A NOVEL TECHNIQUE TO PREVENT IDENTIFYING SENSITIVE MICRO DATA IN DATAMINING

[1]K.L.Prathyusha and [2]K.Venkata Ramaiah

[1,2]Dept. of CSE, Chebrolu Engineering college, Chebrolu, Guntur.dt, AP, India

Abstract— The existing methods could not give a satisfactory result some has results a loss of information and some does not prevent the membership disclosure. In this paper, we present a new idea slicing, which partitions the data both horizontally and vertically. We justify that slicing preserves the data integrity and gives the member protection even it can handle high dimensional data. This can be used in protection of attribute disclosure and develop an algorithm to obey the ℓ-diversity requirement. Our experiments show that slicing gives a better and effective utility better than the existing one.

I. INTRODUCTION

Privacy-preserving publishing of microdata has been studied extensively in recent years. Microdata contain records each of which contains information about an individual entity, such as a person, a household, or an organization. Several microdata anonymization techniques have been proposed. The most popular ones are generalization for k-anonymity and bucketization for ℓ-diversity. In both approaches, attributes are partitioned into three categories: (1) some attributes are identifiers that can uniquely identify an individual, such as Name or Social Security Number; (2) some attributes are Quasi-Identifiers (QI), which the adversary may already know (possibly from other publicly-available databases) and which, when taken together, can potentially identify an individual, e.g., Birth- date, Sex, and Zipcode; (3) some attributes are Sensitive Attributes (SAs), which are unknown to the adversary and are considered sensitive, such as Disease and Salary.

The basic idea of slicing is to break the association cross columns, but to preserve the association within each column. This reduces the dimensionality of the data and preserves better utility than generalization and bucketization. Slicing preserves utility because it groups highly-correlated attributes together, and preserves the correlations between such attributes. Slicing protects privacy because it breaks the associations between uncorrelated attributes, which are infrequent and thus identifying. Note that when the dataset contains QIs and one SA, bucketization has to break their correlation; slicing, on the other hand, can group some QI attributes with the SA, preserving attribute correlations with the sensitive attribute.

In both generalization and bucketization, one first removes identifiers from the data and then partitions tuples into buckets. The two techniques differ in the next step. Generalization transforms the QI-values in each bucket into “less specific but semantically consistent” values so that tuples in the same bucket cannot be distinguished by their QI values. In bucketization, one separates the SAs from the QIs by randomly permuting the SA values in each bucket. The anonymized data consists of a set of buckets with permuted sensitive attribute values.

In this paper, we introduce a novel data anonymization technique called slicing to improve the current state of the art. Slicing partitions the dataset both vertically and horizontally. Vertical partitioning is done by grouping attributes into columns based on the correlations among the attributes. Each column contains a subset of attributes that are highly correlated. Horizontal partitioning is done by grouping tuples into buckets. Finally, within each bucket, values in each column are randomly permutated (or sorted) to break the linking between different columns.

II. EXISTING SYSTEM

First, many existing clustering algorithms (e.g., k- means) requires the calculation of the “centroids”. But there is no notion of “centroids” in our setting where each attribute forms a data point in the clustering space. Second, k-medoid method is very robust to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the order in which the data points are examined does not affect the clusters computed from the k-medoid method.

A. Disadvantages

Existing anonymization algorithms can be used for column generalization, e.g.,Mondrian . The algorithms can be applied on the sub-table containing only attributes in one column to ensure the anonymity requirement.

Existing data analysis (e.g., query answering) methods can be easily used on the sliced data.

Existing privacy measures for membership disclosure protection include differential privacy and presence.

III. Proposed System

We present a novel technique called slicing, which partitions the data both horizontally and vertically. We show that slicing preserves better data utility than generalization and can be used for membership disclosure protection. Another important advantage of slicing is that it can handle high-dimensional data. We show how slicing can be used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data that obey the ℓ-diversity requirement. Our workload experiments confirm that slicing preserves better utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute.

B. Advantages

We introduce a novel data anonymization technique called slicing to improve the current state of the art.

We show that slicing can be effectively used for preventing attribute disclosure, based on the privacy requirement of ℓ-diversity.

We develop an efficient algorithm for computing the sliced table that satisfies ℓ-diversity. Our algorithm partitions attributes into columns, applies column generalization, and partitions tuples into buckets. Attributes that are highly-correlated are in the same column.

We conduct extensive workload experiments. Our results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original data (which may overfit the model). Our experiments also show the limitations of bucketization in membership disclosure protection and slicing remedies these limitations.

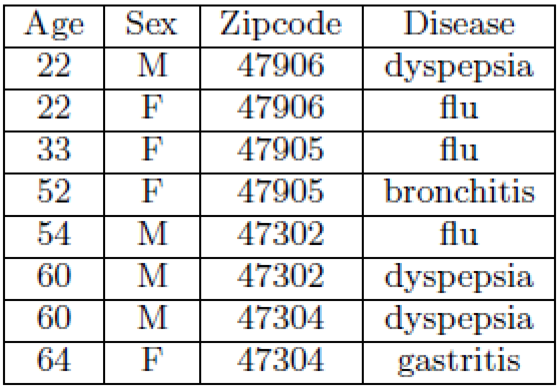
IV. Modules and its description

1) Module Description

Original Data
Generalized Data
Bucketized Data
Multiset-based Generalization Data
One-attribute-per-Column Slicing Data
Sliced Data

Original Data:

We conduct extensive workload experiments. Our results confirm that slicing preserves much better data utility than generalization. In workloads involving the sensitive attribute, slicing is also more effective than bucketization. In some classification experiments, slicing shows better performance than using the original data.

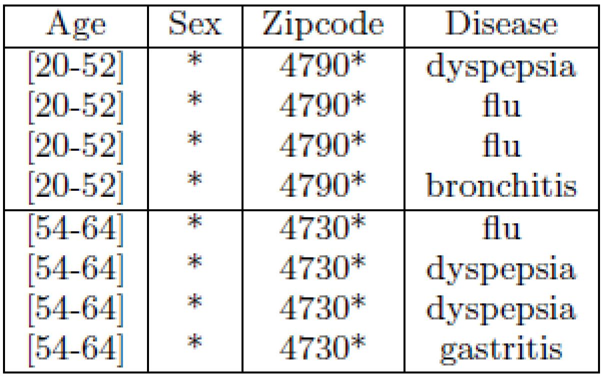


Age	Sex	Zipcode	Disease
22	M	47906	dyspepsia
22	F	47906	flu
33	F	47905	flu
52	F	47905	bronchitis
54	M	47302	flu
60	M	47302	dyspepsia
60	M	47304	dyspepsia
64	F	47304	gastritis

(a) The original table

Generalized Data:

Generalized Data, in order to perform data analysis or data mining tasks on the generalized table, the data analyst has to make the uniform distribution assumption that every value in a generalized interval/set is equally possible, as no other distribution assumption can be justified. This significantly reduces the data utility of the generalized data.



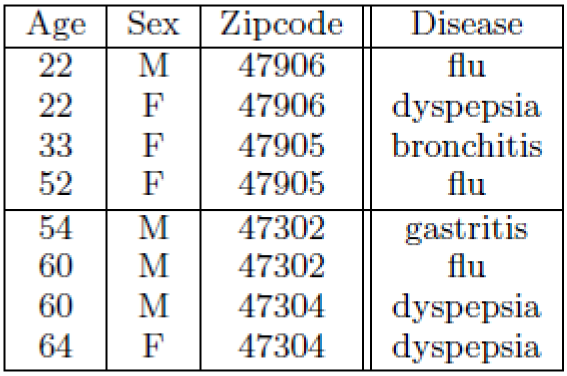
Age	Sex	Zipcode	Disease
[20-52]	*	4790*	dyspepsia
[20-52]	*	4790*	flu
[20-52]	*	4790*	flu
[20-52]	*	4790*	bronchitis
[54-64]	*	4730*	flu
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	dyspepsia
[54-64]	*	4730*	gastritis

(b) The generalized table

Bucketized Data:

We show the effectiveness of slicing in membership disclosure protection. For this purpose, we count the number of fake tuples in the sliced data. We also compare the number of matching buckets for original tuples and that for fake tuples.

Our experiment results show that bucketization does not prevent membership disclosure as almost every tuple is uniquely identifiable in the bucketized data.

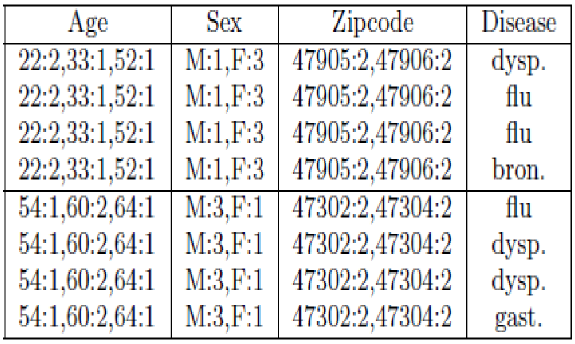


Age	Sex	Zipcode	Disease
22	M	47906	flu
22	F	47906	dyspepsia
33	F	47905	bronchitis
52	F	47905	flu
54	M	47302	gastritis
60	M	47302	flu
60	M	47304	dyspepsia
64	F	47304	dyspepsia

(c) The bucketized table

Multiset-based Generalization Data:

We observe that this multiset-based generalization is equivalent to a trivial slicing scheme where each column contains exactly one attribute, because both approaches preserve the exact values in each attribute but break the association between them within one bucket.

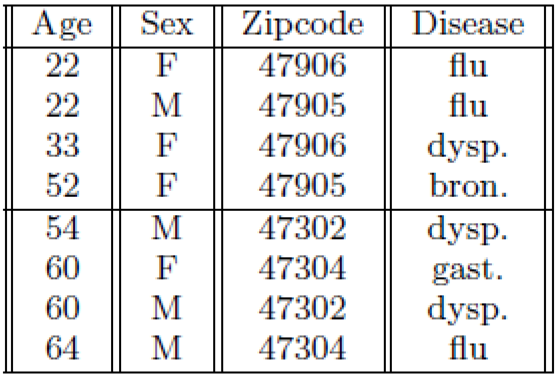


Age	Sex	Zipcode	Disease
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	dysp.
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	flu
22:2,33:1,52:1	M:1,F:3	47905:2,47906:2	bron.
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	flu
54:1,60:2,64:1	M:3,F:1	47302:2,47304:2	dysp.
54:1,60:2,64:1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

(d) Multiset-based generalization

One-attribute-per-Column Slicing Data:

We observe that while one-attribute-per-column slicing preserves attribute distributional information, it does not preserve attribute correlation, because each attribute is in its own column. In slicing, one groups correlated attributes together in one column and preserves their correlation. For example, in the sliced table shown in Table correlations between Age and Sex and correlations between Zipcode and Disease are preserved. In fact, the sliced table encodes the same amount of information as the original data with regard to correlations between attributes in the same column.

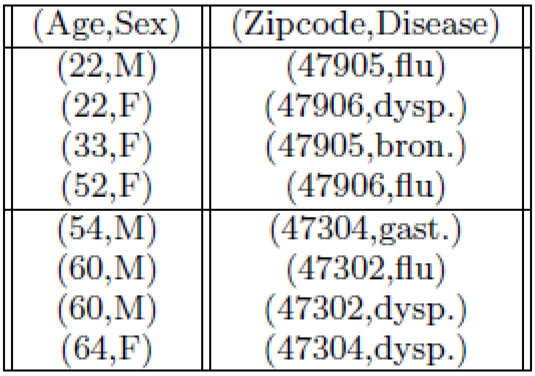


Age	Sex	Zipcode	Disease
22	F	47906	flu
22	M	47905	flu
33	F	47906	dysp.
52	F	47905	bron.
54	M	47302	dysp.
60	F	47304	gast.
60	M	47302	dysp.
64	M	47304	flu

(e) One-attribute-per-column slicing

Sliced Data:

Another important advantage of slicing is its ability to handle high-dimensional data. By partitioning attributes into columns, slicing reduces the dimensionality of the data. Each column of the table can be viewed as a sub-table with a lower dimensionality. Slicing is also different from the approach of publishing multiple independent sub-tables in that these sub-tables are linked by the buckets in slicing.



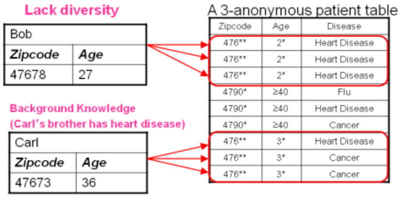
(Age,Sex)	(Zipcode,Disease)
(22,M)	(47905,flu)
(22,F)	(47906,dysp.)
(33,F)	(47905,bron.)
(52,F)	(47906,flu)
(54,M)	(47304,gast.)
(60,M)	(47302,flu)
(60,M)	(47302,dysp.)
(64,F)	(47304,dysp.)

(f) The sliced table

V. System Design and IMPLEMENTATION

Architecture of the system

There are several types of recordings for generalization. The recoding that preserves the most information is local recoding. In local recoding, one first groups tuples into buckets and then for each bucket, one replaces all values of one attribute with a generalized value. Such a recoding is local because the same attribute value may be generalized differently when they appear in different buckets.



Bob	
Zipcode	*Age*
47678	27

Carl	
Zipcode	*Age*
47673	36

A 3-anonymous patient table

Zipcode	Age	Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
476**	2*	Heart Disease
4790*	≥40	Flu
4790*	≥40	Heart Disease
4790*	≥40	Cancer
476**	3*	Heart Disease
476**	3*	Cancer
476**	3*	Cancer

System Architecture

Algorithm Used:

Slicing Algorithms:

Our algorithm consists of three phases: attribute partitioning, column generalization, and tuple partitioning. We now describe the three phases.

```
Algorithm tuple-partition(T, ℓ)
Q = {T}; SB = ∅.
whileQ is not empty
remove the first bucket B from Q; Q = Q − {B}.
splitB into two buckets B1 and B2, as in Mondrian.
if diversity-check(T, Q ∪ {B1,B2} ∪SB, ℓ)
Q = Q ∪ {B1,B2}.
else SB = SB ∪ {B}.
return SB.
Algorithm diversity-check(T,T_, ℓ)
for each tuple t ∈T, L[t] = ∅.
for each bucket B in T_
recordf(v) for each column value v in bucket B.
for each tuple t ∈T
calculatep(t,B) and find D(t,B).
L[t] = L[t] ∪ {hp(t,B),D(t,B)i}.
for each tuple t ∈T
calculatep(t, s) for each s based on L[t].
ifp(t, s) ≥ 1/ℓ, return false.
return true.
```

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. Although column generalization is not a required phase, it can be useful in several aspects. First, column generalization may be required for identity/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/bucketization where each tuple can belong to only one equivalence-class/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.

VI. RESULT AND DISCUSSION

This paper presents a new approach called slicing to privacy-preserving microdata publishing. Slicing overcomes, the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. We illustrate how to use slicing to prevent attribute disclosure and membership disclosure. Our experiments show that slicing preserves better data utility than generalization and is more effective than bucketization in workloads involving the sensitive attribute. The general methodology proposed by this work is that: before anonymzing the data, one can analyze the data characteristics and use these characteristics in data anonymization.

VII. CONCLUSION

Slicing is a promising technique for handling high-dimensional data. By partitioning attributes into columns,we protect privacy by breaking the association of uncorrelated attributes and preserve data utility by preserving the association between highly-correlated attributes.

VIII. ENHANCEMENT

A malicious data miner may have access to differently perturbed copies of the same data through various means, and may combine these diverse copies to jointly infer additional information about the original data that the data owner does not intend to release. Preventing such diversity attacks by providing MLT-PPDM services.

REFERENCES

[1] C. Aggarwal. On k-anonymity and the curse ofdimensionality. In VLDB, pages 901–909, 2005.

[2] A. Asuncion and D. Newman. UCI machine learningrepository, 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 multidimensionaladversarial 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 whilepreserving privacy. In PODS, pages 202–210, 2003.

[7] C. Dwork. Differential privacy. In ICALP, pages 1–12,2006.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith.Calibrating noise to sensitivity in private dataanalysis. In TCC, pages 265–284, 2006.

[9] J. H. Friedman, J. L. Bentley, and R. A. Finkel. Analgorithm for finding best matches in logarithmicexpected time. TOMS, 3(3):209–226, 1977.

Ms.K.L.PRATHYUSHA is a student of Chebrolu Engineering College, Chebrolu. Presently she is pursuing her M.Tech [CSE] from this college and she received her B.Tech from LITAM College, affiliated to JNTUK University, in the year 2009.

Mr.K.VenkataRamaiah ,excellent teacher Received M.Tech (CSE) from Bharat University, B.tech from JNT University, Hyderabad, is working as Associate Professor and HOD, Department of CSE, M.Tech Computer science engineering , Chebrolu Engineering college. He has 11years of teaching experience in various engineering colleges.