Performance Analysis of Clustering in Privacy Preserving Data Mining

Bipul Roy
NIELIT, Itanagar
E-Sector, Near Shiv Mandir
Naharlagun, Arunachal Pradesh-791110

Abstract:

Privacy is becoming an increasingly important issue in many data mining applications. This has triggered the development of many privacy preserving data mining techniques. A frequently used disclosure protection method is data perturbation. When used for data mining, it is desirable that perturbation preserves statistical relationships between attributes, while providing adequate protection for individual confidential data. Existing perturbation methods typically require that the statistical properties of the data can be specified with known distributions. We propose a tree-based perturbation method that can be easily used for perturbing data with knowing the underlying distributions. Our method employs a kd-tree technique to recursively partition a dataset into smaller subsets such that data records within each subset are more homogeneous after each partition. Once the partitioning process is completed, the confidential data in each subset are perturbed using microaggregation. An experimental study shows that our proposed method outperforms additive and multiplicative noise perturbation methods for clustering applications.

Keywords: Privacy, data mining, data perturbation, microaggregation, kd-trees.

1. Introduction

In recent years advances in technology facilitated collection and storage of vast amount of data. Many organizations, including large and small businesses and hospitals and government's bodies rely on data for day-to-day operations as well as marketing, planning and research purposes. Examples include criminal records used by law enforcement and national security agencies, medical records used for treatment and research purposes and shopping records used for marketing and enhancing business strategies. The benefits of the information extracted from such data can hardly be overestimated.

In order to resolve the conflict between data mining and privacy protection, researchers in the data mining community have proposed various methods. Agrawal and Srikant (2000) considered building a decision tree classifier from data where the confidential values have been perturbed. By using distribution reconstruction procedure, the authors were able to build classifiers whose accuracy is comparable to that of classifiers built with original data. There has been extensive research in the area of statistical databases (SDBs) on how to provide summary statistical information without disclosing individual's confidential data. The privacy issue arises in SDB when summary statistics are derived on very few (or a single) individuals data. In this case, releasing the summary statistics leads to disclosure of individual confidential data. The methods for preventing such disclosure can be broadly classified into two categories: query restriction, which prohibits queries that would reveal confidential data; and data perturbation, which alters individual data in a way such that the summary statistics remain approximately the same.

Privacy preserving data mining (PPDM) [1] refers to the area of data mining that seeks to safeguards sensitive information from unsolicited or unsanctioned disclosure. Most traditional data mining techniques analyze and model the dataset statistically, in aggregation, while privacy preservation

is primary concerned with protecting against disclosure of individual data records. This domain separation points to the technical feasibility of PPDM.

The process of grouping a set of objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. The clustering technique in which distance measure is used as a similarity measure is called distance based clustering technique. A de- tails survey on clustering techniques can be found in [15, 16]. Clustering is a widely used data mining technique in many applications such as customer behavior analysis, targeted marketing, and many others.

The main objective in privacy preserving data mining is to develop algorithms for modifying the original data in some way, so that the private data and private knowledge remain private even after the mining process.

The problem that arises when confidential information can be derived from released data by unauthorized users is also commonly called the database inference problem. We claim that a solution for such a problem requires two vitals techniques: anonymity [12, 13] to remove identifiers (e.g., names, social insurances numbers, addresses, etc.) in the first phase of privacy preserving protection, and data perturbation to protect some sensitive attributes (e.g. salary, age, etc.) since the released data, after removing identifiers, may contain other information can be linked with other datasets to re-identify

individuals or entries [14]. In this paper, we focus on the latter technique. Specifically, we consider the case in which confidential numerical attributes are distorted in order to meet privacy protect in clustering analysis.

To address privacy concerns in clustering analysis, we need to design specific data perturbation methods that enforce privacy without losing the benefit of mining. The proposed data perturbation methods that in the literature pertain to the context of statistical databases

[9, 10, 11, 12, 13]. They do not apply data clustering as they have limitations when the perturbed attributes are considered as a vector in the Euclidean space. For instance, let us suppose that some confidential attributes (e.g. salary, age) are rep- resented by the points in a 2-D discrete space for clustering analysis. If we distort these attributes using any perturbed methods proposed in the literature, the clusters obtained after perturbing the data would be very different from those mined from the original database. The main problem is that many points would move from one cluster to another jeopardizing the notion of similarity between data points in the global space. Consequently, this introduces the problem of misclassification. Therefore, the perturbation has to be uniformly applied to all attributes to guarantee safeguarding the global distances between data points, or even to slightly modify the distance between some points.

1.1 Our contributions

In this paper, we introduce a tree-based data perturbation (kd-tree) that distorts confidential numerical attributes in order to meet privacy protection in clustering analysis. Our main contributions in this paper are as follows:

A potential problem with perturbation trees is that when the boundaries between data clusters are not axis parallel, perturbation trees could create groups within which data points are dissimilar.

To alleviate this problem:

If the mid-range value coincides with the value of the attribute in one or more records, then the mid-range value (i.e., the threshold) can be randomly reduced or increased by epsilon so that the records lie on one partition.

For more effectiveness first divide the whole dataset into large blocks using perturbation tree, and then microaggregate the data within each blocks using MMA.

2. Background of the Work

2.1 Classification of Privacy Preserving Techniques

There are many approaches adopted for privacy preserving data mining [2].

It can be classified on the following dimensions:

2.1.1 Suppression

Privacy can be preserved by simply suppressing all sensitive data before any disclosure or computation occurs. Given a database, we can suppress specific attributes in particular records as dictated by our privacy policy.

For a partial suppression, an exact attribute value can be replaced with a less informative value by rounding (e.g. $23.45 to $20.00), top-coding (e.g.age above 70 is set to 70), generalization (e.g. address to zip code), by using intervals (e.g. age 23 to 20-25, name ‘Johnson’ to ‘J-K') etc. Often the privacy guarantee trivially follows from the suppression policy. However, the analysis may be difficult if the choice of alternative suppressions depends on the data being suppressed, or if there is dependency between disclosed and suppressed data. Suppression cannot be used if data mining requires full access to the sensitive values.

2.1.2 Randomization

Suppose there is one central server, e.g. of a company, and many customers, each having a small piece of information. The server collects the information and performs data mining to build an aggregate data model. The randomization approach [28] protects the customers' data by letting them randomly perturb their records before sending them to the server, taking away some true information and introducing some noise. At the server's side, statistical estimation over noisy data is employed to recover the aggregates needed for data mining. Noise can be introduced e.g. by adding or multiplying random values to numerical attributes [5] or by deleting real items and adding bogus" items to set-valued records [26, 29]. Given the right choice of the method and the amount of randomization, it is sometimes possible to protect individual values while estimating the aggregate model with relatively high accuracy.

2.1.3 Cryptography

The cryptographic approach to PPDM assumes that the data is stored at several private parties, who agree to disclose the result of a certain data mining computation performed jointly over their data. The parties engage in a cryptographic protocol, i.e. they exchange messages encrypted to make some operations efficient while others computationally intractable. In effect, they “blindly" run their data mining algorithm. The assumptions include restrictions on the input data and permitted disclosure, the computational hardness of certain mathematical operations such as factoring a large integer, and the adversarial potential of the parties involved: the parties may be passive (honest-but-curious, running the protocol correctly but taking advantage of all incoming messages) or malicious (running a different protocol), some parties may be allowed to collude (represent a single adversary) etc.

2.1.4 Summarization

This approach to PPDM consists of releasing the data in the form of a "summary" that allows the (approximate) evaluation of certain classes of aggregate queries while hiding the individual records. In a sense, summa- rization extends randomization, but a summary is often expected to be much shorter, ideally of sub-linear size with respect to the original dataset. The idea goes back to statistical databases, where two summarization techniques were studied and widely applied: sampling and tabular data representation [9, 30]. Sampling corresponds to replacing the private dataset with a small sample of its records, often combined with suppression or perturbation of their values to prevent re-identification. Tabular representation summarizes

data in a collection of aggregate quantities such as sums, averages or counts, aggregated over the range of some attributes while other attributes are fixed, similarly to OLAP (On Line Analytical Processing) cubes. Verifying privacy guarantees for tabular data is challenging because of the potential for disclosure by inference.

2.2 Data Perturbation

The methods based on the data perturbation approach fall into two main categories known as probability distribution category and fixed data perturbation category. In the probability distribution category, the security control method replaces the original databases by another sample from the same distribution or by the distribution itself. On the other hand, the fixed data perturbation methods discussed in the literature has been developed exclusively for either numerical data or categorical data. These methods usually require that a dedicated transformed database is created for secondary use, and they have evolved from a simple method for a single attribute to multi attribute methods.

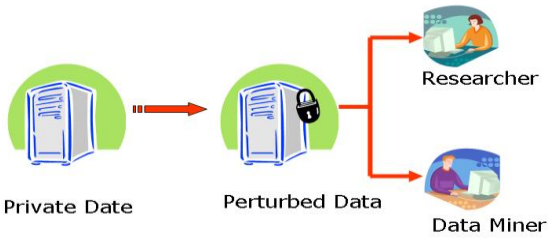

Figure 2.1: Data perturbation Technique

2.3 Microaggregation

Another approach to data perturbation is microaggregation (MA) [6]. Microaggregation is one of the most popular, studied and used microdata protection methods. It builds small clusters of at least k elements of v attributes and replaces the original records by the centroid of the cluster to which the records belong. The goal of a microaggregation method is to minimize the total Sum of Square Error. If a microaggregation method is applied to all the V attributes of the original dataset X at the same time; then, the resulting protected dataset X' satisfies the property of k-anonymity each protected record can correspond to at least k original records. However, in order to increase the statistical utility of the released (protected) information, statistical agencies usually split the whole dataset X in blocks of a few attributes, and then apply a microaggregation method to each block, independently. In this way, k-anonymity is not preserved any more. Univariate Microaggregation (UMA) MA perturbs data by aggregating confidential values, instead adding noise.

For a data set with a single confidential attribute univariate microaggregation (UMA) involves sorting records by the confidential attribute, grouping adjacent records into groups of small sizes, and replacing the individual confidential values in each group with the into groups average. Similar to SAN and MN, UMA causes bias in the variance of the confidential attributes, as well as in the relationships between attributes.

2.3.1 Multivariate Microaggregation (MMA)

Multivariate microaggregation (MMA) differs from UMA in that it groups data using a clustering technique that is based on a multi-dimensional distance measure. As a result, the relationships between attributes are expected to be better preserved. However, this benefit comes with a higher computational time complexity (O(N2) for a data set of N records), which could be inefficient for large data sets.

2.4 Clustering Analysis

2.4.1 Clustering Technique

Clustering techniques [15, 16] fall into a group of undirected data mining tools. The goal of undirected data mining is to discover structure in the data as a whole. There is no target variable to be predicted, thus no distinction is being made between independent and dependent variables. Clustering techniques are used for combining observed examples into clusters (groups) which satisfy two main criteria:

- each group or cluster is homogeneous; examples that belong to the same group are similar to each other.
- each group or cluster should be different from other clusters, that is, examples that belong to one cluster should be different from the examples of other clusters.

Depending on the clustering technique, clusters can be expressed in different ways:

- Identified clusters may be exclusive, so that any example belongs to only one cluster.
- they may be overlapping; an example may belong to several clusters.
- they may be probabilistic, whereby an example belongs to each cluster with a certain probability.
- clusters might have hierarchical structure, having crude division of examples at highest level of hierarchy, which is then refined to subclusters at lower levels.

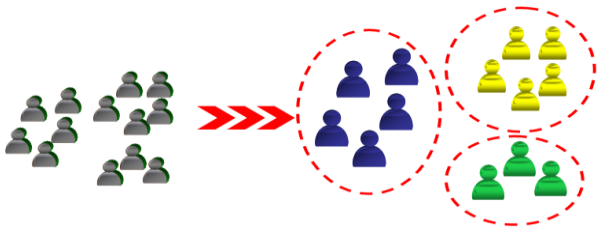

Figure 2.2: Clustering Technique

2.4.2 k-means algorithm

This algorithm has as an input a predefined number of clusters that is the k from its name. Means stands for an average, an average location of all the members of a particular cluster. When dealing with clustering techniques, one has to adopt a notion of a high dimensional space, or space in which orthogonal dimensions are all attributes from the table of data we are analyzing. The value of each attribute of an example represents a distance of the example from the origin along the attribute axes. Of course, in order to use this geometry efficiently, the values in the data set must all be numeric (categorical data must be transformed into numeric ones!) and should be normalized in order to allow fair computation of the overall distances in a multi-attribute space [8].

k-means algorithm [15] is a simple, iterative procedure, in which a crucial concept is the one of centroid. Centroid is an artificial point in the space of records which represents an average location of the particular cluster. The coordinates of this point are averages of attribute values of all examples that belong to the cluster.

The sketch of Partition Based Clustering- k-means algorithm is given as follows:

k-means algorithm:

1. Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the k centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

2.4.3 Important issues in automatic cluster detection

Most of the issues related to automatic cluster detection are connected to the kinds of questions we want to be answered in the data mining project, or data preparation for their successful application.

3. Concepts of KD TREE

3.1 Introduction

In computer science, a kd-tree (short for k-dimensional tree) [7] is a space partitioning data structure for organizing points in a k-dimensional space.

kd-trees are a useful data structure for several applications, such as searches involving a multidimensional search key (e.g. range searches and nearest neighbor searches). kd-trees are a special case of BSP (Binary Space Partitioning) trees.

3.2 Informal Description

The kd-tree is a binary tree in which every node is a k-dimensional point.

Every non-leaf node generates a splitting hyperplane that divides the space into two subspaces. Points left to the hyperplane represent the left sub- tree of that node and the points right to the hyperplane by the right subtree. The hyperplane direction is chosen in the following way: every node split to sub-trees is associated with one of the k-dimensions, such that the hyper plane is perpendicular to that dimension vector. So, for example, if for a particular split the “x" axis is chosen, all points in the subtree with a smaller “x" value than the node will appear in the left subtree and all points with larger “x" value will be in the right sub tree.

3.3 Operations on kd-trees

3.3.1 Construction

Since there are many possible ways to choose axis-aligned splitting planes, there are many different ways to construct kd-trees. The canonical method of kd-tree construction has the following constraints:

- As one moves down the tree, one cycles through the axes used to select the splitting planes. (For example, the root would have an x-aligned plane, the root's children would both have y-aligned planes, the root's grandchildren would all have z-aligned planes, and the next level would have an x-aligned plane, and so on.)
- Points are inserted by selecting the median of the points being put into the subtree, with respect to their coordinates in the axis being used to create the splitting plane.

This method leads to a balanced kd-tree, in which each leaf node is about the same distance from the root. However, balanced trees are not necessarily optimal for all applications.

Note also that it is not required to select the median point. In that case, the result is simply that there is no guarantee that the tree will be balanced. A simple heuristic to avoid coding a complex linear-time median finding algorithm nor using an O(n log n) sort is to use sort to find the median of a fixed number of randomly selected points to serve as the cut line. Practically this technique often results in nicely balanced trees.

3.3.2 Perturbation Tree

Originally developed in [17], a kd-tree is a data structure for partitioning and storing data. A kd-tree recursively divides a data set into smaller subsets such that data points within each subset are more homogeneous after each partition. Kd-trees are primarily used for numeric data. A kd-tree algorithm typically selects the attribute with the largest variance and splits the data into two subsets at the median or midrange of the attribute. Such splitting criteria optimize the efficiency of the partitioning process [17]. After each partition, numeric values (of the attribute selected for partitioning) within a subset are relatively closer to each other. The computational time complexity for kd-trees is of O (N log N) [17].

3.4 Normalization

Objects (e.g. individuals, patterns, events) are usually represented as points (vectors) in a multidimensional space. Each dimension represents a distinct attribute describing an object. Thus, a set of objects is represented as an m x n matrix D, where there are m rows, one for each object, and n columns, one for each attribute. This matrix is referred to as a data matrix, represented as follows:

$$D = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

The attributes in a data matrix are sometimes normalized before being used.

The main reason is that different attributes may be measured on different scales (e.g. centimeters and kilograms). For this reason, it is common to standardize the data so that all attributes are on the same scale. There are many methods for data normalization [28]. We review only two of them in this section: min-max normalization and z-score normalization. Min-max normalization performs a linear transformation on the original data. Each attribute is normalized by scaling its values so that they fall within a small specific range, such as 0.0 and 1.0. Min-max normalization maps a value V of an attribute A to V' as follows:

$V' = \frac{V - \max_A}{\max_A - \min_A} X (new\_\max_A - new\_\min_A) + new\_\max_A$ Where $\max_A$ *and* $\min_A$ represent the minimum and maximum values of an attributes A, respectively, while new $\min_A$ and new $\max_A$ are the new range in which the normalized data will fall.

When the actual minimum and maximum of an attribute are unknown, or when there are outliers that dominate the min-max normalization, z- score normalization (also called zero-mean normalization) should be used.

In z-score normalization, the values for an attribute A are normalized based on the mean and the standard deviation of A. A value V mapped to V' as follows:

$$V' = \frac{V - \overline{A}}{\sigma_A}$$

Where $\overline{A}$ and σ_A are the mean and the standard deviation of the attributes A, respectively.

4. Proposed Method

In this project, we focus on Privacy Preserving Clustering in Data Mining. Notably when personal data is shared for clustering analysis, we should adopt many techniques to limit the disclosure of confidential information from the unauthorized users, so that the private data and private knowledge remain private even after the mining process. To address privacy concerns in clustering analysis, we need to design specific data perturbation methods that enforce privacy without loosing the benefit of mining. We propose a method, called perturbation trees, that uses a recursive partitioning technique to divide a data set into subsets that contain similar data. The partitioned data are perturbed using microaggregation technique. Since the data partitioned based on the joint properties of multiple confidential and non confidential attributes, the relationships between attributes are expected to be reasonably preserved. Further, the proposed method is computationally efficient.

The confidential attributes are perturbed using all of the nonconfidential attributes in the perturbation tree, sometimes some leaf of the perturbation tree may contain more than pre-specified numbers of records. For replace the records of the leaf with the leaf average cause more information loss. To stop the information loss better to use microaggregate technique.

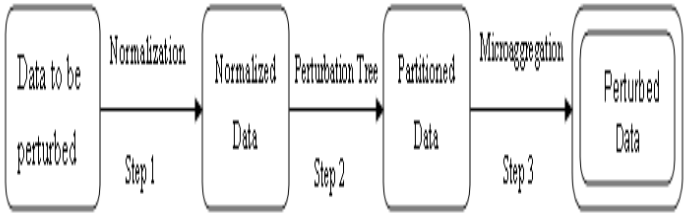

Figure 4.1: Major steps of the data perturbation before clustering analysis

The propose algorithm is given below:

Proposed Perturbation tree algorithm:

1. Let J be number of attributes, including the confidential attributes, in the data. Normalize the data to the unit scale.
2. Let Z be normalized data matrix at the current node. Compute the variance of each dimension, Var(Z_1),…Var(ZJ), based on Z. Let J^* be the dimension with the maximum variance; that is,

$$J^* = \arg\max Var(Z_1), \cdots Var(Z_J)$$

3. Find the median or mid-range of attribute J*, where the mid-range is defined as
[min (ZJ*) + max (ZJ*)] /2. Partition Z into two subsets (Child nodes) based on the median or mid-range. If the mid-range value coincides with the value of the attribute in one or more records, then the mid-range value (i.e. the threshold) can be randomly reduced or increased by epsilon so that the records lie on one partition.
4. Repeat step2 and step3 for each of the two child nodes until attributes are exhausted or the node contains less than a pre specified minimum number of records.
5. For each node that contains more than a pre-specified maximum number of records apply k-means algorithm to further subdivide the group of records.
6. Apply the microaggregation on the records of the node.

4.1 Demonstration of Perturbation Tree

The proposed algorithm, based on the kd-tree technique, demonstrate how it works, consider the synthetic data set in table 4.1, where Age and YearEdu are nonconfidential and income is confidential.

No	Age	YearEdu	Iincome (Original)	Income (Perturbed)
1	25	16	54	57.0
2	31	14	55	52.0
3	32	18	60	57.0
4	36	12	49	52.0
5	43	16	65	61.3
6	48	20	70	71.5
7	50	13	57	61.3
8	53	18	73	71.5
9	56	14	62	61.3

Table 4.1: An illustrative example

The data is plotted in Fig 4.2, where the size and labeled value of a bubble represents an income value. The algorithm first selects Age as the splitting attribute based on maximum variance criterion, and splits the data at the midrange 40.5.

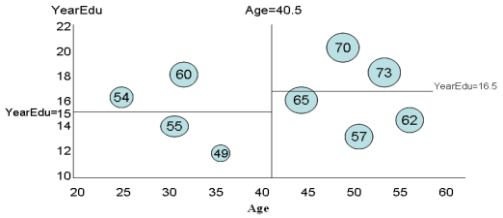

Figure 4.2: Plot of data and partitioning (single attribute)

It continues the splitting process on the partitioned sets until each subset contains no more than three records. The entire process can be illustrated in a tree structure, as shown in Fig. 4.3.

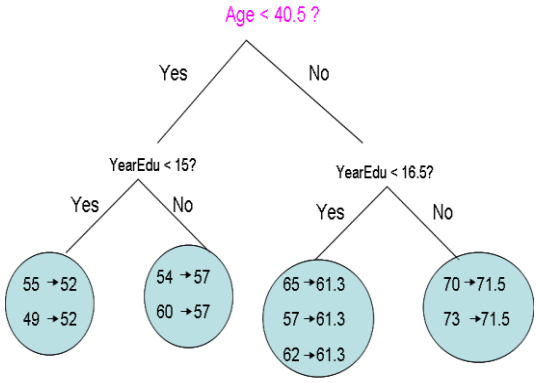

Figure 4.3: The perturbation tree

The confidential values at a leaf are then replaced with the average value at the leaf in the perturbed set. It can be seen that data points within a partitioned region are similar to each other, in both nonconfidential and confidential attributes. The perturbation tree essentially uses an estimate of the conditional expectation of the confidential data in a region as the perturbed value for all records in that region.

For example, the perturbed value for the two cases (#2 and #4) in the lower left region is determined by the estimate of the conditional expectation E(Income|Age < 40:5; YearEdu < 15).The use of conditional expectation effectively preserves the relationships between confidential and nonconfidential attributes.

Since data records are partitioned along all dimensions (attributes) in a perturbation tree, confidential values in a region are generally less close (homogeneous) than those in UMA when there is a single confidential attribute.

As a result, perturbation trees are expected to have lower disclosure risk than UMA, given the same amount of perturbation. The parameter leaf-size in a perturbation tree can be used to control the trade-off between disclosure risk and information loss. That is, the more records a leaf contains, the more severely the data are perturbed, which means a lower disclosure risk and a higher information loss.

The table 4.2 given below shows Education, Experience is nonconfidential attributes and Wages, HRA is confidential attributes. The algorithm first selects Experience as the splitting attribute based on the maximum variance criterion, and splits the data at the midrange of 26.5. It continues the splitting process on the partitioned sets until each subset contains no more than three records.

Education	Experience	Wages(Confidential)	HRA(Confidential)
8	21	5	8
9	42	4	12
14	25	6	14
17	20	4	10
14	27	7	7
13	19	13	11
10	27	4	15
12	17	19	17
16	11	13	19
12	29	8	14

Table 4.2: An synthetic data set

The Fig.4.4 given below shows two confidential attributes are plotted and partitioning in the region.

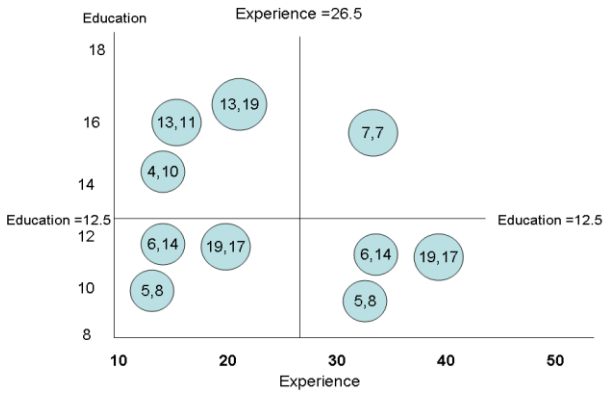

Figure 4.4: Plot of data and partitioning (multiple attributes)

To alleviate the problem of higher information loss, we microaggregate the data within each leaf using multivariate microaggregation.

We now describe some properties for the data perturbed by the perturbation trees.

1. The mean of the perturbed data generated by a perturbation tree is always equal to the mean of the original data.

2. The variance of the perturbed data generated by a perturbation tree is always smaller than or equal to the variance of the original data.

3. The perturbation trees builds from the perturbed records are the same or very similar to the trees built from the original records.

5. Experimental Evaluation

Our goal is to obtain accurate data mining results with the valid amount of data perturbation of a particular data set. We conducted experiments on four real world data sets to evaluate the proposed algorithm. All the experiments were conducted on a PC, Acer Incorporated Intel(R), Pentium(R) 4 CPU 3.00 GHz, 504 MB of RAM running a Linux operating system. All the implementations are done in C++ Language. The effectiveness is measured in terms of the proportion of the points that are grouped in the same clusters after we apply a perturbation on the data.

5.1 Data sets overview

Here all the numerical attributes of the data set is used only.

1. The first data set, called AISOffer, is taken from [18]. It consists of 443 records of MIS faculty salary offers, with 4 attributes, including salary offered, position, course load, number of years teaching, etc.

Salary was considered as the confidential attribute.

2. The second data set, Wages [19], is a population survey containing data with 8 attributes on 534 individuals, representing Wages (confidential), age, gender, education, marital status.

3. The third data set, Wine [26], is a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars with 13 attributes and 178 records. Alcohol was considered as the confidential attribute.

4. The four data set, Housing1 [20], has 506 records with 13 attributes, including housing price (confidential), age, number of rooms, tax, crimerate by town.

To investigate how well the relationships between confidential and nonconfidential attributes are maintained, we performed clustering analyses. We compared the effectiveness of our methods with respect to partition based clustering method. To do so, we selected k-means, the most well-known and commonly used partitioning method [16].

5.2 Performance Analysis of Experimental Results

The table 5.1 shows the perturbed data for confidential attributes (Experience and Wages), for a portion of Wages data set by perturbation tree.

Education	Experience (Original)	Experience (Perturbed)	Wages (Original)	Wages (Perturbed)
8	21	29.57	5.1	4.38
9	42	29.57	4.95	8.36
12	1	14.86	6.67	10.62
12	4	14.86	4	10.62
12	17	14.86	7.5	10.62

13	9	16.8	13.07	10.62
10	27	29.57	4.45	8.38
12	9	16.8	19.47	10.62
16	11	16.8	13.28	10.62
12	9	16.8	8.75	10.62
12	17	14.86	11.35	10.62
12	19	14.86	11.5	10.62
8	27	29.57	6.5	8.38
9	30	29.57	6.25	8.38
9	29	16.8	19.98	8.38
12	37	14.86	7.3	14.75
7	44	29.57	8	8.38
12	26	16.8	22.2	14.75
11	16	29.57	3.65	4.38

Table 5.1: Example Results

The table 5.2 given below shows the results for a portion of Wages data set when the microaggregation is applied to the partitioned data.

Education	Experience (Original)	Experience (Perturbed)	Wages (Original)	Wages (Perturbed)
8	21	18.5	5.1	4.375
9	42	43	4.95	6.03
12	1	2.5	6.67	6.73
12	4	2.5	4	6.73
12	17	22.5	7.5	6.73
13	9	9.67	13.07	12.3
10	27	28	4.45	6.03
12	9	9.67	19.47	19.47
16	11	9.67	13.28	12.3
12	9	9	8.75	6.73
12	17	22.5	11.35	12.3
12	19	22.5	11.5	12.3
8	27	28	6.5	6.03
9	30	28	6.25	6.03
9	29	29	19.98	19.98
12	37	22.5	7.3	14.75
7	44	43	8	6.03
12	26	26	22.2	14.75
11	16	18.5	3.65	4.375

Table 5.2: Experimental Results for Microaggregation

5.3 Measuring Effectiveness

The effectiveness of a perturbation method is measured [4] in terms of the number of legitimate points grouped in the original and the distorted databases. After perturbing the data, the clusters in the original databases should be equal to those ones in the distorted database. However, this is not always the case, and we can have some potential problems after data transformation: either a noise data point end-up clustered, a point from a cluster becomes a noise point, or a point from a cluster migrates to a different cluster. We call this problem Misclassification Error, and it is measured in terms of the percentage of legitimate data points that are not well classified in the distorted database. Ideally, the misclassification error should be 0%.

The misclassification error, denoted by M_E is measured as

$$M_E = \frac{1}{N} X \left(\left| Cluster_i(D) \right| - \left| Cluster_i(D') \right| \right)$$

Where N represents the number of points in the original data set, k is the number of clusters under analysis, and $\left|Cluster_i(X)\right|$ represents the number of legitimate data points of the i^{th} cluster in the database X.

Effectiveness of our method mainly depends on the k-value which is used to group the records before and after final perturbation. With less value of k, points distort less and cluster in the original data set remains same with the perturbed data set, while with large value of k, points distort more and one or more points migrate to a different cluster. We checked the effectiveness of the proposed method in terms of this k and clustering results obtained before and after perturbation. The result of misclassification is shown in Table 5.3.

Method	K=2	K=3	K=4	K=5	K=6
kd-tree (single attribute)	0.00	0.035	0.035	0.035	0.07
kd-tree (multiple attributes)	0.00	0.035	0.105	0.21	0.21
UMA	0.00	0.105	0.105	0.07	0.07
MMA	0.00	0.140	0.105	0.20	0.20

Table 5.3: Results of Misclassification Error of our methods

Here we can point out the results of misclassification as obtained by Oliveira and Zaiane in [4] in four different transformation methods. This is shown in Table 5.4.

Method	K=2	K=3	K=4	K=5	K=6
TDP	0.00	0.00	0.07	0.07	0.07
SDP	0.00	0.03	0.06	0.08	0.08
RDP	0.00	0.15	0.15	0.17	0.13
HDP	0.00	0.08	0.10	0.08	0.08

Table 5.4: Results of Misclassification obtained in four data transformation methods

From above two tables we can conclude that our method is comparable with the randomization only methods in terms of accuracy while achieving better security. These results suggest that our technique perform well for comprising the infeasible goal of having both privacy and accuracy for clustering analysis.

6. Conclusion

This report presents a tree-based data perturbation approach for privacy preserving data mining. We have shown that the method is both efficient and effective, due to the recursive divide and conquer technique adopted with microaggreagation. The method was evaluated in light of several synthetic as well as real life data and the performance has been found satisfactory.

REFERENCES

[1] A. Evfimievski and T. Grandison. Privacy Preserving Data Mining. ACM SIGMOD Record, Vol. 29, No. 2, page 439-450, 2000.

[2] V.S. Verykios, E. Bertino, I.N. Fovino, L.P. Provenza, Y. saygin and Y. Theodoridis. State-of the-art in Privacy Preserving Data Mining. SIGMOD Record, Vol. 33, No. 1, March 2004.

[3] V. Estivill-Castro and L. Brankovic. Data Swapping: Balancing Privacy against Precision in Mining for Logic Rules. In Proc. of Data Warehousing and Knowledge Discovery DaWaK-99, page 389-398, Florence, Italy, August 1999.

[4] S.R.M. Oliveira and O.R. Zaiane. Privacy Preserving Clustering By Data Transformation. In Proc. of the 18th Brazilian Symposium on Databases, pages 304-318, Manaus, Brazil, October 2003.

[5] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In Proc. of the 2000 ACM SIGMOD International Conference on Management of Data, page 439-450, Dallas, Texas, May 2000.

[6] J. Nin, J. Herranz, and V. Torra. Attribute Selection in Multivariate Microaggregation. In Proc. of the 2008 International Workshop on Privacy and Anonymity in Information Society, page 51-60, Nates, France.

[7] A.W. Moore. An Introductory Tutorial on kd-trees. University of Cambridge, 1991.

[8] R.C. Gonzalez and R.E. Woods. Digital Image Processing. Addison- Wesley Publishing Company, 1992.

[9] N.R. Adam and J.C. Worthmann. Security-Control Methods for Statistical Databases: Comparative Study. ACM Computing Surveys, 21(4):515-556, December 1989.

[10] D.E. Denning and J. Schlorer. Inference Controls for Statistical Databases. IEEE Computer, 16(7):69-82, July 1983.

11] S. Castano, M. Fugini, G. Martella, and P. Samarati. Database Security. Addison-Wesley Longman Limited, England 1995.

[12] M.K. Reiter and A.D. Rubin. Crowds: Anonymity for Web Transactions. The ACM Transaction on Information and System Security,1(1):66-92, 1998.

[13] W. Klosgen. Anonymization Techniques for knowledge Discovery in Databases. In Proc. of the First International Conference of Knowledge Discovery and Data Mining(KDD-95), page 186-191,Montreal,Canada<august 1995.

[14] P. Samarati. Protecting Respondents' Identities in Microdata Release. IEEE Transactions of Knowledge and data engineering, 13(6):1010-1027, 2001.

[15] A.K. Pujari. Data Mining Techniques. Universities Press, 2007.

[16] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2007.

[17] S. Sarkar and X.B Liu. A Tree-Based Data Perturbation Approach for Privacy Preserving Data Mining. IEEE Transactions on Knowledge and Data Engineering. Vol. 18, N0. 9, Page 1278-1283, September 2006.

[18] D. Galletta, \MIS Faculty Salary Survey," Mar. 2004, http://www.pitt.edu/ galletta/.

[19] E.R. Berndt, The Practice of Econom trics. New York: Addison-Wesley, 1991.

[20] D. Harrison and D.L. Rubinfeld, \Hedonic Prices and the Demand for Clean Air," J. Environmental Economices and Management, Vol. 5, page 81-102, 1978.

[21] J.F. Traub, Y. Yemini and H. Wozniakowski, \The Statistical Security of a Statistical Database," ACM Trans. Database Systems, vol. 9, no. 4, pp. 672-679, 1984.

[22] S. Greengard. Privacy: Entitlement or Illusion, Personnel Journal, pages 74-88, 1996.

[23] M. Culnan. How Did They Get My Name?. An Exploratory Investigation of Consumer Attitudes towards Secondary Information Use, MIS Quarterly, pages 341-363, 1993.

[24] M. Rotenberg. Protecting Privacy, Communications of the ACM, pages 164, 1992.

[25] Stanford Student Computer and Network Privacy Project. A Study of Student Privacy Issues at Stanford University, Communications of the ACM, page 23-25, 2002.

[26] C. Blake. UCI Repository of Machine Learning databases. Institute of Pharmaceutical and Food Analysis and Technologies, Italy, 1998. (Available online at http://archive.ics.uci.edu/ml/machine-learning- databases/wine/wine.data).

[27] A. Evfimievski, R. Agrawal and R. Srikant. Privacy Preserving Mining of Association Rules. In Proc. 8th ACM SIGKDD International Con- ference on Knowledge Discovery and Data Mining, Edmonton, Canada, pages 217-228, 2002.

[28] V.S Verykio, A.K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni. Association Rule Marketing. IEEE Transactions on Knowledge and Data Engineering, page 434-447, 2004.

[29] P. Jefferies, Multimedia, Cyberspace & Ethics. In Proc. of International Conference on Information Visualization (IV2000), page 99-104, London, England, July 2000.

[30] S.J. Rizvi and J.R. Haritsa. Maintaining Data Privacy in Association Rule Mining. In Proc. of the 28th International Conference on Very Large Databases (VLDB'02), Hong Kong, China, 2002.