that avoids effective analysis of element associations.

Compare with generalization bucketization is do better performance in data utilizing. It will work in some limitations it will not prevent membership disclosure protection in the first, because bucketization circulates the QI procedures, an antagonism can discover whether a particular has a record in the circulated data or not. 87% of the persons in the United States could be inimitably identified by using only three attributes (Sex, Birthdate and Zipcode). A microdata generally encloses many other elements besides those three elements. This means that the membership information of most individuals could be conditional from the bucketized table. Second, bucketization needs a clear departure between QIs and SAs. Though, in several datasets, it is imprecise which attributes are QIs and which are SAs. Third, by extrication the sensitive element from the QI attributes, bucketization breaks the element correlations between the Quasi Interfaces and the SAs.

2. PROPOSED WORK

Here, we are introducing an innovative data anonymization technique calling slicing for improve the current state of the art. Slicing partitions the dataset into both vertically and horizontally. Vertical segregating is done by assemblage attributes into columns based on the relationships among the elements. Each and every column consist a subclass of attributes those are highly interrelated. Horizontal segregation is done by gathering tuples into buckets. Finally, in the each bucket, values are in each column are arbitrarily permutated to disruption the linking between different columns.

The initial idea of slicing break relationship in the cross column, but needs to sphere association with each column that will reduce the dimensionality of data and it will provide better utilization than bucketization and generalization. Slicing reserves utility because it will gather the high correlated elements together and preserves the relation between those attributes. Slicing provide security because it breaks the relationship between uncorrelated attributes, where those are not frequent and hence identifying. Note that whenever the dataset encloses one SA and QIs, bucketization will break their relationship; on the other hand slicing, can gather some QuasiInterfaces attributes with the SA, preservative element correlations with the delicate attribute.

Finally, we are conducting wide workload experimentations. Our results endorse that slicing conserves much well data utility than generalization. In our workloads including the composite attribute, slicing is also more effective than bucketization. In some cataloguing experiments, slicing shows the best performance than using the original data. Our trials also show the boundaries of bucketization protection in membership disclosure and slicing remedies these limitations.

2.1 SLICING

In this session we are going to discuss about the novel approach called slicing with some example, how it will provide efficient security on microdata than bucketization and generalization.

Table 1 displays the microdata tables and their anonymizations versions using various anonimyzation techniques. Table 1(a) shows the original data. The three Quasi Interface attributes are {Sex, Age, Zipcode}, and Disease the sensitive attribute. A generalized table it will satisfies 4-anonymity is displayed in Table 1(b), a bucketized table that fulfills 2-diversity is displays in Table 1(c), the each attribute value is replaced in generalized table with the multiset of values in bucket is shown in the Table 1(d), and the two sliced tables are displayed in the Table 1(e) and 1(f). The Slicing, that will first partitions element into columns. Each and every column consist a subset of attributes. This is vertically segregates the table. For example, the sliced table in Table 1(f) consists of 2 columns: the first column encloses {Age, Sex} and second column comprises {Zipcode, Disease}. The sliced table is displayed in Table 1(e) contains 4 columns, where each column encloses exactly one attribute.

Slicing also segregates the each record into buckets, each and every bucket contains the information about record, this will horizontally segregate the table data.

Sex	Age	Zip code	Disease
22	M	47906	dyspepsia
22	F	47906	flu
33	F	47905	flu
52	F	47905	bronchitis
54	M	47302	Flu
60	M	47302	dyspepsia
60	M	47304	dyspepsia
64	F	47304	gastritis

(a) The original table

Sex	Age	Zipcode	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	Flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

(b) The generalized table

(c)

Sex	Age	Zipcode	Disease
22	M	47906	flu
22	F	47906	dyspepsia
33	F	47905	bronchitis
52	F	47905	flu
54	M	47302	gastritis
60	M	47302	Flu
60	M	47304	dyspepsia
64	F	47304	dyspepsia

(d) The bucketized table

Sex	Age	Zipcode	Disease
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	dysp.
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	Bron.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	Flu
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	dysp.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	dysp.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	gast.

(e) Multiset-based generalization

Sex	Age	Zipcode	Disease
22	F	47906	flu
22	M	47905	flu
33	F	47906	dysp.
52	F	47905	bron.
54	M	47302	Dysp.
60	F	47304	Gast.
60	M	47302	Dysp.
64	M	47304	flu

(f) One-attribute-per-column slicing

(Age,Sex)	(Zipcode,Disease)
(22,M)	(47905,flue)
(22,F)	(47906,dysp.)
(33,F)	(47905,bron.)
(52,F)	(47905,flu)
(54,M)	(47304,gast.)
(60,M)	(47302,flu)
(60,M)	(47302,dysp.)
(64,F)	(47304,dysp.)

(g) The sliced table

Table 1: An original microdata table and its anonymized versions using various anonymization techniques

2.2 Formalization of Slicing

Assume T is the microdata table. T having d attributes: d = {A_1, A_2, A_3, …, A_d} and their element domains are {D[A_1], D[A_2], D[A_3], …., D[A_d]}. A row t ∈ T can be represented as t = (t [A1], t [A2], ..., t[Ad]) where t[Ai] ($1 \leq i \leq d$) is the Ai value of t.

Definition 1: (Attribute columns and partitions). An element partition having of several subsets of A, such that each attribute fits to exactly one subset. Each subset of attributes are called as columns. Specifically, let there be c columns C1, C2, . . . ,Cc, then ∪c i=1Ci = A and for any $1 \leq i1 \neq i2 \leq c$, Ci1 ∩ Ci2 = ∅.

For easiness of conversation, we think only one sensitive attribute S. If in case the data comprises multiple sensitive attributes, one could whichever consider them separately or consider their joint distribution. Exactly one of the c columns contains S. Without loss of overview, let the column that having S be the last column Cc. This column is moreover called the sensitive column. All the other columns {C1,C2, . . . ,Cc−1} having only QI attributes.

Definition 2: (Tuple partition and buckets). A tuple partition contains of few subsets of T, such that every tuple pertaining to exactly one subset. Every subset of tuples is called as a bucket. Specifically, let there is a b buckets B1, B2, . . . , Bb, then ∪b i=1Bi = T and for any $1 \leq i1 \neq i2 \leq b$, Bi1 ∩ Bi2 =∅.

Definition 3 (Slicing). Specified a microdata table T, a slicing of T is specified by an attribute partition and a tuple partition.

For example, Table 1(e) and Table 1(f) are 2 sliced tables. In the Table 1(e), the attribute partition is {{Age}, {Sex}, {Zipcode}, {Disease}} and the other tuple partition is {{t1, t2, t3, t4}, {t5, t6, t7, t8}}. In Table 1(f), the attribute partition is {{Age, Sex}, {Zipcode, Disease}} and the tuple partition is {{t1, t2, t3, t4}, {t5, t6, t7, t8}}. Often times, slicing also comprises column generalization.

Definition 4 (Column Generalization). Displayed a microdata table T and a column Ci = {Ai1,Ai2, . . . ,Aij}, the column generalization for Ci is defined as a set of no overlying j dimensional areas that completely cover D[Ai1] × D[Ai2] × . . . × D[Aij]. A column generalization maps every value of Ci to the area in which the value is enclosed.

2.3 Comparison with Generalization

Nowadays few recoding methods are available for generalization in local systems these recoding techniques will preserve more information in the local systems. In the local recoding systems, they first cluster the tuples in buckets, after that each bucket one value attribute is replaces with generalized values. This recoding is local, because this generalization may be done differently in another tuples, even though if the same values are appears in the different bucket.

We now describing that slicing preserving more information compare with the local recoding technique. Assume uses same tuple partition is used. We will reach this by showing Slicing is better than the following enrichment is better than local coding approach. Instead of using a generalized value to replace more precise attribute values, one use the multiset of precise values in each and every bucket. For example, Table 1(b) was a generalized table, and Table 1(d) was the result of utilizing multiset of exact values rather than generalized values. For the attribute Age of the first bucket, we will use the multiset of perfect values {22,22,33,52} instead the generalized interval [22 − 52]. The multiset of particular values delivers lot of information about the spreading of values in every attribute than the generalized interval. Therefore, using multiset of correct values reserves more information than generalization.

Another essential benefit of slicing is its capability to handlebar high-dimensional data. By segregating the attributes into columns, slicing decreases the dimensionality of the data. Every column in the table can be watched as a sub-table with a minor dimensionality. Slicing is different from the method of publishing multiple independent sub-tables in that these sub-tables are connected by the buckets in slicing.

2.4 Comparison with Bucketization

To do comparison with slicing with bucketization, we initially note that bucketization can be watched as a distinctive case of slicing, where there are accurately two columns: one column encloses only the SA, and the other comprises all the QIs. The benefit of doing slicing on bucketization can be understood follows. First, by segregating attributes into more than two columns, slicing could be used for prevent membership disclosure.

Second, dissimilar bucketization, which entails a clear separation of QI features and the sensitive attribute, slicing is used without such a parting. For dataset such as the survey data, one often cannot clearly separate QIs from SAs because there is no single external public database that one can use to determine which attributes the adversary already knows. Slicing can be useful for such data. Finally, by allowing a column to contain both some QI attributes and the most important attribute, attribute correlations between the sensitive attribute and the QI attributes are conserved. For example, in Table 1(f), Zipcode and Disease form one column, permitting inferences about their associations. Attribute correlations are important utility in the data publishing. To workloads that consider attributes in parting, one can simply issue two tables, one comprising all QI attributes and one consist the sensitive attribute.

2.4 Privacy Threats

When publishing microdata there are three types of privacy disclosure threats: the 1) membership disclosure, whenever the dataset needs to be published, it needs to select from large population and the selection criteria sensitive data like a particular disease values. One desires to prevent opponents from access whether one's record is included in the published dataset or not.

Second type is identity disclosure, which arises when a discrete is linked to a specific record in the released table. In few situations, one needs to keep from identity disclosure when the opponent is indefinite of membership. In this situation, defense against membership disclosure supports protect against identity disclosure. In other conditions, some opponent may previously know that an entity's record is in the distributed dataset, in which situation, membership disclosure protection whichever does not apply or is inadequate.

The third type of disclosure is attribute disclosure, this arises whenever new information about some individuals is exposed, i.e., the unconfined data will make it possible to suppose the attributes of an individual exactly than it could be probable before to release. Similar to the case of identity disclosure, we need to deliberate opponents who already know the membership information. Identity disclosure signs to attribute disclosure. A discrete is re-identified, once there is distinctiveness disclosure and the corresponding sensitive value is revealed. Attribute disclosure can arise with or without identity disclosure, e.g., whenever the sensitive values of all matching tuples are the same.

3. SLICING ALGORITHAM

Here we are presenting an effective slicing algorithm to accomplish ℓ-diverse slicing. IN given microdata table T and two parameters c and ℓ, the algorithm calculates the sliced table that contains c columns and fulfills the privacy requirement of ℓ-diversity.

Our algorithm contains three phases: column generalization, attribute partitioning and tuple partitioning. Now we define the three phases.

4.1 Attribute Partitioning

Our algorithm segregates the attributes so that highly interrelated attributes are arranged in the same column. This is somewhat good for both activities like utility and privacy. In the case of data utility, gathering highly correlated attributes preserves the correlations among those elements. In the case of privacy, in the relationship of uncorrelated attributes shows higher identification threats than the relationship of highly correlated attributes cause the overtone of uncorrelated attribute values is more less numerous and therefore more recognizable. Therefore, it is somewhat better to disruption the relationships between uncorrelated elements, while defensive privacy. In this section, we first analyze the connections between sets of attributes and then group attributes based on their correlations.

4.2 Column Generalization

In the second section, tuples are generalized to fulfill some common occurrence requirement. We wish to concentrate on column generalization is not a crucial phase in our algorithm. As shown by Tao and Xiao, bucketization 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 1: The tuple-partition algorithm

It is providing the same level of privacy protection as generalization, with the attribute disclosure.

4.3 Tuple Partitioning

In the tuple separating section, tuples are divided into buckets. For tuple partition we have to change the Mondrian algorithm Dissimilar Mondrian k-anonymity, no generalization is performed to the tuples; we have to use Mondrian for the determination of segregating tuples into buckets.

The main aim of the tuple-partition algorithm is to validate whether a sliced table satisfies ℓ-diversity (line 5). Figure 2 shows a explanation of the diversity-check algorithm. For every 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] ∪ {hp(t,B),D(t,B)i}.
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 2: The diversity-check algorithm

Tuple t, the algorithm conserves a list of statistics L[t] about its matching buckets. Each element in the list L[t] comprises statistics about one matching bucket B: the matching possibility p (t, B) and the distribution of candidate sensitive values D (t,B).

4. CONCLUSION

This paper provides a new technique called slicing to providing privacy microdata publishing. Slicing overwhelms the limitations of bucketization and generalization and preserves better efficacy while securing against privacy threats. We demonstrate how to usage slicing to stop attributes disclosure and membership disclosure. Our experiment displays that slicing preserves best data utility than generalization and is more efficient than bucketization in workloads concerning the sensitive attribute.

The general technique proposed by this work is that:
before doing data anonymization, somebody can analyze the characteristics of data and they will use these characteristics in data anonymization. The basis is that one can design best data anonymization techniques when we know the data better.

5. REFERENCES

[1] C. Aggarwal. On k-anonymity and the curse of dimensionality. In VLDB, pages 901–909, 2005.
[2] A. Asuncion and D. Newman. UCI machine learning repository, 2007
[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In PODS, pages 128–138, 2005.
[4] J. Brickell and V. Shmatikov. The cost of privacy:
destruction of data-mining utility in anonymized data publishing. In KDD, pages 70–78, 2008.
[5] B.-C. Chen, R. Ramakrishnan, and K. LeFevre. Privacy skyline: Privacy with multidimensional adversarial knowledge. In VLDB, pages 770–781, 2007.
[6] H. Cramt'er. Mathematical Methods of Statistics. Princeton, 1948.
[7] I. Dinur and K. Nissim. Revealing information while preserving privacy. In PODS, pages 202–210, 2003.[35] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In VLDB, pages 139–150, 2006.
[8] X. Xiao and Y. Tao. Output perturbation with query relaxation. In VLDB, pages 857–869, 2008.
[9] Y. Xu, K. Wang, A. W.-C. Fu, and P. S. Yu. Anonymizing transaction databases for publication. In KDD, pages 767–775, 2008.

AUTHOR PROFILE

Mr. G. Sai Raghunath is currently pursuing M.Tech in the Department of Computer Science & Engineering, Visvesvaraya College of Engineering and Technology, M.P Patelguda, Ibrahimpatnam (M), Ranga Reddy(D), India. His research interests include Data Security.

Sri Dr. Bhaludra Raveendranadh Singh working as Associate Professor & Principal in Visvesvaraya College of Engineering and Technology. He obtained M.Tech, Ph.D(CSE)., is a young, decent, dynamic Renowned Educationist and Eminent Academician, has overall 20 years of teaching experience in different capacities. He is a life member of CSI, ISTE and also a member of IEEE (USA).

Ms's. Sangeetha M working as Assoc. Professor & HOD (CSE). She has completed bachelor of technology from Swamy Ramananda Theertha Institute of Science & Technology and Post-graduation from Jawaharlal Nehru Technological University,Kakinada campus and is having 12 years of teaching experience.