New anonymization technique for Handling high dimensional data and secure publishing by using L-diversity slicing approach

G.Anil Srinivas  
Sr.Asst. Prof  
Vishnu Institute of Engg & Tech  
Bhimavaram, India

M.Venkatesh  
Department of CSE  
Vishnu Institute of Engg & Tech  
Bhimavaram, India

ABSTRACT: In recent years, for many kinds of structured data, data anonymization techniques have been subject of research. There are several anonymization techniques such as Generalization and Bucketization for privacy conserving data publishing. But Generalization & Bucketization has several drawbacks. Generalization loses considerable amount of information, especially for high dimensional data, Bucketization, on the other hand, does not prevent membership disclosure and does not apply for data that do not have a clear separation between quasi-identifying attributes and sensitive attributes. In this paper, a novel technique called slicing is proposed for handling dimensional data and secure publishing. This technique partitions the data both horizontally and vertically.

Keywords: - Microdata, High dimensional data, data anonymization, data publishing, data security, generalization and bucketization.

Introduction:  
Microdata are information at the level of individual respondents. For instance, a national census (A periodic count of the population) might collect age, home address, educational level, employment status, and many other variables, recorded separately for every person. Anonymization means without out a name or nameless. Data anonymization means making some data anonymous. By using Data anonymization data holder can putting some data anonymously in database for data recipient, for providing privacy to reduce the possibility of identifying sensitive information about individuals. Microdata are stored in a table, and each record (row) corresponds to one individual. Each record has a number of attributes, which can be divided into the following three categories: Identifiers, Quasi Identifiers (QI), Sensitive Attributes (SAs).
Background:

Present data anonymization techniques such as Generalization for k-anonymity losses considerable amount of information for high-dimensional data, so it cannot be applied to high dimensional data and it does not protect against attacks based on background knowledge.

Bucketization does not prevent membership disclosure, because bucketization publishes the QI values in their original forms, an adversary can find out whether an individual has a record in the published data or not. It requires a clear separation between QI’s and SA’s, but in many data sets, it is unclear which attributes are QI’s and which are SA’s and it breaks the attributes correlations between the QI’s and the SA’s. This paper L-diversity slicing introduces to overcome all the drawbacks of bucketization and generalization.

Motivation:

The generalization for k-anonymity loses considerable amount of information for high-dimensional data. This is due to the following three reasons.

First, generalization for k-anonymity suffers from the curse of dimensionality. It is useful for fewer amounts of data in each bucket which are closed to each other, so less amount of data the generalizing the records would not lose too much information. However, in high dimensional data, most data points have similar distances with each other, forcing a great amount of generalization to satisfy k-anonymity even for relatively small k’s.

Second, 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.

Third, it doesn’t prevent the attacks from background knowledge.

While bucketization has better data utility than generalization, it has several limitations.

First, bucketization does not prevent membership disclosure, because bucketization publishes the QI values in their original forms, an adversary can find out whether an individual has a record in the published data or not. In present situation 87 percent of the individuals in the United States can be uniquely identified using only three attributes (Birth date, Sex, and Zip code). A Microdata (e.g., census data) usually contains many other attributes besides those three attributes. This means that the membership information of most individuals can be inferred from the bucketized table.

Second, bucketization requires a clear separation between QIs and SAs. However, in many data sets, it is unclear which attributes are QIs and which are SAs.

Third, to separating sensitive attribute from the QI attributes, bucketization breaks the attribute correlations between the QIs and the SAs.
Proposed system:

Proposed system introduces a novel data anonymization technique called slicing to improve the current state of the art. Slicing partitions the data set 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 permuted (or sorted) to break the linking between different columns. The basic idea of slicing is to break the association across 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 data set 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 existing systems for preserving privacy, k-anonymity is used for generalization and l-diversity is used for bucketization. But in both approaches have limitation. To overcome those limitations there is a new algorithm called l-diverse slicing.

ℓ-Diverse Slicing:

ℓ-Diverse Slicing is a novel algorithm for slicing technique. This algorithm is implemented from previously exiting paradigms such as k-anonymity and l-diversity.

Let $P(t, B)$ be the probability that $t$ is in bucket $B$.

$P(t, s)$, is the probability that $t$ takes a sensitive value $s$. $P(t, s)$ is calculated using the law of total probability.

Computing $p(t, B)$:

Given a tuple $t$ and a sliced bucket $B$, the probability that $t$ is in $B$ depends on the fraction of $t$'s column values that match the column values in $B$. $P(s/t, B)$ is the probability that $t$ takes sensitive value $s$ given that $t$ is in bucket $B$. Then according to the law of total probability, the probability $P(t, s)$ is

$$p(t, s) = \sum_{B} p(t, B) P(t, s)$$

If some column value of $t$ does not appear in the corresponding column of $B$, it is certain that $t$ is not in $B$. In general, bucket $B$ can potentially match $|B|^c$ tuples, where $|B|$ is the number of tuples in $B$. 

540
Let $f_i(t, B)$ ($1 \leq I \leq c - 1$) be the fraction of occurrences of $t[C_i]$ in $B[C_i]$ and let $f_{c}(t,B)$ be the fraction of occurrences of $t[C_c - \{S\}]$ in $B[C_c - \{S\}]$. Note that, $C_c - \{S\}$ is the set of QI attributes in the sensitive column.

$f_i(t, B)$ measures the matching degree on column $C_i$, between tuple $t$ and bucket $B$. Because each possible candidate tuple is equally likely to be an original tuple, the matching degree between $t$ and $B$ is the product of the matching degree on each column, i.e., $f(t, B)=\prod_{1 \leq I \leq c} f_i(t, B)$. Note that $\sum B f(t, B) = 1$ and when $B$ is not a matching bucket of $t$.

Tuple $t$ may have multiple matching buckets, $t$’s total matching degree in the whole data is $f(t) = \sum B f(t, B)$. The probability that $t$ is in bucket $B$ is

$$p(t, B) = \frac{f(t, B)}{f(t)}$$

**Computing $p(t, B)$:**

Suppose that $t$ is in bucket $B$, to determine $t$’s sensitive value, one needs to examine the sensitive column of bucket $B$. Since the sensitive column contains the QI attributes, not all sensitive values can be $t$’s sensitive value. Only those sensitive values whose QI values match $t$’s QI values are $t$’s candidate sensitive values.

Let $D(t,B)$ is the distribution of $t$’s “candidate sensitive values” in bucket $B$.

**Slicing algorithm:**

Microdata table $T$ and two parameters $C$ and $l$ the algorithm computes the sliced table that consists of $c$ columns and satisfies the privacy requirement of ‘$l$-diversity.

**Attribute Partitioning:**

The proposed system algorithm partitions attributes so that highly correlated attributes are in the same column. This is good for both utility and privacy. In terms of data utility, grouping highly correlated attributes preserves the correlations among those attributes. In terms of privacy, the association of uncorrelated attributes presents higher identification risks than the association of highly correlated attributes because the association of uncorrelated attribute values is much less frequent and thus more identifiable. Therefore, it is better to break the associations between uncorrelated attributes, in order to protect privacy. In
this phase, proposed system first computes the correlations between pairs of attributes and then cluster attributes based on their correlations.

Measures of Correlation: -

Two widely used measures of association are Pearson correlation coefficient and mean-square contingency coefficient. Pearson correlation coefficient is used for measuring correlations between two continuous attributes. Mean-square contingency coefficient is a chi-square measure of correlation between two categorical attributes. We choose to use the mean-square contingency coefficient because most of our attributes are categorical. Given two attributes A1 and A2 with domains \{V_{11}, V_{12}, \ldots, V_{1d1}\} and \{V_{21}, V_{22}, \ldots, V_{2d2}\} respectively. Their domain sizes are thus d1 and d2, respectively. The mean-square contingency coefficient between A1 and A2 is defined as

\[
\Phi^2(A_1,A_2) = \frac{1}{\min\{d_1, d_2\} - 1} \sum_{i=1}^{d_1} \sum_{j=1}^{d_2} \left( \frac{f_{ij} - f_i f_j}{f_i f_j} \right)^2
\]

Here, 
\(f_i\) is the fraction of occurrences of \(V_{1i}\) and 
\(f_j\) is the fraction of occurrences of \(V_{1j}\). 
\(f_{ij}\) is the fraction of occurrences of \(V_{1i}\) and \(V_{1j}\) in the data. 
Therefore, \(f_{i.}\) and \(f_{.j}\) are the marginal totals of \(f_{ij}\). 
\(f_i = \sum_{j=1}^{d_2} f_{ij}\) and \(f_j = \sum_{i=1}^{d_1} f_{ij}\)

Attribute Clustering: -

Having computed the correlations for each pair of attributes, proposed system uses clustering to partition attributes into columns. In proposed system algorithm, each attribute is a point in the clustering space. The distance between two attributes in the clustering space is defined as 
\(D(A_1,A_2) = 1 - \Phi^2(A_1,A_2)\), which is in between of 0 and 1. Proposed system chooses the k-medoid method for the following reasons.

Column Generalization: -

First, column generalization may be required for identity/membership disclosure protection. If a column value is unique in a 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. Second, when column generalization is applied, to achieve the same level of privacy against attribute disclosure, bucket sizes can be smaller

Tuple Partitioning:-
In the tuple partitioning phase, tuples are partitioned into buckets. Proposed system modifies the Mondrian algorithm for tuple partition. Unlike Mondrian k-anonymity, no generalization is applied to the tuples; proposed system use Mondrian for the purpose of partitioning tuples into buckets.

Table 1: Tuple partition Algorithm

<table>
<thead>
<tr>
<th>Algorithm tuple-partition (T, ℓ)</th>
<th>Algorithm diversity-check(T,T*, ℓ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Q = {T}; SB = ∅.</td>
<td>1. for each tuple t ∈ T, L[t] = ∅.</td>
</tr>
<tr>
<td>2. while Q is not empty</td>
<td>2. for each bucket B in T</td>
</tr>
<tr>
<td>3. Remove the first bucket B</td>
<td>3. record f(v) for each column value v in bucket B.</td>
</tr>
<tr>
<td>from Q; Q = Q − {B}.</td>
<td>4. for each tuple t ∈ T</td>
</tr>
<tr>
<td>4. Split B into two buckets B1</td>
<td>5. Calculate p(t, B) and find D(t, B).</td>
</tr>
<tr>
<td>and B2, as in Mondrian.</td>
<td>6. ( L[t] = L[t] \cup {p(t, B), D(t, B)} )</td>
</tr>
<tr>
<td>5. if diversity-check(T, Q ∪</td>
<td>7. for each tuple t ∈ T</td>
</tr>
<tr>
<td>{B1,B2} ∪ SB, ℓ)</td>
<td>8. Calculate p(t, s) for each s based on L[t].</td>
</tr>
<tr>
<td>6. Q = Q ∪ {B1,B2}.</td>
<td>9. if ( p(t, s) \geq 1/ℓ ), return false.</td>
</tr>
<tr>
<td>7. else SB = SB ∪ {B}.</td>
<td>10. Return true</td>
</tr>
<tr>
<td>8. Return SB.</td>
<td></td>
</tr>
</tbody>
</table>

Step 1: In the initial stage proposed system considers a queue of buckets Q and a set of sliced buckets SB. Initially Q contains only one bucket which includes all tuples and SB is empty. So \( Q = \{T\}; SB = ∅ \).

Step 2: In each iteration the algorithm removes a bucket from Q and splits the bucket into 2 buckets. \( Q = Q − \{B\} \) for \( ℓ \)-diversity check \( (T, Q ∪ \{B1,B2\} ∪ SB, ℓ) \); The main part of tuple partitioning algorithm is to check whether a sliced table satisfies \( ℓ \)-diversity.

Step 3: In the diversity check algorithm for each tuple t, it maintains a list of statistics \( L[t] \) contains Statistics about one matching bucket \( B \). \( t \in T, L[t] = ∅ \) The matching probability \( p(t, B) \) and the
distribution of candidate sensitive values $D(t, B)$.

Step 4: $Q = Q \cup \{B_1, B_2\}$ here two buckets are moved to the end of the Q.

Step 5: else $SB = SB \cup \{B\}$ in this step proposed system cannot split the bucket more so the bucket is sent to SB.

Step 6: Thus a final result return SB, here when Q becomes empty we have Computed the sliced table. The set of sliced buckets is SB .So, finally Return SB.

Conclusion

In this paper, a new approach called slicing to privacy-conserving micro data publishing. Slicing overcomes the limitations of generalization and bucketization and preserves better utility while protecting against privacy threats. Proposed system illustrates how to use slicing to prevent attribute disclosure and membership disclosure. 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 anonymizing the data, one can analyze the data characteristics and use these characteristics in data anonymization. The rationale is that one can design better data anonymization techniques when we know the data better.

REFERENCES


[3] Li Tiancheng “Privacy Preservation in Data Publishing and Sharing” August 2010 Purdue University

[4] Li Tianchung, Li inghui, Senior Member, IEEE, Jian Zhang, Member, IEEE, and Ian Molloy “Slicing: A New Approach for Privacy Preserving Data Publishing “


Authors Profile

Mr.G.Anil Srinivas is working as Sr.Asst.prof in computer science department at Vishnu institute of engineering and technology, bhimavaram, A.P, India his research interest includes Data mining, computer networks and semantic web. Email: anilsrinivas@yahoo.com

Mr.M.Venkatesh pursuing M.Tech C.S.E. in Vishnu institute of Engineering and technology, bhimavaram, Andhra Pradesh, India. His research interests include Data Mining, Email: lmanne.venkatesh@gmail.com