A Privacy Preservation Masking Method to Support Business Collaboration

Stanley R. M. Oliveira[1]

[1]Embrapa Informática Agropecuária
Av. André Tosello, 209 – Caixa Postal 6041
13083-886 – Campinas – SP – Brazil

stanley@cnptia.embrapa.br

Abstract. *This paper introduces a privacy preservation masking method to support business collaboration, called Dimensionality Reduction-Based Transformation (DRBT). This method relies on the intuition behind random projection to mask the underlying attribute values subject to cluster analysis. Using DRBT, data owners are able to find a solution that meets privacy requirements and guarantees valid clustering results. DRBT was validated taking into account five real datasets. The major features of this method are: a) it is independent of distance-based clustering algorithms; b) it has a sound mathematical foundation; and c) it does not require CPU-intensive operations.*

1. Introduction

In the business world, data clustering has been used extensively to find the optimal customer targets, improve profitability, market more effectively, and maximize return on investment supporting business collaboration [Berry and Linoff 1997]. Combining different data sources provides better clustering analysis opportunities. For example, it does not suffice to cluster customers based on their purchasing history, but combining purchasing history, vital statistics and other demographic and financial information for clustering purposes can lead to better and more accurate customer behavior analysis.

The fundamental question addressed in this paper is: *how can data owners share data for clustering, supporting business collaboration, without jeopardizing the privacy of their customers?* Clearly, achieving privacy preservation when sharing data for clustering poses new challenges for novel uses of data mining technology. Each application poses a new set of challenges. Let us consider the following real-life motivating example:

Two organizations, an Internet marketing company and an on-line retail company, have datasets with different attributes for a common set of individuals. These organizations decide to share their data for clustering to find the optimal customer targets so as to maximize return on investments. How can these organizations learn about their clusters using each other's data without learning anything about the attribute values of each other?

The above scenario describes a problem of privacy-preserving clustering (PPC), which is referred to as *PPC over vertically partitioned data.* To address such a scenario, this paper introduces a privacy preservation masking method to support business collaboration, called Dimensionality Reduction-Based Transformation (DRBT). This method allows data owners to find a trade-off between privacy, accuracy, and communication cost. Communication cost is the cost (typically in size) of the data exchanged between parties in order to achieve secure clustering.

Dimensionality reduction techniques have been studied in the context of pattern recognition [Fukunaga 1990], information retrieval [Bingham and Mannila 2001], and data mining [Faloutsos and Lin 1995]. To our best knowledge, dimensionality reduction has not been used in the context of data privacy in any detail. The notable exception is our preliminary work presented in [Oliveira and Zaïane 2004].

Although there exists a number of methods for reducing the dimensionality of data, such as feature extraction methods, multidimensional scaling and principal component analysis (PCA), this paper focuses on random projection, a powerful method for dimensionality reduction. The accuracy obtained after the dimensionality has been reduced, using random projection, is almost as good as the original accuracy [Kaski 1999, Achlioptas 2001]. The key idea of random projection arises from the Johnson-Lindenstrauss lemma [Johnson and Lindenstrauss 1984]: "if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved."

The motivation for exploring random projection is based on the following aspects. First, it is a general data reduction technique. In contrast to the other methods, such as PCA, random projection does not use any defined interestingness criterion to optimize the projection. Second, random projection has shown to have promising theoretical properties for high dimensional data clustering [Fern and Brodley 2003, Bingham and Mannila 2001]. Third, despite its computational simplicity, random projection does not introduce a significant distortion in the data. Finally, the dimensions found by random projection are not a subset of the original dimensions but rather a transformation, which is relevant for privacy preservation.

In this work, random projection is used to mask the underlying attribute values subject to clustering, protecting them from being revealed. In tandem with the benefit of privacy preservation, the method DRBT benefits from the fact that random projection preserves the distances (or similarities) between data objects quite nicely. It is shown analytically and experimentally that using DRBT, a data owner can meet privacy requirements without losing the benefit of clustering. The major features of the method DRBT are: a) it is independent of distance-based clustering algorithms; b) it has a sound mathematical foundation; and c) it does not require CPU-intensive operations.

This paper is organized as follows. Section 2 provides the basic concepts that are necessary to understand the issues addressed in this paper. In Section 3, the research problem is described. In Section 4, we introduce the method DRBT to address PPC over vertically partitioned data. Related work is reviewed in Section 5. The experimental results are presented in Section 6. Finally, Section 7 presents our conclusions.

2. Background

2.1. Data Matrix

Objects (e.g., individuals, observations, events) are usually represented as points (vectors) in a multi-dimensional space. Each dimension represents a distinct attribute describing the object. Thus, objects are represented as an $m \times n$ matrix D, where there are m rows, one for each object, and n columns, one for each attribute. This matrix may contain binary, categorical, or numerical attributes. It is referred to as a data matrix, as can be seen in Figure 1.

$$\mathrm{D} = \begin{bmatrix} a_{11} & \dots & a_{1k} & \dots & a_{1n} \\ a_{21} & \dots & a_{2k} & \dots & a_{2n} \\ \vdots & & \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mk} & \dots & a_{mn} \end{bmatrix}$$

Figure 1. The data matrix structure.

$$D_M = \begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \dots & \dots & \dots & & \\ d(m,1) & d(m,2) & \dots & \dots & 0 \end{bmatrix}$$

Figure 2. The dissimilarity matrix structure.

The attributes in a data matrix are sometimes transformed before being used. The main reason is that different attributes may be measured on different scales (e.g., centimeters and kilograms). When the range of values differs widely from attribute to attribute, attributes with large range can influence the results of the cluster analysis. 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 [Han and Kamber 2001]. 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. 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, a value for an attribute A is normalized by subtracting it from the mean of A and then dividing the result by the standard deviation of A.

2.2. Dissimilarity Matrix

A dissimilarity matrix stores a collection of proximities that are available for all pairs of objects. This matrix is often represented by an $m \times m$ table. In Figure 2, we can see the dissimilarity matrix D_M corresponding to the data matrix D in Figure 1, where each element $d(i,j)$ represents the difference or dissimilarity between objects i and j.

In general, $d(i,j)$ is a non-negative number that is close to zero when the objects i and j are very similar to each other, and becomes larger the more they differ. Several distance measures could be used to calculate the dissimilarity matrix of a set of points in d-dimensional space [Han and Kamber 2001]. The Euclidean distance is the most popular distance measure. If i = $(x_{i1}, x_{i2}, ..., x_{in})$ and j = $(x_{j1}, x_{j2}, ..., x_{jn})$ are n-dimensional data objects, the Euclidean distance between i and j is given by:

$$d(i,j) = \left[\ \sum_{k=1}^{n} |x_{ik} - x_{jk}|^2\ \right]^{1/2} \qquad (1)$$

The Euclidean distance satisfies the following constraints: (1) $d(i,j) \geq 0$: distance is a non-negative number; (2) $d(i,i) = 0$: the distance of an object to itself; (3) $d(i,j) = d(j,i)$: distance is a symmetric function; and (4) $d(i,j) \leq d(i,k) + d(k,j)$: distance satisfies the triangular inequality.

2.3. Dimensionality Reduction

When data vectors are defined in a high-dimensional space, it is computationally intractable to use data analysis or pattern recognition algorithms which repeatedly compute similarities or distances in the original data space. It is therefore necessary to reduce the dimensionality before, for instance, clustering the data [Faloutsos and Lin 1995].

The goal of the methods designed for dimensionality reduction is to map d-dimensional objects into k-dimensional objects, where $k \ll d$ [Kruskal and Wish 1978]. One of the methods designed for dimensionality reduction is random projection. This method has been shown to have promising theoretical properties since the accuracy obtained after the dimensionality has been reduced, using random projection, is almost as good as the original accuracy. More formally, when a vector in d-dimensional space is projected onto a random k dimensional subspace, the distances between any pair of points are not distorted by more than a factor of $(1 \pm \epsilon)$, for any $0 < \epsilon < 1$, with probability $O(1/n^2)$, where n is the number of objects under analysis [Johnson and Lindenstrauss 1984].

A random projection from d dimensions to k dimensions is a linear transformation represented by a $d \times k$ matrix R, which is generated by first setting each entry of the matrix to a value drawn from an i.i.d. $\sim N$(0,1) distribution (i.e., zero mean and unit variance) and then normalizing the columns to unit length. Given a d-dimensional dataset represented as an $n \times d$ matrix D, the mapping $D \times R$ results in a reduced-dimension dataset D', i.e.,

$$D'_{n\times k} = D_{n\times d} R_{d\times k} \tag{2}$$

Random projection is computationally very simple. Given the random matrix R and projecting the $n \times d$ matrix D into k dimensions is of the order $O(ndk)$, and if the matrix D is sparse with about c nonzero entries per column, the complexity is of the order $O(cnk)$ [Bingham and Mannila 2001, Johnson and Lindenstrauss 1984].

Clearly, the choice of the random matrix R is one of the key points of interest. The elements r_{ij} of R are often Gaussian distributed, but this need not to be the case. Achlioptas [Achlioptas 2001] showed that the Gaussian distribution can be replaced by a much simpler distribution, as follows:

$$r_{ij} = \sqrt{3} \times \begin{cases} +1 & with\, probability \quad 1/6 \\ 0 & with\, probability \quad 2/3 \\ -1 & with\, probability \quad 1/6 \end{cases} \tag{3}$$

In fact, practically all zero mean, unit variance distributions of r_{ij} would give a mapping that still satisfies the Johnson-Lindenstrauss lemma. Achlioptas' result means further computational savings in database applications since the computations can be performed using integer arithmetics.

3. Privacy-Preserving Clustering: Problem Definition

The goal of privacy-preserving clustering (PPC) is to mask the underlying attribute values of objects subjected to clustering analysis. In doing so, the privacy of individuals would be protected.

The problem of PPC can be stated as follows: Let D be a relational database and C a set of clusters generated from D. The goal is to transform D into D' so that the following restrictions hold:

- A transformation $\mathfrak{T}$ when applied to D must preserve the privacy of individual records, so that the released database D' conceals the values of confidential attributes, such as salary, disease diagnosis, credit rating, and others.

- The similarity between objects in D' must be the same as that one in D, or just slightly altered by the transformation process. Although the transformed database D' looks very different from D, the clusters in D and D' should be as close as possible since the distances between objects are preserved or marginally changed.

3.1. PPC over Vertically Partitioned Data

Consider a scenario wherein k parties, such that $k \geq 2$, have different attributes for a common set of objects, as mentioned in the real-life example, in Section 1. Here, the goal is to do a join over the k parties and cluster the common objects. However, before sharing the data for clustering, each party k must apply random projection to its set of attributes to ensure privacy preservation. The data matrix for this case is given as follows:

$$\begin{array}{c} \vdash \text{ Party 1 } \dashv\vdash \text{ Party 2 } \dashv\vdash \ \ldots \ \dashv\vdash \text{ Party } k \ \dashv \\ \begin{bmatrix} a_{11} \ldots a_{1i} & a_{1i+1} \ldots a_{1j} & & a_{1p+1} \ldots a_{1n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1} \ldots a_{mi} & a_{mi+1} \ldots a_{mj} & & a_{mp+1} \ldots a_{mn} \end{bmatrix} \end{array} \tag{4}$$

In this approach for PPC over vertically partitioned data, one of the parties is the central one which is in charge of merging the data and finding the clusters in the merged data. After finding the clusters, the central party would share the clustering results with the other parties. The challenge here is how to move the data from each party to a central party concealing the values of the attributes of each party. However, before moving the data to a central party, each party must transform its data to protect the privacy of the attribute values. We assume that the existence of an object (ID) should be revealed for the purpose of the join operation, but the values of the associated attributes are private.

4. The Privacy Preservation Masking Method

In this section, it is shown that the triple-goal of achieving privacy preservation and valid clustering results at a reduced communication cost in PPC can be accomplished by random projection. We refer to this solution as the Dimensionality Reduction-Based Transformation (DRBT).

4.1. General Assumptions

The solution to the problem of PPC based on random projection draws the following assumptions:

- The data matrix subjected to clustering contains only numerical attributes that must be transformed (masked) to protect individuals' data values before the data sharing for clustering occurs.

- In PPC over vertically partitioned data, the IDs of the objects are used for the join purposes between the parties involved in the solution.

One interesting characteristic of the solution based on random projection is that, once the dimensionality of a database is reduced, the attribute names in the released database are irrelevant. We refer to the released database as a *disguised database*, which is shared for clustering.

4.2. PPC over Vertically Partitioned Data

In this approach, each party k ($k \geq 2$) must apply the random projection over its dataset and then send the reduced data matrix to a central party. Note that any of the k parties can be the central one. These k parties must satisfy the following constraint: *The attributes split across the k parties are mutually exclusive. More formally, if $A(D_1), A(D_2)..., A(D_k)$ are a set of attributes of the k parties, $\forall i \neq j$ $A(D_i) \cap A(D_j) = \emptyset$. The only exception is that IDs are shared for the join purpose.*

The solution based on random projection for PPC over vertically partitioned data is performed as follows:

- *Step 1 - Individual transformation*: If k parties, $k \geq 2$, share their data in a collaborative project for clustering, each party k_i must transform (by masking) its data using random projection.
- *Step 2 - Data exchanging or sharing*: Once the data are disguised by using random projection, the k parties are able to exchange the data among themselves. However, one party could be the central one to aggregate and cluster the data.
- *Step 3 - Sharing clustering results*: After the data have been aggregated and mined in a central party k_i, the results could be shared with the other parties.

To illustrate how this solution works, let us consider the sample relational database in Table 1. For simplicity, this example is only for one party. This sample contains real data from the Cardiac Arrhythmia Database available at the UCI Repository of Machine Learning Databases [Blake and Merz 1998]. The attributes for this example are: *age*, *weight*, *h_rate* (number of heart beats per minute), *int_def* (number of intrinsic deflections), *QRS* (average of QRS duration in msec.), and *PR_int* (average duration between onset of P and Q waves in msec.).

ID	age	weight	h_rate	int_def	QRS	PR_int
123	75	80	63	32	91	193
342	56	64	53	24	81	174
254	40	52	70	24	77	129
446	28	58	76	40	83	251
286	44	90	68	44	109	128

Table 1. A sample of the cardiac arrhythmia database.

In this case, the dimension of the dataset was reduced from 6 to 3, i.e., 50% of the attributes in the original dataset. Two random projections were used, RP_1 and RP_2. The first refers to the random projection using a random matrix in which each entry was drawn from an i.i.d. N(0,1) distribution and then normalizing the columns to unit length. In the second, each element r_{ij} of the random matrix was computed using Equation (3).

After applying random projection to the dataset, the attribute values of the transformed dataset are masked to preserve the privacy of individuals. Table 2 shows the attribute values of the transformed database with 3 dimensions, using both RP_1 and RP_2. In this table, we have the attributes labeled *Att1*, *Att2*, and *Att3* since we do not know the labels for the disguised dataset.

ID	D' using RP_1			D' using RP_2		
	Att1	Att2	Att3	Att1	Att2	Att3
123	-50.40	17.33	12.31	-55.50	-95.26	-107.96
342	-37.08	6.27	12.22	-51.00	-84.29	-83.13
254	-55.86	20.69	-0.66	-65.50	-70.43	-66.97
446	-37.61	-31.66	-17.58	-85.50	-140.87	-72.74
286	-62.72	37.64	18.16	-88.50	-50.22	-102.76

Table 2. The disguised dataset D' using RP_1 and RP_2.

As can be seen in Table 2, the attribute values are entirely different from those in Table 1.

4.3. How Secure is the DRBT?

In the previous sections, we showed that masking a database using random projection is a promising solution for PPC over vertically partitioned data. Now we show that random projection also has promising theoretical properties for privacy preservation. In particular, we demonstrate that a random projection from d dimensions to k, where $k \ll d$, is a non-invertible transformation.

Lemma 1 *A random projection from d dimensions to k dimensions, where $k \ll d$, is a non-invertible linear transformation.*

Proof: A classic result from Linear Algebra asserts that there is no invertible linear transformation between Euclidean spaces of different dimensions [Auer 1991]. Thus, if there is an invertible linear transformations from $\Re^m$ to $\Re^n$, then the constraint $m = n$ must hold. A random projection is a linear transformation from $\Re^d$ to $\Re^k$, where $k \ll d$. Hence, a random projection from d dimensions to k dimensions is a non-invertible linear transformation. □

Even when sufficient care is taken, a solution that adheres to DRBT can be still vulnerable to disclosure. For instance, if an adversary knows the positions of $d + 1$ points (where d is the number of dimensions) and the distances between these points, then he can make some estimates of the coordinates of all points. In [Caetano 2004], Caetano shows that if an adversary knows the dissimilarity matrix of a set of points and the coordinates of $d + 1$ points, where d is the number of dimensions of the data points, it is possible to disclose the entire dataset. However, this result holds if and only if the $d + 1$ points do not lie in a $(d - 1)$-dimensional vector subspace.

On the other hand, it is important to note that the violation of the solution that adheres to DRBT becomes progressively harder as the number of attributes (dimensions) in a database increases since an adversary would need to know $d + 1$ points to disclose the original data.

4.4. The Accuracy of the DRBT

When using random projection, a perfect reproduction of the Euclidean distances may not be the best possible result. The clusters in the transformed datasets should be equal to those in the original database. However, this is not always the case, and we have some potential problems after dimensionality reduction: a) a noise data point ends up clustered;

	c'_1	c'_2	...	c'_k
c_1	$freq_{1,1}$	$freq_{1,2}$	...	$freq_{1,k}$
c_2	$freq_{2,1}$	$freq_{2,2}$	...	$freq_{2,k}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_k	$freq_{k,1}$	$freq_{k,2}$	...	$freq_{k,k}$

Table 3. The number of points in cluster c_i that falls in cluster c'_j in the transformed dataset.

b) a point from a cluster becomes a noise point; and c) a point from a cluster migrates to a different cluster. In this research, we focus primarily on partitioning methods. In particular, we use K-means [Macqueen 1967, Han and Kamber 2001], one the most used clustering algorithms. Since K-means is sensitive to noise points and clusters all the points in a dataset, we have to deal with the third problem mentioned above (a point from a cluster migrates to a different cluster).

Our evaluation approach focuses on the overall quality of generated clusters after dimensionality reduction. We compare how closely each cluster in the transformed data matches its corresponding cluster in the original dataset. To do so, we first identify the matching of clusters by computing the matrix of frequencies showed in Table 3. We refer to such a matrix as the clustering membership matrix (CMM), where the rows represent the clusters in the original dataset, the columns represent the clusters in the transformed dataset, and $freq_{i,j}$ is the number of points in cluster c_i that falls in cluster c'_j in the transformed dataset.

After computing the frequencies $freq_{i,j}$, we scan the clustering membership matrix calculating precision, recall, and F-measure for each cluster c'_j with respect to c_i in the original dataset [Larsen and Aone 1999]. These formulas are given by the following equations:

$$Precision\ (P) = \frac{freq_{i,j}}{|c'_i|} \tag{5}$$

$$Recall\ (R) = \frac{freq_{i,j}}{|c_i|} \tag{6}$$

where $|X|$ is the number of points in the cluster X.

$$F - measure\ (F) = \frac{2 \times P \times R}{(P + R)} \tag{7}$$

For each cluster c_i, we first find a cluster c'_j that has the highest F-measure among all the c'_l, $1 \leq l \leq k$. Let $F(c_i)$ be the highest F-measure for cluster c_i, we denote the overall F-measure (OF) as the weighted average of $F(c_i)$, $1 \leq i \leq k$, as follows:

$$OF = \frac{\sum_{i=1}^{k} |c_i| \times F(c_i)}{\sum_{i=1}^{k} |c_i|} \tag{8}$$

In section 6., the results of the performance evaluation are based on Equation (8).

4.5. The Complexity of the DRBT

One of the major benefits of a solution that adheres to the DRBT is the communication cost to send a disguised dataset from one party to a central one. In general, a disguised data matrix is of size $m \times k$, where m is the number of objects and k is the number of attributes (dimensions). The complexity of DRBT is of the order $O(m \times k)$, however $k \ll m$.

To quantify the communication cost of one solution, we consider the number of bits or words required to transmit a dataset from one party to a central or third party. Using DRBT, the bit communication cost to transmit a dataset from one party to another is $O(mlk)$, where l represents the size (in bits) of one element of the $m \times k$ disguised data matrix.

5. Related Work

Some effort has been made to address the problem of PPC over distributed data. The existing solutions fall in two categories: *PPC over horizontally partitioned data* and *PPC over vertically partitioned data*. In the former approach, different objects are described with the same schema in all partitions, while in the latter approach, the attributes of objects are split across many partitions.

A solution for PPC over horizontally partitioned data was proposed in [Meregu and Ghosh 2003]. This solution is based on generative models. In this approach, rather than sharing parts of the original data or perturbed data, the parameters of suitable generative models are built at each local site. Then such parameters are transmitted to a central location. The best representative of all data is a certain “mean” model. It was empirically shown that such a model can be approximated by generating artificial samples from the underlying distributions using Markov Chain Monte Carlo techniques. This approach achieves high quality distributed clustering with acceptable privacy loss and low communication cost.

Regarding PPC over over vertically partitioned data, the idea behind this solution is that two or more parties want to conduct a computation based on their private inputs. The issue here is how to conduct such a computation so that no party knows anything except its own input and the results. This problem is referred to as the secure multi-party computation problem [Pinkas 2002]. The existing solution that falls in this category was introduced in [Vaidya and Clifton 2003]. Specifically, a method for k-means was proposed when different sites contain different attributes for a common set of entities. In this solution, each site learns the global clusters, but learns nothing about the attributes at other sites. This work ensures reasonable privacy while limiting communication cost.

In the approach presented in this paper, the attributes of a database are reduced to a smaller number. The idea behind this data transformation is that by reducing the dimensionality of a database to a sufficiently small value, one can find a trade-off between privacy and accuracy. Once the dimensionality of a database is reduced, the released database preserves (or slightly modifies) the distances between data points. In addition, this solution protects individuals’ privacy since the underlying data values of the objects subjected to clustering are completely different from the original ones.

6. Experimental Results

6.1. Datasets

DRBT was validated taking into account five real datasets. These datasets are described as follows:

1. Accidents: This dataset concerning traffic accidents was obtained from the National Institute of Statistics (NIS) for the region of Flanders in Belgium. There are 340,183 traffic accident records included in the dataset, and 18 columns of this dataset were used after removing missing values.
2. Mushroom: This dataset is available at the UCI Repository of Machine Learning Databases [Blake and Merz 1998]. Mushroom contains records drawn from The Audubon Society Field Guide to North American Mushrooms. There are 8,124 records and 23 numerical attributes.
3. Chess: The format for instances in this database is a sequence of 37 attribute values. Each instance is a board-descriptions of a chess endgame. The first 36 attributes describe the board. The last (37th) attribute is the classification: “win” or “nowin”. Chess is available at the UCI Repository of Machine Learning Databases [Blake and Merz 1998] and contains 3,196 records.
4. Connect: This database contains all legal 8-ply positions in the game of connect-4 in which neither player has won yet, and in which the next move is not forced. Connect is composed of 67,557 records and 43 attributes without missing values. This dataset is also available at the UCI Repository of Machine Learning Databases [Blake and Merz 1998].
5. Pumsb: The Pumsb dataset contains census data for population and housing. This dataset is available at http://www.almaden.ibm.com/software/quest. There are 49,046 records and 74 attribute values without missing values.

6.2. Methodology

Two series of experiments were performed to evaluate the effectiveness of DRBT when addressing PPC over vertically partitioned data. Our evaluation approach focused on the overall quality of generated clusters after dimensionality reduction. One question that we wanted to answer was: *What is the quality of the clustering results mined from the transformed data when the data are both sparse and dense?*

Our performance evaluation was carried out through the following steps:

- *Step 1*: we normalized the attribute values of the five real datasets used in our experiments using the z-score normalization. Normalization gives to all attributes the same weight.

- *Step 2*: we considered random projection based on two different approaches to mask a database before data sharing. First, the traditional way to compute random projection, by setting each entry of the random matrix R_1 to a value drawn from an i.i.d. N(0,1) distribution and then normalizing the columns to unit length. Second, we used the random matrix R_2 where each element r_{ij} is computed using Equation (3). We refer to the former random projection as RP_1 and the latter as RP_2. We repeated each experiment (for random projection) 5 times. The results shown later are the average values.

- *Step 3*: we selected K-means to find the clusters in our performance evaluation. K-means is one of the best known clustering algorithm and is scalable [Macqueen 1967, Han and Kamber 2001].
- *Step 4*: we compared how closely each cluster in the transformed dataset matches its corresponding cluster in the original dataset. We expressed the quality of the generated clusters by computing the F-measure given in Equation (8). Considering that K-means is not deterministic (due to its use of random seed selection), we repeated each experiment 10 times. We then computed the minimum, average, maximum, and standard deviation for each measured value of the F-measure. We present the results by showing only the average value.

6.3. Measuring the Effectiveness of the DRBT in each Party Individually

Table 4 shows the results of the F-measure for the dataset Accidents. We reduced the original 18 dimensions to 12. Considering that K-means is not deterministic, we repeated each experiment 10 times and computed the minimum, average, maximum, and standard deviation for each measured value of the F-measure. We simplify the results by showing only one dataset (Accidents). The values of the F-measure for the other datasets followed the same patterns. We present the values of the F-measure only for the random projection RP_2 since its results were slightly better than those yielded by RP_1.

Data Transformation	k = 2				k = 3			
	Min	Max	Avg	Std	Min	Max	Avg	Std
RP_2	0.931	0.952	0.941	0.014	0.903	0.921	0.912	0.009
Data Transformation	k = 4				k = 5			
	Min	Max	Avg	Std	Min	Max	Avg	Std
RP_2	0.870	0.891	0.881	0.010	0.878	0.898	0.885	0.006

Table 4. Average of the F-measure (10 trials) for the Accidents dataset.

We noticed that the values of the F-measure for the Chess and Connect datasets were relatively low when compared with the results of the F-measure for the other datasets. The main reason is that the data points in these datasets are densely distributed. Thus, applying a partitioning clustering algorithm (e.g., K-means) to datasets of this nature increases the number of misclassified data points. On the other hand, when the attribute values of the objects are sparsely distributed, the clustering results are much better.

6.4. Measuring the Effectiveness of the DRBT over Vertically Partitioned Data

Now we move on to measure the effectiveness of DRBT to address PPC over vertically partitioned data. To do so, we split the Pumsb dataset (74 dimensions) from 1 up to 4 parties (partitions) and fixed the number of dimensions to be reduced (38 dimensions). Table 5 shows the number of parties, the number of attributes per party, and the number of attributes in the merged dataset which is subjected to clustering. Recall that in a vertically partitioned data approach, one of the parties will centralize the data before mining.

In this example, each partition with 37, 25, 24, 19, and 18 attributes was reduced to 19, 13, 12, 10, and 9 attributes, respectively. We applied the random projections RP_1 and RP_2 to each partition and then merged the partitions in one central repository.

No. of parties	No. of attributes per party	No. of attributes in the merged dataset
1	1 partition with 74 attributes	38
2	2 partitions with 37 attributes	38
3	2 partitions with 25 and 1 with 24 attributes	38
4	2 partitions with 18 and 2 with 19 attributes	38

Table 5. An example of partitioning for the Pumsb dataset.

Subsequently, we also evaluated the quality of clusters generated by mining the merged dataset and comparing the clustering results with those mined from the original dataset. To do so, we computed the F-measure for the merged dataset in each scenario, i.e., from 1 up to 4 parties. We varied the number of clusters from 2 to 5. Table 6 shows values of the F-measure (average and standard deviation) for the Pumsb dataset over vertically partitioned data. These values represent the average of 10 trials considering the random projection RP_2.

No. of parties	k = 2		k = 3		k = 4		k = 5	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
1	0.909	0.140	0.965	0.081	0.891	0.028	0.838	0.041
2	0.904	0.117	0.931	0.101	0.894	0.059	0.840	0.047
3	0.874	0.168	0.887	0.095	0.873	0.081	0.801	0.073
4	0.802	0.155	0.812	0.117	0.866	0.088	0.831	0.078

Table 6. Average of the F-measure (10 trials) for the Pumsb dataset over vertically partitioned data.

We notice from Table 6 that the results of the F-measure slightly decrease when we increase the number of parties in the scenario of PPC over vertically partitioned data. Despite this fact, the DRBT is still effective to address PPC over vertically partitioned data in preserving the quality of the clustering results as measured by F-measure.

6.5. Discussion on the DRBT When Addressing PPC

The evaluation of the DRBT involves three important issues: security, communication cost, and quality of the clustering results. We discussed the issues of security in Section 4.3 based on Lemma 1, and the issues of communication cost and space requirements in Section 4.5. In this Section, we have focused on the quality of the clustering results. We have learned some lessons from this evaluation, as follows:

- *The application domain of the DRBT*: we observed that the DRBT does not present acceptable clustering results in terms of accuracy when the data subjected to clustering are dense. Slightly changing the distances between data points by random projection results in misclassification, i.e., points will migrate from one cluster to another in the transformed dataset. This problem is somehow understandable since partitioning clustering methods are not effective to find clusters in dense data. The Connect dataset is one example which confirms this finding. On the other hand, our experiments demonstrated that the quality of the clustering results obtained from sparse data is promising.
- *The versatility of the DRBT*: using the DRBT, a data owner can tune the number of dimensions to be reduced in a dataset trading privacy, accuracy, and communication costs before sharing the dataset for clustering.

- *The choice of the random matrix*: from the performance evaluation of the DRBT we noticed that the random projection RP_2 yielded the best results for the values of F-measure, in general. The random projection RP_2 is based on the random matrix proposed in Equation (3).

7. Conclusions

In this paper, we have showed analytically and experimentally that Privacy-Preserving Clustering (PPC) is to some extent possible. To support our claim, we introduced a new masking method to address PPC over vertically partitioned data, called the Dimensionality Reduction-Based Transformation (DRBT). This method was designed to support business collaboration considering privacy regulations. The DRBT relies on the idea behind random projection to mask the underlying attribute values subject to clustering. In doing so, the privacy of individuals would be protected. Random projection has recently emerged as a powerful method for dimensionality reduction. It preserves distances between data objects quite nicely, which is desirable in cluster analysis.

We evaluated the DRBT taking into account three important issues: security, communication cost, and accuracy (quality of the clustering results). Our experiments revealed that using DRBT, a data owner can meet privacy requirements without losing the benefit of clustering since the similarity between data points is preserved or marginally changed. From the performance evaluation, we suggested guidance on which scenario a data owner can achieve the best quality of the clustering when using the DRBT. In addition, we suggested guidance on the choice of the random matrix to obtain the best results in terms of the error produced on the datasets and the values of F-measure.

The highlights of the DRBT are as follows: a) it is independent of distance-based clustering algorithms; b) it has a sound mathematical foundation; and c) it does not require CPU-intensive operations.

Currently, we are expanding our work with a probabilistic analysis to supplement the empirical results, which require further exploration. In particular, we are interested in analyzing under which conditions privacy can be violated.

8. Acknowledgments

The author would like to thank Osmar Zaïane for his comments and suggestions in the early stage of this work.

9. References

References

Achlioptas, D. (2001). Database-Friendly Random Projections. In *Proc. of the 20th ACM Symposium on Principles of Database Systems*, pages 274–281. Santa Barbara, CA, USA.

Auer, J. W. (1991). *Linear Algebra With Applications*. Prentice-Hall Canada Inc., Scarborough, Ontario, Canada.

Berry, M. and Linoff, G. (1997). *Data Mining Techniques - for Marketing, Sales, and Customer Support*. John Wiley and Sons, New York, USA.

Bingham, E. and Mannila, H. (2001). Random Projection in Dimensionality Reduction: Applications to Image and Text Data. In *Proc. of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250. San Francisco, CA, USA.

Blake, C. and Merz, C. (1998). UCI Repository of Machine Learning Databases, University of California, Irvine, Dept. of Information and Computer Sciences.

Caetano, T. S. (2004). *Graphical Models and Point Set Matching*. PhD thesis, Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil.

Faloutsos, C. and Lin, K.-I. (1995). FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets. In *Proc. of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 163–174. San Jose, CA, USA.

Fern, X. Z. and Brodley, C. E. (2003). Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *Proc. of the 20th International Conference on Machine Learning (ICML 2003)*. Washington DC, USA.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. 2nd. Edition. Academic Press.

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

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of Lipshitz Mapping Into Hilbert Space. In *Proc. of the Conference in Modern Analysis and Probability*, pages 189–206. volume 26 of Contemporary Mathematics.

Kaski, S. (1999). Dimensionality Reduction by Random Mapping. In *Proc. of the International Joint Conference on Neural Networks*, pages 413–418. Anchorage, Alaska.

Kruskal, J. B. and Wish, M. (1978). *Multidimensional Scaling*. Sage Publications, Beverly Hills, CA, USA.

Larsen, B. and Aone, C. (1999). Fast and Effective Text Mining Using Linear-Time Document Clustering. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16–22. San Diego, CA, USA.

Macqueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. In *Proc. of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. Berkeley: University of California Press, Vol. 1.

Meregu, S. and Ghosh, J. (2003). Privacy-Preserving Distributed Clustering Using Generative Models. In *Proc. of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, pages 211–218. Melbourne, Florida, USA.

Oliveira, S. R. M. and Zaïane, O. R. (2004). Privacy-Preserving Clustering by Object Similarity-Based Representation and Dimensionality Reduction Transformation. In *Proc. of the Workshop on Privacy and Security Aspects of Data Mining (PSADM'04) in conjunction with the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 21–30. Brighton, UK.

Pinkas, B. (2002). Cryptographic Techniques For Privacy-Preserving Data Mining. *SIGKDD Explorations*, 4(2):12–19.

Vaidya, J. and Clifton, C. (2003). Privacy-Preserving K-Means Clustering Over Vertically Partitioned Data. In *Proc. of the 9th ACM SIGKDD Intl. Conf. on Knowlegde Discovery and Data Mining*, pages 206–215. Washington, DC, USA.