Privacy Preserving Clustering By Data Transformation

Stanley R. M. Oliveira[1,2]
[1]Embrapa Informática Agropecuária
André Tosello, 209 - Barão Geraldo
13083-886 - Campinas, SP, Brasil
oliveira@cs.ualberta.ca

Osmar R. Zaïane[2]
[2]Department of Computing Science
University of Alberta
Edmonton, AB, Canada, T6G 1K7
zaiane@cs.ualberta.ca

Abstract

Despite its benefit in a wide range of applications, data mining techniques also have raised a number of ethical issues. Some such issues include those of privacy, data security, intellectual property rights, and many others. In this paper, we address the privacy problem against unauthorized secondary use of information. To do so, we introduce a family of geometric data transformation methods (GDTMs) which ensure that the mining process will not violate privacy up to a certain degree of security. We focus primarily on privacy preserving data clustering, notably on partition-based and hierarchical methods. Our proposed methods distort only confidential numerical attributes to meet privacy requirements, while preserving general features for clustering analysis. Our experiments demonstrate that our methods are effective and provide acceptable values in practice for balancing privacy and accuracy. We report the main results of our performance evaluation and discuss some open research issues.

1. Introduction

Huge volumes of detailed personal data are regularly collected and analyzed by applications using data mining. Such data include shopping habits, criminal records, medical history, credit records, among others [4]. On the one hand, such data is an important asset to business organizations and governments both to decision making processes and to provide social benefits, such as medical research, crime reduction, national security, etc. [12]. On the other hand, analyzing such data opens new threats to privacy and autonomy of the individual if not done properly.

The threat to privacy becomes real since data mining techniques are able to derive highly sensitive knowledge from unclassified data that is not even known to database holders. Worse is the privacy invasion occasioned by secondary usage of data when individuals are unaware of "behind the scenes" use of data mining techniques [13]. As an example in point, Culnan [6] made a particular study of secondary information use which she defined as "the use of personal information for other purposes subsequent to the original transaction between an individual and an organization when the information was collected." The key finding of this study was that concern over secondary use was correlated with the level of control the individual has over the secondary use. As a result, individuals are increasingly feeling that they are losing control over their own personal information that may reside on thousands of file servers largely beyond the control of existing privacy laws. This scenario has led to privacy invasion on a scale never before possible.

The challenging problem that we address in this paper is: how can we protect against the abuse of the knowledge discovered from secondary usage of data and meet the needs of organizations and governments to support decision making or even to promote social benefits? We claim that a solution for such a problem requires two vital techniques: *anonymity* [15, 17] to remove identifiers (e.g. names, social insurance numbers, addresses, etc.) in the first phase of privacy protection, and *data transformation* to protect some sensitive attributes (e.g. salary, age, etc.) since the released data, after removing identifiers, may contain other information that can be linked with other datasets to re-identify individuals or entities [19]. 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 protection in clustering analysis, notably on partition-based and hierarchical methods.

The intuition behind such methods is to partition a dataset into new classes (clusters) of similar objects. The goal is to group objects to achieve high similarity between objects within individual clusters (interclass similarity) and low similarity between objects that belong to different clusters (intraclass similarity) [11]. Clustering is widely used in many applications such as customer behaviour analysis, targeted marketing, and many others.

A motivating example for the privacy problem in data clustering could be found in business collaboration. Two or more companies have a very large dataset of records of their customers' buying activities. These companies decide to cooperatively conduct data clustering on their datasets for their mutual benefit since this collaboration brings them an advantage over other competitors. The goal is to subdivide a market into distinct subsets of customers where any subset may be selected as a market to be reached with a distinct marketing mix. However, these companies would like to transform their data in such a way that the privacy of their customers cannot be violated. Is it possible for these companies to benefit from such collaboration by sharing their data while preserving the private information of their customers?

To address privacy concerns in clustering analysis, we need to design specific data transformation methods that enforce privacy without loosing the benefit of mining. The proposed data perturbation methods in the literature pertain to the context of statistical databases [1, 7, 5, 16]. They do not apply to 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 and age) are represented by points in a 2D 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.

In this paper, we introduce a family of geometric data transformation methods (GDTMs) that distort confidential numerical attributes in order to meet privacy protection in clustering analysis. We benefit from the work on image processing [10]. Of particular interest is work on geometric transformation of digital images, notably the idea behind translation, scaling, and rotation. We also benefit from the work on statistical databases, particularly the intuition behind data distortion. We show that our transformation data methods are simple, independent of clustering algorithms, preserve the general features of the clusters, and have a sound mathematical foundation. Although our approach does not provide a comprehensive solution to the problem of privacy preservation in data mining, we argue that our approach is a simple building block toward privacy preserving data clustering. To date, such schemata have not been explored in detail.

This paper is organized as follows. Related work is reviewed in Section 2. In Section 3, we provide the basic concepts that are necessary to understand the scope and the issues addressed in this paper. We introduce our family of geometric data transformation methods in Section 4. In Section 5, we present the experimental results and discussion. Finally, Section 6 presents our conclusions and a discussion of future work.

2. Related Work

Some effort has been made to address the problem of privacy preservation in data mining. This effort has been restricted basically to classification and association rules. The class of solutions for this problem rely on data partition, data sanitization, randomization and data distortion. In this work, we focus on the last two categories.

Estivill-Castro and Brankovic [8] introduced a method for ensuring partial disclosure while allowing a miner to explore detailed data. In this approach, one first builds a local decision tree over true data, and then swaps values amongst records in a leaf node of the tree to generate randomized training data. The swapping is performed over the confidential attribute only, where the confidential attribute is the class label. This approach deals with a trade-off: statistical precision against security level, i.e., the closer to the root, the higher the security but lower the precision.

Agrawal and Srikant [3] considered the case of building a decision-tree classifier from training data in which the values of individual records have been perturbed, by adding random values from a probability distribution. The resulting data records look very different from the original records and the distribution of data values is also very different from the original distribution. While it is not possible to accurately estimate original values in individual data records, they proposed a novel reconstruction procedure to accurately estimate the distribution of original data values. The distribution reconstruction process naturally leads to some loss of information, but the authors argue that this is acceptable in many practical situations.

In [2], the authors proposed a new algorithm for distribution reconstruction which is more effective than that proposed in [3], in terms of the level of information loss. This algorithm, based on Expectation Maximization (EM) algorithm, converges to the maximum likelihood estimate of the original distribution based on the perturbed data, even when a large amount of data is available. They also pointed out that the EM algorithm was in fact identical to the Bayesian reconstruction proposed in [3], except for the approximation partitioning values into intervals.

Evfimievski et al. [9] proposed a framework for mining association rules from transactions consisting of categorical items in which the data has been randomized to preserve privacy of individual transactions. The idea behind this approach is that some items in each transaction are replaced by new items not originally present in this transaction. In doing so, some true information is taken away and some false information is introduced, which seems to have obtained a reasonable privacy protection. In general, this strategy is feasible to recover association rules, less frequent than originally, and preserve privacy using a straightforward uniform randomization. Although privacy is preserved on average, confidential information leaks through uniform randomization for some fraction of transactions.

More recently, the data distortion approach has been applied to boolean association rules [18]. Again, the idea is to modify data values such that reconstruction of the values for any individual transaction is difficult, but the rules learned on the distorted data are still valid. One interesting feature of this work is a flexibility definition of privacy. For instance, the ability to correctly guess a value of ‘1’ from the distorted data can be considered a greater threat to privacy than correctly learning a ‘0’. This scheme is based on probabilistic distortion of user data, which is composed of a privacy metric and an analytical formula. Although this framework provides a high degree of privacy to the user and retains a high level of accuracy in the mining results, mining the distorted database can be, apart from being error-prone, significantly more expensive in terms of both time and space as compared to mining the original database.

The work presented here differs from the related work in some aspects, as follows: First, we aim to address the problem of privacy preservation in clustering analysis. To our best knowledge, this problem has not been considered so far. Our proposed solution and those ones in the related work are complementary. Second, we study the impact of our data transformation schemes in the original database by quantifying how much information is preserved after transforming a database. So, our focus is not only on protecting individual data records, but also on providing accurate data for clustering analysis.

3. Basic Concepts

In this section, we briefly review the basic concepts that are necessary to understand the issues addressed in this paper. We start by giving the main idea behind data perturbation, followed by the basics of geometric transformation of digital images.

3.1. The Basics of 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 [1, 5]. In the probability-distribution category, the security-control method replaces the original database 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 have 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. In all cases, such methods involve the addition of noise term with the mean 0, and hence result in no bias in estimating the mean. In this paper, we focus on fixed-data perturbation methods.

In its simplest form, fixed-data perturbation methods involve perturbing a confidential attribute X by adding some noise term e to result in the perturbed attribute Y. When this method is used for multi-attribute databases, each attribute in the database is perturbed independently of the others. In general, this method is described as $Y = X + e$, where e is drawn from some probability distribution (e.g. Uniform, Normal) with mean 0 and a known variance to the data [1]. These methods are referred to as Additive Data Perturbation (ADP). Apart from ADP methods, Multiplicative Data Perturbation (MDP) can also be used to provide aggregate statistics, while protecting the privacy of individuals represented in a database. In such a method, for a single confidential attribute X, the perturbed attribute Y is described as $Y = Xe$, where e has a mean of 1.0 and a specified variance [16]. Since the mean of $e =$ 1.0, there is no bias in estimating the mean. When the MDP method is used to distort multiple confidential attributes, each attribute must be perturbed independently of other attributes.

Fixing the perturbation of an attribute, using either ADP or MDP methods, prevents users from improving the estimates of the value of a field in a record by repeating queries. For this reason these methods are suitable for released databases [5, 16].

3.2. The Basics of Imaging Geometry

For the sake of simplicity, we provide the basics of imaging geometry in a 2D discrete space. However, the foundations are scalable to other dimensions. A digital image $a[m, n]$ described in a 2D discrete space is derived from an analog image $a(x, y)$ in a 2D continuous space through a sampling process that is frequently referred to as digitization. The 2D continuous image $a(x, y)$ is divided into N rows and M columns. The intersection of a row and a column is termed a pixel. The value assigned to the integer coordinates $[m, n]$ with $m = 0, 1, 2, ..., M - 1$ and $n = 0, 1, 2, ..., N - 1$ is $a[m, n]$ [10].

There are some transformations that can be applied to digital images to transform an input image $a[m, n]$ into an output image $b[m, n]$. In this work, we consider the transformations translation, scaling, and rotation. We are expressing such transformations in a two-dimensional Cartesian coordinate system, in which a point has coordinates denoted (X, Y). The same transformations can be extrapolated to high dimensional data spaces.

Translation is the task to move a point with coordinates (X, Y) to a new location by using displacements (X_0, Y_0). The translation is easily accomplished by using a matrix representation $v' = Tv$, where T is a 2 × 3 transformation matrix depicted in Figure 1A, v is the vector column containing the original coordinates, and v' is a column vector whose coordinates are the transformed coordinates. This matrix form is also applied to Scaling and Rotation.

Scaling by factors S_x and S_y along the X and Y axes is given by the transformation matrix seen in Figure 1B.

Rotation is a more challenging transformation. In its simplest form, this transformation is for the rotation of a point about the coordinate axes. Rotation of a point in a 2D discrete space by an angle θ is achieved by using the transformation matrix depicted in Figure 1C. The rotation angle θ is measured clockwise and this transformation affects the values of X and Y coordinates.

$$\begin{bmatrix} 1 & 0 & X_0 \\ 0 & 1 & Y_0 \end{bmatrix} \qquad \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \qquad \begin{bmatrix} cos\,\theta & sin\,\theta \\ -sin\,\theta & cos\,\theta \end{bmatrix}$$

(A) (B) (C)

Figure 1: (A) Transformation matrix for Translation; (B) Transformation matrix for Scaling; (C) Transformation matrix for Rotation.

4. The Family of Geometric Data Transformation Methods

In this section, we introduce the family of geometric data transformation methods (GDTM) that we propose to meet privacy preservation in clustering analysis.

4.1. Basic Definitions

For this paper, the data is assumed to be a matrix D_{mn}, where each of the m rows is an observation, O_i, and each observation contains values for each of the n attributes, A_i. The matrix D_{mn} may contain categorical and numerical attributes. However, our GDTMs rely on d numerical attributes, such that $d \leq n$. Thus, the $m \times d$ matrix, which is subject to transformation, can be thought of as a vector subspace V in the Euclidean space such that each vector $v_i \in V$ is the form $v_i = (a_1, ..., a_d)$, $1 \leq i \leq d$, where $\forall i$ a_i is one instance of A_i, $a_i \in \Re$, and $\Re$ is the set of real numbers.

The vector subspace V must be transformed before releasing the data for clustering analysis in order to preserve privacy of individual data records. To transform V into a distorted vector subspace V', we need to add or even multiply a constant noise term e to each element v_i of V. To do so, we define a *uniform noise vector* as follows:

Definition 1 (Uniform Noise Vector) *Let $N = (\langle o1 : OP_1, e_1 : NT_1\rangle, ..., \langle o_d : OP_d, e_d : NT_d\rangle)$ be a uniform noise vector, and for $1 \leq i \leq d$, let $D_i(OP)$ be the set of operations associated with the domain of OP_i, and let $D_i(E)$ be the set of noisy term associated with the domain of NT_i. An instance of N that satisfies the domain constraints is a vector of the form: $\{[\langle o_1 : op_1, e_1 : nt_1\rangle, ..., \langle o_d : op_d, e_d : nt_d\rangle] \mid \forall i \; op_i \in D_i(OP), nt_i \in D_i(E)\}$.*

The set of operations $D_i(OP)$ takes the values {Mult, Add, Rotate}, where *Mult* and *Add* correspond to a multiplicative and additive noise applied to one confidential attribute respectively. *Rotate,* denoted by $A_i \circlearrowright A_j$, implies that all instances of the attributes A_i and A_j are rotated by a common angle. In the next sections, we exemplify the use of the uniform noise vector N.

Given the uniform noise vector N, we can transform the vector subspace V into the vector subspace V' by using a geometric transformation function.

Definition 2 (Geometric Transformation Function) *Let V be a d-dimensional vector subspace, where each element v_i, $1 \leq i \leq d$, is the form $v_i = (a_1, ..., a_d)$, and each a_i in v_i is one observation of a confidential numerical attribute, and let N = $(\langle op_1, e_1\rangle, ..., \langle op_d, e_d\rangle)$ be a uniform noise vector. We define a geometric transformation function f as a bijection of d-dimensional space into itself which transforms V into V' by distorting all attributes of v_i in V according to its corresponding i-th element in N. Each vector v' of V' is the form $v' = (\langle a_1 \; [op_1] \; e_1\rangle, ..., \langle a_d \; [op_d] \; e_d\rangle)$, and $\forall i$, $\langle a_i \; [op_i] \; e_i\rangle \in \Re$.*

In this paper, we consider the following geometric transformation functions: *Translation*, *Scaling*, and *Rotation* whose corresponding operations are *Add*, *Mult*, and *Rotate*. Based on the previous definitions, we can define a geometric transformation method (GDTM) as follows:

Definition 3 (Geometric Data Transformation Method) *A geometric data transformation method of dimension d is a ordered pair, defined as $GDTM = (V, f)$ where:*

- *$V \subseteq \Re^d$ is a representative vector subspace of data points to be transformed;*
- *f is a geometric transformation function, $f : \Re^d \rightarrow \Re^d$.*

For our GDTMs, the inputs are the vectors of V, composed of confidential numerical attributes only, and the uniform noise vector N, while the output is the transformed vector subspace V'. Our GDTM algorithms require only one scan, in most cases. All transformation data algorithms have essentially two major steps: (1) Identify the noise term and the operation that must be applied to each confidential attribute. This step refers to the instantiation of the uniform noise vector N; (2) Based on the uniform noise vector N, defined in the previous step, transform V into V' using a geometric transformation function.

4.2. The Translation Data Perturbation Method

In the Translation Data Perturbation Method, denoted by TDP, the observations of confidential attributes in each $v_i \in V$ are perturbed using an additive noise perturbation. The noise term applied to each confidential attribute is constant and can be either positive or negative. The set of operations $D_i(OP)$ takes only the value {Add} corresponding to a additive noise applied to each confidential attribute. The sketch of the TDP algorithm is given as follows:

```
**TDP_Algorithm**
**Input:** $V$, $N$
**Output:** $V'$
Step 1. **For** each confidential attribute $A_j$ in $V$, where $1 \leq j \leq d$ **do**
            1. Select the noise term $e_j$ in $N$ for the confidential attribute $A_j$
            2. The $j$-th operation $op_j \leftarrow \{\text{Add}\}$
Step 2. **For** each $v_i \in V$ **do**
            **For** each $a_j$ in $v_i = (a_1, ..., a_d)$, where $a_j$ is the observation of the $j$-th attribute **do**
                1. $a'_j \leftarrow$ Transform($a_j,\ op_j,\ e_j$)
**End**
```

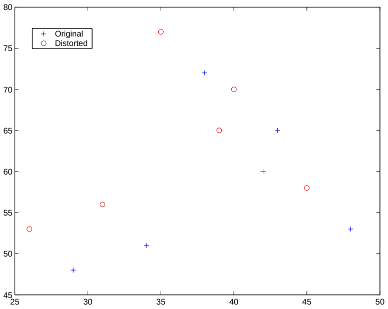
To illustrate how the TDP method works, let us consider the sample relational database in Figure 2A. In this example, the column $O\#$ represents observations. Note that we have removed the identifiers. Suppose we are interested in grouping individuals based on the attributes *Age* and *Salary*, but the attributes are confidential. To do so, we apply our TDP method. The uniform noise vector for this example is $N = (\langle Add, -3\rangle, \langle Add, 5000\rangle)$. Figure 2B shows the distorted database, and the points before and after distortion can be seen in Figure 2C.

O#	Occupation	City	Age	Salary
1	Student	Edmonton	29	48,000
2	Executive	Calgary	38	72,000
3	Professor	Edmonton	34	51,000
4	Lawyer	Vancouver	43	65,000
5	Dentist	Victoria	42	60,000
6	Nurse	Toronto	48	53,000

(A)

O#	Occupation	Age	Salary
1	Student	26	53,000
2	Executive	35	77,000
3	Professor	31	56,000
4	Lawyer	40	70,000
5	Dentist	39	65,000
6	Nurse	45	58,000

(B)



(C)

Figure 2: (A): A sample relational database; (B): A translation data perturbation corresponding to the original sample; (C): The representation of the points before "+" and after "o" the perturbation.

4.3. The Scaling Data Perturbation Method

In the Scaling Data Perturbation Method, denoted by SDP, the observations of confidential attributes in each $v_i \in V$ are perturbed using a multiplicative noise perturbation. The noise term applied to each confidential attribute is constant and can be either positive or negative. The set of operations $D_i(OP)$ takes only the value $\{\text{Mult}\}$ corresponding to a multiplicative noise applied to each confidential attribute. The sketch of the SDP algorithm is given as follows:

```
**SDP_Algorithm**
**Input:** $V$, $N$
**Output:** $V'$
Step 1. **For** each confidential attribute $A_j$ in $V$, where $1 \leq j \leq d$ **do**
            1. Select the noise term $e_j$ in $N$ for the confidential attribute $A_j$
            2. The $j$-th operation $op_j \leftarrow \{\text{Mult}\}$
Step 2. **For** each $v_i \in V$ **do**
            **For** each $a_j$ in $v_i = (a_1, ..., a_d)$, where $a_j$ is the observation of the $j$-th attribute **do**
                1. $a'_j \leftarrow$ Transform($a_j,\ op_j,\ e_j$)
**End**
```

To illustrate how the SDP method works, let us consider the sample relational database in Figure

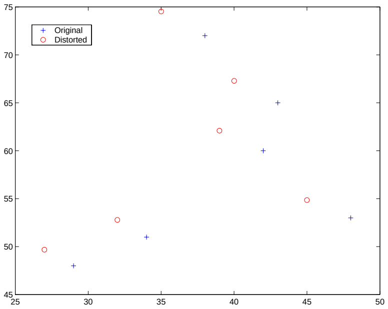
3A. Note that this sample database is identical to the one presented in Figure 2A, but it is repeated for clarity. In this example, we are interested in grouping individuals based on the attributes *Age* and *Salary*. The uniform noise vector for this example is $N = (\langle Mult, 0.94\rangle, \langle Mult, 1.035\rangle)$. Figure 3B shows the distorted database, and the points before and after distortion can be seen in Figure 3C. Note that the values of the attribute age are rounded to be consistent with the values in the real world.

O#	Occupation	City	Age	Salary
1	Student	Edmonton	29	48,000
2	Executive	Calgary	38	72,000
3	Professor	Edmonton	34	51,000
4	Lawyer	Vancouver	43	65,000
5	Dentist	Victoria	42	60,000
6	Nurse	Toronto	48	53,000

(A)

O#	Occupation	Age	Salary
1	Student	27	49,680
2	Executive	35	74,520
3	Professor	32	52.785
4	Lawyer	40	67,275
5	Dentist	39	62,100
6	Nurse	45	54,855

(B)



(C)

Figure 3: (A): A sample relational database; (B): A scaling data perturbation corresponding to the original sample; (C): The representation of the points before "+" and after "o" the perturbation.

4.4. The Rotation Data Perturbation Method

The Rotation Data Perturbation Method, denoted by RDP, works differently from our previous methods. In this case, the noise term is an angle θ. The rotation angle θ, measured clockwise, is the transformation applied to the observations of the confidential attributes. The set of operations $D_i(OP)$ takes only the value $\{\text{Rotate}\}$ that identifies a common rotation angle between the attributes A_i and A_j. Unlike the previous methods, RDP may be applied more than once to some confidential attributes. For instance, when a rotation transformation is applied this affects the values of two coordinates. In a 2D discrete space, the X and Y coordinates are affected. In a 3D discrete space or higher, two variables are affected and the others remain without any alteration. This requires that one or more rotation transformations are applied to guarantee that all the confidential attributes are distorted in order to preserve privacy. The sketch of the RDP algorithm is given as follows:

```
**RDP_Algorithm**
**Input:** $V$, $N$
**Output:** $V'$
Step 1. **For** every two attributes $A_j, A_k$ in $V$, where $1 \leq j \leq d$ and $1 \leq k \leq d$ **do**
            1. Select an angle $\theta$ for the confidential attributes $A_j, A_k$
            2. The $j$-th operation $op_j \leftarrow \{\text{Rotate}\}$
            3. The $k$-th operation $op_k \leftarrow \{\text{Rotate}\}$
Step 2. **For** each $v_i \in V$ **do**
            **For** each $a_l$ in $v_i = (a_1, ..., a_d)$, where $a_l$ is the observation of the $l$-th attribute **do**
                1. $a'_l \leftarrow$ Transform($a_l$, $op_l$, $e_l$)
**End**
```

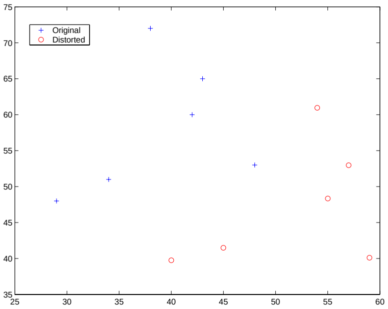
For the sake of simplicity, we illustrate how the RDP method works in a 2D discrete space. Let us consider the sample relational database in Figure 4A Iidem to Figure 2A and Figure 3A). In this example, we are interested in grouping individuals based on the attributes *Age* and *Salary*. The uniform noise vector for this example is $N = (\langle Age \circlearrowright Sal, 13.7\rangle)$. Figure 4B shows the distorted database, and the points before and after distortion can be seen in Figure 4C. Note that the values of the attribute age are rounded to be consistent with the values in the real world.

O#	Occupation	City	Age	Salary
1	Student	Edmonton	29	48,000
2	Executive	Calgary	38	72,000
3	Professor	Edmonton	34	51,000
4	Lawyer	Vancouver	43	65,000
5	Dentist	Victoria	42	60,000
6	Nurse	Toronto	48	53,000

(A)

O#	Occupation	Age	Salary
1	Student	40	39,766
2	Executive	54	60,951
3	Professor	45	41,496
4	Lawyer	57	52,966
5	Dentist	55	48,354
6	Nurse	59	40,123

(B)



(C)

Figure 4: (A): A sample relational database; (B): A rotation data perturbation corresponding to the original sample; (C): The representation of the points before "+" and after "o" the perturbation.

4.5. The Hybrid Data Perturbation Method

The Hybrid Data Perturbation Method, denoted by HDP, combines the strength of our previous methods: TDP, SDP and RDP. In this scheme, we select randomly one operation for each confidential attribute that can take the values {Add, Mult, Rotate} in the set of operations $D_i(OP)$. Thus, each confidential attribute is perturbed using either an additive, a multiplicative noise term, or a rotation. The sketch of the HDP algorithm is given as follows:

```
HDP_Algorithm
Input: $V$, $N$
Output: $V'$
Step 1. For each confidential attribute $A_j$ in $V$, where $1 \leq j \leq d$ do
            1. Select the noise term $e_j$ in $N$ for the confidential attribute $A_j$
            2. The $j$-th operation $op_j \leftarrow$ {Add,Mult,Rotation}
Step 2. For each $v_i \in V$ do
            For each $a_j$ in $v_i = (a_1, ..., a_d)$, where $a_j$ is the observation of the $j$-th attribute do
                1. $a'_j \leftarrow$ Transform($a_j,\ op_j,\ e_j$)
End
```

Let us consider the sample relational database in Figure 5A to illustrate how the HDP method works. In this example, we are interested in grouping individuals based on the attributes *Age* and *Salary*. The uniform noise vector for this example is $N = (\langle Add, 2\rangle, \langle Mult, 0.93\rangle)$. Rotation is not used in this example. Figure 5B shows the distorted database, and the points before and after distortion can be seen in Figure 5C.

5. Experimental Results

In this section, we present the results of our performance evaluation. We start by describing the methodology that we used. Then we study the effectiveness of our GDTMs under partition-based and hierarchical methods followed by an analysis of the privacy level.

5.1. Methodology

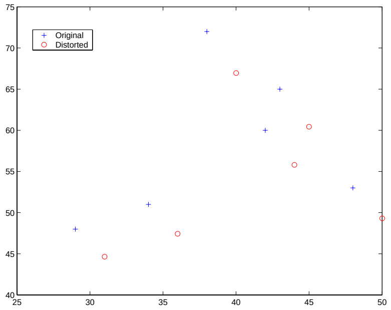
We compared our GDTMs against each other and with respect to the following benchmarks: (1) the result of clustering analysis without transformation; (2) the results of Additive Data Perturbation Method, ADP, that has been widely used for inference control in statistical databases [7, 5, 16].

O#	Occupation	City	Age	Salary
1	Student	Edmonton	29	48,000
2	Executive	Calgary	38	72,000
3	Professor	Edmonton	34	51,000
4	Lawyer	Vancouver	43	65,000
5	Dentist	Victoria	42	60,000
6	Nurse	Toronto	48	53,000

(A)

O#	Occupation	Age	Salary
1	Student	31	44,640
2	Executive	40	66,960
3	Professor	36	47,430
4	Lawyer	45	60,450
5	Dentist	44	55,800
6	Nurse	50	49,290

(B)



(C)

Figure 5: (A): A sample relational database; (B): A hybrid data perturbation corresponding to the original sample; (C): The representation of the points before “+” and after “o” the perturbation.

To measure the effectiveness of our methods, we performed two series of experiments. In the first series, 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 [11]. The second series of experiments focused on a hierarchical clustering method. For this case, we used the Chameleon algorithm that explores dynamic modeling in hierarchical clustering [14].

All the experiments were conducted on a PC, AMD Athlon 1900/1600 (SPEC CFP2000 588), with 1.2 GB of RAM running a Linux operating system. We used five different synthetic datasets, each with 6000 points in a 2D discrete space. For each dataset, we analyzed a specific number of clusters ranging from 2 to 6 clusters. The effectiveness is measured in terms of the proportion of the points that are grouped in the same clusters after we apply a transformation on the data. We refer to such points as legitimate ones.

For the sake of simplicity, we considered the transformation of two confidential attributes: *Age* and *Salary*. The noise term e for the ADP scheme has a Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 100$. The uniform noisy vector for TDP, SDP, HDP, and RDP are $N(TDP) = (\langle Add, -3\rangle, \langle Add, 6, 235\rangle), N(SDP) = (\langle Mult, 0.93\rangle, \langle Mult, 0.89\rangle), N(RDP) = (\langle Age \circlearrowright Sal, 356.71\rangle)$ and $N(HDP)$ combines the three previous ones.

5.2. Measuring Effectiveness

The effectiveness is measured in terms of the number of legitimate points grouped in the original and the distorted databases. After transforming 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 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. Since the clustering methods we used, K-Means and Chameleon, do not consider noise points, we concentrate only on the third case. 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 follows:

$$M_E = \frac{1}{N} \times \sum_{i=1}^{k} (|Cluster_i(D)| - |Cluster_i(D')|)$$

where N represents the number of points in the original dataset, k is the number of clusters under analysis, and $|Cluster_i(X)|$ represents the number of legitimate data points of the ith cluster in the database X.

We should point out that our formula for misclassification error does not simply consider the number of data points in each cluster. Rather, we take into account the actual cluster of each point. We compare the cluster label of each point before and after distortion.

Method	K-Means					Chameleon				
	K = 2	K = 3	K = 4	K = 5	K = 6	K = 2	K = 3	K = 4	K = 5	K = 6
TDP	0.00	0.00	0.07	0.07	0.07	0.00	0.00	0.00	0.00	0.00
SDP	0.00	0.03	0.06	0.08	0.08	0.00	0.03	0.03	0.00	0.00
RDP	0.02	0.15	0.15	0.17	0.13	0.03	0.10	0.10	0.03	0.03
HDP	0.02	0.08	0.10	0.08	0.08	0.00	0.00	0.00	0.00	0.00
ADP	12.02	27.09	31.45	34.18	39.75	17.33	24.93	36.04	37.92	40.76

Table 1: Results of misclassification for K-Means and Chameleon

In Table 1, we have the results of misclassification for K-Means and Chameleon. We compared our GDTMs with respect to the original dataset. We also compared our techniques against the ADP method [1]. To accomplish that, we ran K-means and Chameleon 20 times and collected the average. Our transformation methods resulted in between 0% and less than 0.2% while ADP reached between 12% and 40% of misclassification with K-Means and between 17% and 41% with Chameleon. This clearly shows that ADP is inadequate for transforming data before clustering since this method jeopardizes the notion of similarity between data points. As a result, the data points are literally shuffled leading to significant misclassification. Thus, hereafter we will present the results of our GDTMs only.

As can be seen in Table 1, our techniques TDP, SDP, RDP and HDP yielded very good results when we compare the cluster analysis of the original and the distorted datasets. In the worst case, only 0.17% of the points are misclassified. In general, TDP and SDP yielded the best values, in terms of accuracy, in all different datasets when running K-Means. However, HDP and RDP presented very good results as well. It is good to point out that these values represent the average of 20 trials. If we considered the statistical mode (i.e. the value occurring most frequently in a series of the 20 observations), all our schemes would yield 0%. These small differences are due to the fact that depending on the distribution of the points, K-means is not completely deterministic. On the other hand, the results obtained from Chameleon in a series of 20 observations were basically the same. Note that in the worst case, only 0.10% of the points are misclassified for RDP and 0.03% for SDP, while TDP and HDP yielded the best values in all the cases. These results suggest that our techniques perform well for comprising the infeasible goal of having both complete privacy and complete accuracy for clustering analysis.

5.3. Quantifying Privacy

While perturbation methods guarantee that complete disclosure will not occur, they may be susceptible to partial disclosure [1]. However, fixed-data perturbation methods minimize this problem since such methods prevent users from improving estimates of a particular attribute by repeating queries. It is therefore necessary to measure the level of security provided by a specific perturbation technique when quantifying privacy by such a method.

Traditionally, the privacy provided by a perturbation technique has been measured as the variance between the actual and the perturbed values [1, 16]. This measure is given by $Var(X - Y)$ where X represents a single original attribute and Y the distorted attribute. This measure can be made scale invariant with respect to the variance of X by expressing security as $Sec = Var(X - Y)/Var(X)$.

Clearly, the above measure to quantify privacy is based on how closely the original values of a modified attribute can be estimated. Table 2 shows the privacy provided by our GDTMs, where for each ordered pair [α_1, α_2], α_1 represents the privacy level for the attribute age, and α_2 represents the privacy level for the attribute salary. These values are expressed in percentage.

Method	Privacy Level (%)				
	K = 2	K = 3	K = 4	K = 5	K = 6
TDP	[0.00; 0.00]	[0.00; 0.00]	[0.00; 0.00]	[0.00; 0.00]	[0.00; 0.00]
SDP	[0.49; 1.21]	[0.49; 1.21]	[0.49; 1.21]	[0.49; 1.21]	[0.49; 1.21]
RDP	[0.84; 0.12]	[0.69; 0.79]	[0.83; 0.13]	[0.78; 0.13]	[0.51; 0.21]
HDP	[0.00; 0.64]	[0.00; 0.64]	[0.00; 0.64]	[0.00; 0.64]	[0.00; 0.64]

Table 2: Results of privacy provided by the GDTMs

Based on the results showed in Table 2, one may claim that our GDTMs could be restrictive in terms of privacy. Indeed, TDP may be sometimes restrictive since the variance of a single attribute always yields 0% of privacy level, even though the individual data records look very different from the original ones. In addition, the results provided by SDP, HDP, and RDP are slightly better than those ones provided by TDP. Apart from the problem of low privacy, a geometric transformation function is invertible so that one may estimate the real values of the data under clustering. To cope with these limitations, we introduce one special procedure to improve the privacy level of our GDTMs in the next section.

5.4. Improving Privacy

The procedure to improve the privacy level of our GDTMs is applied to the transformed database only. This procedure is composed of three steps as follows: *Step 1*: We select a probability distribution (e.g. Normal, Uniform) for each confidential numerical attribute A'_i in V', where $1 \leq i \leq d$. *Step 2*: We randomly select ρ% of the vectors $v'_i \in V'$ to reinforce privacy by adding some noise term to each observation of v'_i according to the corresponding probability distribution selected in the previous step. We refer to the parameter ρ as privacy enhance. *Step 3*: Based on the previous steps, we distort the selected vectors v'_i by using the idea behind the Additive Data Perturbation Method (ADP).

To illustrate how the procedure to improve privacy works, we set the privacy enhance ρ = 5%. The distribution selected for the attribute *Age* was Uniform with parameters [-12, 18] and the distribution selected for the attribute *Salary* was Normal with mean μ = 15,000 and variance σ^2 = 144,000. This example yielded the results of misclassification showed in Table 3.

As can be seen in Table 3, the misclassification error was slightly affected when compared with Table 1. However the privacy level of our GDTMs, presented in Table 4, was improved as expected. These figures clearly show that privacy preserving data mining deals with a trade-off: privacy and accuracy, which are typically contradictory, and improving one usually incurs a cost in the other.

The results of privacy and accuracy can vary depending on the parameter ρ. For example, setting ρ to 10% and keeping the probability distribution of the attributes the same, we slightly decreased the

Method	K-Means					Chameleon				
	K = 2	K = 3	K = 4	K = 5	K = 6	K = 2	K = 3	K = 4	K = 5	K = 6
TDP	1.05	1.18	1.67	1.65	2.13	1.15	1.12	1.43	1.33	2.10
SDP	1.28	1.18	1.87	1.45	2.42	1.27	1.20	1.83	1.50	2.25
RDP	1.12	1.23	1.65	1.43	2.08	1.17	1.18	1.68	1.33	2.10
HDP	1.17	1.17	1.52	1.48	2.25	1.20	1.15	1.48	1.43	2.12

Table 3: Results of misclassification for K-Means and Chameleon with privacy enhance ρ = 5%

Method	Privacy Level (%)				
	K = 2	K = 3	K = 4	K = 5	K = 6
TDP	[3.25; 6.80]	[2.25; 4.71]	[2.13; 4.38]	[1.55; 3.40]	[1.29; 4.43]
SDP	[3.72; 8.04]	[2.73; 5.95]	[2.60; 5.60]	[2.05; 4.59]	[1.81; 5.64]
RDP	[4.08; 6.93]	[2.96; 5.54]	[3.03; 4.52]	[2.25; 3.53]	[1.76; 4.64]
HDP	[3.25; 7.46]	[2.25; 5.37]	[2.13; 5.03]	[1.50; 4.03]	[1.27; 5.07]

Table 4: Results of privacy provided by the GDTMs with privacy enhance ρ = 5%

accuracy of our GDTMs as shown in Table 5. On the other hand, we improved the privacy level as can be seen in Table 6.

Method	K-Means					Chameleon				
	K = 2	K = 3	K = 4	K = 5	K = 6	K = 2	K = 3	K = 4	K = 5	K = 6
TDP	1.93	2.13	2.58	3.58	3.68	1.25	1.53	3.67	3.07	3.82
SDP	2.13	2.22	3.87	3.77	4.74	1.98	1.97	3.82	3.35	4.40
RDP	1.93	2.00	3.77	3.15	5.02	1.42	1.68	3.70	3.08	5.15
HDP	1.95	2.05	3.22	3.37	4.02	1.62	1.63	3.77	3.15	3.90

Table 5: Results of misclassification for K-Means and Chameleon with privacy enhance ρ = 10%

It seems that setting ρ = 10%, in our experiments, we could achieve a good compromise between privacy and accuracy. However, the value of ρ depends on the application since the level of privacy can be interpreted in different contexts.

In general, using our procedure to improve privacy, the results revealed that our GDTMs provide practically acceptable values for privacy preserving data clustering. Most importantly, increasing the value of ρ, hardly can someone reverse the transformation applied to the data. In particular, SDP achieved the best values for accuracy and privacy level in most of experiments. However, the other methods also achieved reasonable results. It should be noticed that a security administrator is able to improve the balance between clustering accuracy and privacy by tuning the parameters of the uniform noise vector N and privacy enhance ρ properly.

6. Conclusions

In this paper, we have introduced a family of geometric data transformation methods (GDTMs) which ensure that the mining process will not violate privacy up to a certain degree of security. Our

Method	Privacy Level (%)				
	K = 2	K = 3	K = 4	K = 5	K = 6
TDP	[7.19; 11.78]	[4.97; 8.17]	[4.71; 7.60]	[3.37; 5.87]	[2.86; 7.67]
SDP	[7.71; 13.15]	[4.47; 9.36]	[5.24; 8.90]	[3.92; 7.21]	[3.40; 8.93]
RDP	[8.14; 11.85]	[5.64; 8.92]	[5.52; 7.69]	[4.15; 5.98]	[3.32; 7.83]
HDP	[7.19; 12.54]	[4.97; 8.80]	[4.71; 8.31]	[3.29; 6.61]	[2.80; 8.35]

Table 6: Results of privacy provided by the GDTMs with privacy enhance ρ = 10%

methods were designed to address the privacy preservation in clustering analysis, notably on partition-based and hierarchical methods. Our proposed methods distort only confidential numerical attributes to meet privacy requirements, while preserving general features for clustering analysis. To our best knowledge this is the first effort toward a building block solution for the problem of privacy preserving data clustering. The other approaches in the literature have been restricted basically to address the privacy problem in the context of classification and association rules.

Our contributions in this paper can be summarized as follows: First, we introduced and validated our GDTMs. Our experiments demonstrated that our methods are effective and provide practically acceptable values for balancing privacy and accuracy. We also showed that the traditional ADP method adopted to successfully provide security to databases against disclosure of confidential information has limitations when the perturbed attributes are considered as a vector in the Euclidean space. The main problem is that such a method strongly introduces changes in the distance of points in the Euclidean space leading to the crucial problem of misclassification. Our second contribution refers to the performance measure that quantifies the fraction of data points which are preserved in the corresponding clusters in the distorted database. Misclassification Error measures the amount of legitimate data points that are not well-classified in the distorted database. In addition, we introduced a procedure to improve the privacy level of our GDTMs and validated such procedure in our experiments.

The work presented herein puts forward the need for new concepts and methods to address privacy protection against data mining techniques, notably in data clustering. We address a scenario in which some numerical confidential attributes of a database are distorted and made available for clustering analysis. In this context, users are free to use their own tools so that the restriction for privacy has to be applied before the mining phase on the data itself by data transformation. The transformed database is available for secondary use and must hold the following restrictions: (1) the distorted database must preserve the main features of the clusters mined from the original database; (2) an appropriate balance between clustering accuracy and privacy must be guaranteed.

The results of our investigation clearly indicate that our methods achieved reasonable results and are promising. Currently, we are extending our work in two directions: (a) we are investigating the impact of our GDTMs on other clustering approaches, such as density-based; (b) we are also designing new methods for privacy preserving clustering when considering the analysis of confidential categorical attributes, which requires further exploration.

7. Acknowledgments

Stanley Oliveira was partially supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) of Ministry for Science and Technology of Brazil, under Grant No. 200077/00-7. Osmar Zaïane was partially supported by a Research Grant from NSERC, Canada. Ersin Özğurlu diligently checked the soundness of our GDTMs. The authors would like to acknowledge the helpful comments made by the anonymous reviewers of this paper.

References

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

[2] D. Agrawal and C. C. Aggarwal. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In *Proc. of ACM SIGMOD/PODS*, pages 247–255, Santa Barbara, CA, May 2001.

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

[4] L. Brankovic and V. Estivill-Castro. Privacy Issues in Knowledge Discovery and Data Mining. In *Proc. of Australian Institute of Computer Ethics Conference (AICEC99)*, Melbourne, Victoria, Australia, July 1999.

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

[6] M. J. Culnan. How Did They Get My Name?: An Exploratory Investigation of Consumer Attitudes Toward Secondary Information. *MIS Quartely*, 17(3):341–363, September 1993.

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

[8] 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*, pages 389–398, Florence, Italy, August 1999.

[9] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy Preserving Mining of Association Rules. In *Proc. of the 8th ACM SIGKDD Intl. Conf. on Knowlegde Discovery and Data Mining*, pages 217–228, Edmonton, AB, Canada, July 2002.

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

[11] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, CA, 2001.

[12] P. Jefferies. Multimedia, Cyberspace & Ethics. In *Proc. of International Conference on Information Visualisation (IV2000)*, pages 99–104, London, England, July 2000.

[13] G. H. John. Behind-the-Scenes Data Mining. *Newletter of ACM SIG on KDDM*, 1(1):9–11, June 1999.

[14] G. Karypis, E.-H. Han, and V. Kumar. Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *IEEE Computer*, 32(8):68–75, 1999.

[15] W. Klösgen. Anonymization Techniques for Knowledge Discovery in Databases. In *Proc. of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, pages 186–191, Montreal, Canada, August 1995.

[16] K. Muralidhar, R. Parsa, and R. Sarathy. A General Additive Data Perturbation Method for Database Security. *Management Science*, 45(10):1399–1415, October 1999.

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

[18] S. J. Rizvi and J. R. Haritsa. Privacy-Preserving Association Rule Mining. In *Proc. of the 28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002.

[19] P. Samarati. Protecting Respondents' Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.