Privacy Preservation in k-Means Clustering by Cluster Rotation

S. S. Shivaji Dhiraj, Ameer M. Asif Khan, Wajhiulla Khan, Ajay Challagalla
Department of Computer Science and Engineering
National Institute of Technology, Warangal
India
shivajidhiraj@yahoo.com ameerfaisal89@gmail.com wajhiulla.khan@gmail.com cjajay@gmail.com

***Abstract*—The use of clustering as a data analysis tool has raised concerns about the violation of individual privacy. This paper proposes a data perturbation technique for privacy preservation in k-means clustering. Data objects that have been partitioned into clusters using k-means clustering are perturbed by performing geometric transformations on the clusters in such a way that the object membership of each cluster and orientation of objects within a cluster remain the same. This geometric transformation is achieved through cluster rotation, i.e., every cluster is rotated about its own centroid. The clusters are first displaced away from the mean of the entire dataset so that no two clusters overlap after the subsequent cluster rotation. We analyze the privacy measure offered by this data perturbation technique and prove that a dataset perturbed by this method cannot be easily reverse engineered, yet is still relevant for cluster analysis.**

Keywords-Data Mining, Clustering, Data Perturbation, Privacy Preservation, Geometric Transformation

I. INTRODUCTION

Advances in data collection and storage technologies have led to the proliferation of large databases that may contain sensitive information about organizations and individuals. Data mining technology is used to extract useful knowledge from this data. On the one hand, such data is an important asset to business organizations and governments for decision-making processes and to provide social benefits, such as medical research, crime reduction, national security, etc. On the other hand, analyzing such data opens new threats to privacy and autonomy of the individual if not done properly [6]. Therefore, individual privacy concerns limit the willingness of the data custodians to share data [1].

Hence, a fruitful direction for future data mining research will be the development of techniques that incorporate privacy concerns [2].This has led to the development of a relatively new and rapidly emerging research area of privacy preserving data mining. There are many approaches which have been adopted for privacy preserving data mining. They can be based on the following dimensions [3]:

- data distribution
- data modification
- data mining algorithm
- data or rule hiding
- privacy preservation

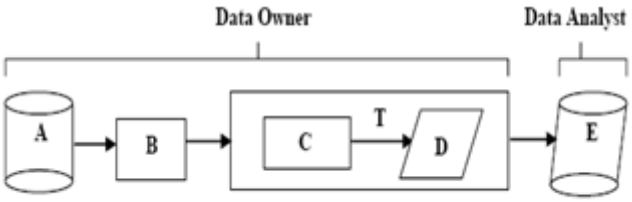
In essence, data perturbation is accomplished by the alteration of an attribute value by a new value, and hence, data perturbation techniques can be classified under the dimension of data modification. Thus, in this way, confidentiality of individual objects can be maintained. The privacy preserving properties of such databases are a result of the perturbation. This technique of data perturbation is applied for privacy preservation in k-means clustering.

II. MOTIVATION

The primary task in data mining is the development of models about aggregated data [2]. Therefore, we can develop accurate models without access to precise information in individual data records. Cluster analysis is a data mining tool for unsupervised learning. We apply the technique of data perturbation for the problem of privacy preservation in k-means clustering.

The k-means clustering problem is one of the most-explored problems in data mining to date [5]. The k-means clustering algorithm partitions a set of n-dimensional data objects into k clusters. However, revealing the exact values of individual objects to the data analyst constitutes a violation of privacy. Our approach towards privacy preserving clustering is driven by the fact that the data analyst need not be given access to the exact value of individual records. Instead, they are presented with a modified view of the data so that the privacy of individual objects is preserved. Therefore, the data in these clusters is perturbed by geometrically transforming the clusters. This perturbation is done by the data owner and is achieved by first displacing the clusters from the centroid of the entire dataset and then rotating these clusters about their respective cluster centroids. The technique is schematically represented in Figure I.

978-1-4244-4547-9/09/$26.00 ©2009 IEEE TENCON 2009



A: the original dataset;
B: the clustering process;
T: the data perturbation technique;
C: cluster displacement;
D: cluster rotation;
E: the perturbed dataset.

Figure I. Schematic Representation of the Data Perturbation Method.

III. RELATED WORK

Some effort has been made to address the problem of privacy preservation in data mining by Agrawal and Srikant [2]. In particular, the problem of Privacy Preserving Clustering has been previously addressed in [6], [7] and [14]. Oliviera and Zaiane [6] addressed the privacy preserving data clustering problem using randomization techniques. Vaidya and Clifton [7] presented a privacy preserving k-means clustering protocol on vertically partitioned data using cryptographic techniques.

The problem of protecting the underlying attribute values when sharing data for clustering has been addressed in [8]. Data Obfuscation (DO) techniques distort data in order to hide information. One application area for DO is privacy preservation. The data obfuscation technique called Nearest Neighbor Data Substitution (NeNDS), which has strong privacy preserving properties and maintains data clusters, has been explored in [4].

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 has been introduced in [6]. A set of hybrid data transformations have been introduced to preserve the confidentiality of categorical data in clustering [9].

The work proposed here differs from previous work in privacy preserving clustering mentioned above. In the proposed technique, the geometric transformations, viz., cluster displacement and cluster rotation, are performed on the individual clusters, and not on the dataset as a whole. Therefore, objects do not migrate from one cluster to another after the perturbation, i.e., the misclassification error (M_E) is 0%.

In [14], a method is proposed for privacy preserving clustering through cluster bulging. In this method, every cluster is geometrically scaled by a randomly generated cluster scaling factor. Cluster scaling changes the intra-cluster distances of objects, but the technique proposed in this paper perturbs data through cluster rotation which being an isometric transformation preserves the intra-cluster distances.

The quality of the perturbed data obtained in the proposed technique of cluster rotation is substantially high. Hence, this method of privacy preserving clustering has a wide scope of application in market research, e-governance and medical data analysis.

IV. OUR APPROACH

A. The k-Means Clustering Algorithm

We briefly review the k-means clustering algorithm. (For a more detailed description, see [10]). The k-means algorithm takes the input parameter k and partitions a dataset of n-dimensional objects into k clusters so that the resulting intra-cluster similarity is high but the inter-cluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity. The k-means procedure is summarized in Algorithm I.

Algorithm I: The k-Means Algorithm for Clustering.

Input:
- k: the number of clusters
- D: a dataset containing m objects

Output: Dataset D partitioned into a set of k clusters.
Method:
(1) Arbitrarily choose k objects from D as the initial cluster centroids;
(2) **repeat**
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) **until** no change;

Algorithm I is the standard k-means algorithm. There have been several modifications and improvements to it. An improved algorithm for learning k has been presented in [11]. Also, there are alternatives to the k-means algorithm that find better clusterings [12]. For the above k-means algorithm, the similarity measure is considered to be Euclidean distance. We illustrate the data perturbation technique for the standard k-means algorithm.

B. The Data Perturbation Method

Revealing the exact values of data objects to a data analyst violates individual privacy. In order to achieve the objective of privacy preservation, the values of the objects are perturbed in such a way that the object membership and shape of each cluster remains the same after the perturbation.

For the purpose of illustrating the data perturbation technique, we assume that the initial k cluster centroids from the dataset D have been randomly chosen. However, depending on the application of clustering, the data owner can apply techniques to refine the k initial points for k-means clustering [13] and apply the data perturbation technique, before revealing the perturbed data to the data analyst. The

data perturbation technique involves two steps: cluster displacement and cluster rotation.

1) Cluster Displacement

Data perturbation is performed by first displacing the clusters obtained by k-means clustering by a *factor* 'λ' from the mean of the entire dataset, so that no two clusters overlap after the subsequent cluster rotation process. Two clusters C_i and C_j are said to overlap if an object from C_i is closer to the centroid of C_j than it is to the centroid of C_j or vice-versa. Consider an n-dimensional dataset D consisting of 'm' objects. Each object A_i in D can be represented as $A_i = (a_{i1}, a_{i2}, a_{i3} \ldots a_{in})$. The mean of the entire dataset is $G = \frac{1}{m} \sum_{i=1}^{m} A_i$.

The centroid of the objects in a cluster is equal to their attribute-wise arithmetic mean. Every cluster must be displaced by a factor λ along the line joining its *cluster centroid* to the mean of the entire dataset, in order not to distort the relative arrangement of the clusters with each other, i.e., the *displacement vector* between every cluster centroid G_i and the mean of the entire dataset G must be multiplied by a factor λ.

To get the new value of G_i, this product vector, $\lambda \times \mathbf{GG_i}$, must be added to the *position vector* **OG** of the mean G, where 'O' represents the origin. The position vector of centroid G_i of cluster C_i is $\mathbf{OG_i}$. After C_i is displaced, the position vector of its centroid G_i' is given by

$$\mathbf{OG_i'} = \mathbf{OG} + \mathbf{GG_i'} .$$

GG_i' is also a vector in the direction of the vector $\mathbf{GG_i}$ having λ times the magnitude of GG_i. Therefore

$$\mathbf{OG_i'} = \mathbf{OG} + \lambda \times \mathbf{GG_i} .$$

Every object in each cluster has to be shifted in such a way that the orientation of objects in the clusters remains the same. This means the position vector of every object A_{ir} in the i^{th} cluster, $\mathbf{OA_{ir}}$ has to be added with this difference vector to get the new position vector of the object after the displacement of the cluster. Therefore, the position vector of every object A_r in every displaced cluster C_i' is given by

$$\mathbf{OA_{ir}'} = \mathbf{OA_{ir}} + \mathbf{G_iG_i'}.$$

$$= \mathbf{OA_{ir}} + \lambda \times \mathbf{GG_i} - \mathbf{GG_i} .$$

$$\mathbf{OA_{ir}'} = \mathbf{OA_{ir}} + (\lambda - 1) \times \mathbf{GG_i} . \quad (1)$$

The new coordinates of the object A_{ir} after C_i is displaced are given by

$$A_{ir}' = \{a_{ir1} + (\lambda - 1)(g_{i1} - g_1), a_{ir2} + (\lambda - 1)(g_{i2} - g_2), \ldots ,$$

$$a_{irn} + (\lambda - 1)(g_{in} - g_n)\} . \quad (2)$$

Using (2), the coordinates of each object in the displaced clusters can be computed. After each cluster is displaced by a factor λ form the mean of the entire dataset, it is rotated about its own cluster centroid. The computation of the value of λ is discussed in the section after cluster rotation.

2) Cluster Rotation

Cluster rotation is performed by a set of isometric transformations on each object within every cluster, i.e., the set of isometric transformations remain the same for all the objects within a given cluster, but, they vary from one cluster to another cluster.

An *isometry* (also called congruence) is a special class of geometric transformations [15]. The essential characteristic of an isometry is that distances between objects are preserved in the process of moving them in an *n*-dimensional Euclidean space. In other words, distance must be an invariant property. Cluster rotation is an isometric transformation [15], where the rotational transformation T is applied to every r^{th} object A_{ir} in a cluster C_i with centroid G_i, such that

$$|T(A_{ir}) - G_i| = |A_{ir} - G_i| .$$

The rotation of clusters is performed by *pair-wise attribute distortion* of all the objects within a cluster. Every cluster is rotated about its own cluster centroid and thus, the cluster centroids themselves do not undergo any change during cluster rotation. The origin is assumed to be at the centroid of every cluster during its rotation.

To perform pair-wise attribute distortion of all the objects within a cluster C_i, $\left\lceil \frac{n}{2} \right\rceil$ pairs of attributes of the dataset are selected such that the values of no attribute remain the same after perturbation, where n is the number of attributes. The vector $V = (a_{irp}, a_{irq})$ representing the values of attributes p and q of an object A_{ir} in C_i (with respect to the cluster centroid of C_i) is transformed into $V' = (a_{irp}', a_{irq}')$ using the matrix representation $V' = V \times R$, where R is the rotational transformation matrix. For a pair of attributes p and q of objects in C_i, the rotational transformation matrix can be represented as

$$R_{ipq} = \begin{pmatrix} \cos\theta_{ipq} & -\sin\theta_{ipq} \\ \sin\theta_{ipq} & \cos\theta_{ipq} \end{pmatrix} \quad (3)$$

In (3), θ_{ipq} is the rotation angle for the pair of attributes p and q. Hence, if V and V' are the values of attributes p and q of an object A_{ir} as measured from the centroid of the cluster containing A_{ir}, represented as 1×2 matrices, and R is the 2×2 rotational transformation matrix shown in (2), then

$$V_{ipq}' = V_{ipq} \times R_{ipq} . \quad (4)$$

To achieve the objective of privacy preserving clustering by rotational data perturbation, the rotational transformation matrix is applied to every pair of attributes for each object within a particular cluster, i.e., to perform this transformation, a pair of attributes must be distorted at a time as shown in (4). Hence, to distort the values of all the attributes of an n-dimensional object the minimum number of distortions required would be $\frac{n}{2}$ if n is even and $\frac{n+1}{2}$ if n is odd, that is, $\left\lceil \frac{n}{2} \right\rceil$ rotations are required.

In this method, the rotation angle θ_{pq} is generated randomly for every pair of attributes p and q of the objects within a cluster C_i. θ_{pq} is constant for a given pair of attributes p and q within a cluster C_i, i.e., the rotational transformation matrix R is the same for all the objects within a cluster for a given pair of attributes p and q. Therefore, the shape of a cluster and the relative arrangement of objects within a cluster remain the same during the cluster rotation process.

3) Calculation of Cluster Displacement Factor λ

In order to achieve the objective of preserving the shape and object membership of a cluster after rotation about its cluster centroid, every cluster has to be displaced by a factor λ along the line joining the cluster centroid to the mean of the entire dataset in such a way that no two clusters overlap during the subsequent cluster rotation.

Every object belongs to a certain cluster by virtue of the distance of that object from the centroid of that cluster. Therefore, if no two clusters are to overlap after the rotation process, the clusters must be displaced to such an extent that the object in C_i which is farthest from its centroid G_i must be closer to G_i than it is to any other cluster centroid G_j, i.e., for every pair of clusters C_i and C_j we have to find a λ_{ij} such that the above condition is satisfied. We consider the following worst-case scenario.

Let A_i and A_j be the objects farthest from G_i and G_j, in clusters C_i and C_j respectively. The worst-case scenario occurs when the perturbed objects A_i' and A_j' lie on the line joining G_i and G_j after rotation of C_i and C_j. Therefore, C_i and C_j must be displaced from the mean of the entire dataset so that

$$|\mathbf{G_i'G_j'}| > 2 \times |\mathbf{A_i''G_i'}| . \quad (5)$$

and

$$|\mathbf{G_i'G_j'}| > 2 \times |\mathbf{A_j''G_j'}| . \quad (6)$$

where G_i' and G_j' are the coordinates of the centroids of the displaced clusters C_i' and C_j' and A_i" and A_j" are the coordinates of A_i and A_j after cluster displacement and cluster rotation. From (5) and (6), we get

$$|\mathbf{G_i'G_j'}| > 2 \times \text{maximum } \{|\mathbf{A_i''G_i'}|, |\mathbf{A_j''G_j'}|\} .$$

$$|\mathbf{G_i'G_j'}| = \lambda_{ij} \times |\mathbf{G_iG_j}| .$$

Therefore, λ_{ij} must satisfy the condition

$$\lambda_{ij} > 2 \times \text{maximum } \{|\mathbf{A_i''G_i'}|, |\mathbf{A_j''G_j'}|\} / |\mathbf{G_iG_j}| . \quad (7)$$

Hence, λ_{ij} can be any value greater than the right hand side of (7). So we consider

$$\lambda_{ij} = 2 \times \text{maximum } \{|\mathbf{A_i''G_i'}| + 1, |\mathbf{A_j''G_j'}| + 1\} / |\mathbf{G_iG_j}| . \quad (8)$$

The value of λ_{ij} is calculated for every pair of clusters C_i and C_j in the dataset. However, to ensure that the relative arrangement of clusters remains same, the maximum of all the values of λ_{ij} is taken as the displacement factor λ for the dataset. If the maximum of all the values of λ_{ij} is less than 1, i.e., the clusters are already far apart, then λ is taken as 1 and no cluster displacement is needed. Since cluster rotation is an isometric transformation, the intra-cluster distances of objects within each cluster remains the same before and after rotation.

Thus, the coordinates of every object after the cluster displacement process are recomputed using (2). These displaced clusters are rotated about their respective cluster centroids using (4). The data perturbation technique is summarized in Algorithm II.

Algorithm II: Algorithm for Data Perturbation by Cluster Rotation.

Input: Dataset D containing 'm' n-dimensional objects in k clusters

Output: Perturbed dataset D' containing 'm' n-dimensional objects in k clusters

Method:

(1) For each cluster C_i, calculate the maximum distance d_i of an object in C_i from its centroid G_i;
(2) For each pair of clusters C_i and C_j, calculate $\lambda_{ij} > 2 \times \max \{ d_i, d_j \} /$ (distance between the centroids, G_i and G_j);
(3) Displace the clusters by λ, the maximum of all λ_{ij}s around the mean of the dataset if $\lambda > 1$;
(4) For each object in cluster C_i, apply rotational transform about centroid G_i to each pair of attributes p and q, such that $V_{pq}' = V_{pq} \times R_{pq}$, where θ_{pq} is constant for all data objects in C_i;

The input to the data perturbation technique is a dataset containing k clusters. These k clusters are formed using the standard k-means clustering algorithm or a suitable variation of it. The appropriate variant of the k-means algorithm to be used is chosen by the data owner, considering the requirements of the data analyst. Therefore, the data analyst is presented with the perturbed dataset, along with the k final cluster centroids. Thus data analyst can use the k clusters obtained for exploring the interrelationships among samples to make an assessment of the sample structure.

4) Normalization of the Perturbed Dataset

If the domains of the attribute values of the objects are transformed into other undesirable domains (e.g., negative values introduced to a non-negative attribute domain) as a result of the perturbation, then the dataset can be normalized by scaling it with a suitable multiplicative factor around the mean of the entire dataset.

However, if the statistical summary of the data is to be preserved, the cluster centroids must remain the same in the perturbed dataset and the original dataset. To achieve this, the scaling factor for normalization can be taken as $\frac{1}{\lambda}$. It must be noted that this normalization process does not affect the quality of the clusters formed as the entire dataset is being scaled by the same factor around the mean of the dataset.

5) Computational Costs

The proposed data perturbation technique has a complexity of the order

$$O(k \times m_{avg} \times n) .$$

where k is the number of clusters, m_{avg} is the average number of objects in a cluster and n is the number of attributes. A similar technique proposed by Oliveira et al [6] has a complexity of O(m × n), which is comparable to that of the proposed technique. Unlike the rotation based transformation in [6], the proposed technique requires the input data to be clustered using the k-means clustering algorithm. Also, for a given pair of attributes, the random function which generates the rotation angles has to be invoked k times, once for each cluster whereas in [6] rotation angles for pair-wise attribute distortion are computed only once for the entire dataset. A similar comparison can be made by another similar technique proposed by Ketel et al [18].

Thus, the proposed technique is computationally more expensive than some comparable methods. However, these additional computational costs are effectively offset by the increased security measure offered by this method through independent rotation of clusters to perturb the data.

V. EXPERIMENTS AND ANALYSIS

A. Quantifying Privacy

Traditionally, the privacy provided by a perturbation technique has been measured as the variance between the actual and the perturbed values [16]. This measure is given by Var(X – Y) where X represents a single original attribute and Y the distorted attribute. Privacy level can be specified by the metric

$$Sec = \frac{Var(X - Y)}{Var(X)} . \quad (9)$$

The value of Sec can be computed as

$$Sec = \frac{Var(|A_i'' - A_i|)}{Var(|A_i|)} . \quad (10)$$

where A_i" represents the perturbed object, and A_i represents the unperturbed object; 'm' is the number of objects. Therefore, the value of Sec for a perturbed dataset indicates the degree of perturbation. Thus, Sec can be taken as the measure of privacy offered by this technique.

B. Hiding Failure

Hiding failure is the portion of sensitive information that is not hidden by the application of a privacy preservation technique [19]. The percentage of sensitive information that is still discovered, after the data has been sanitized gives an estimate of the hiding failure parameter. Most of the developed privacy preserving algorithms are designed with the goal of obtaining zero hiding failure. In [20], Oliveira and Zaiane define the hiding failure (HF) as the percentage of restrictive patterns that are discovered from the sanitized database. It is measured as follows:

$$HF = \frac{\#R_p(D')}{\#R_p(D)} . \quad (11)$$

where $\#R_p(D)$ and $\#R_p(D')$ denote the number of restrictive patterns discovered from the original data base D and the sanitized database D′ respectively. Ideally, the HF is 0.

In data perturbation by cluster rotation, every cluster is rotated about its centroid, i.e., the value of every object in the dataset is modified. Therefore, no sensitive information is discovered after the application of the data perturbation method. Hence, the Hiding Failure for the proposed data perturbation technique is 0.

C. Misclassification Error (M_E)

The quality of the perturbed data obtained in this method, with respect to cluster analysis, can be measured by the misclassification error percentage. Misclassification error is measured in terms of the percentage of legitimate data points that are not well-classified in the distorted database. A lower misclassification error is desirable as it signifies a smaller change in the object membership of clusters. Ideally, the misclassification error should be 0%. The misclassification error, denoted by M_E, is measured as

$$M_E = \frac{1}{k} \times \sum_{i=1}^{k} (|C_i| - |C_i'|) . \quad (12)$$

where k is the number of clusters in the dataset, and C_i' is the perturbed cluster corresponding to the unperturbed cluster C_i.

In [9], a set of hybrid transformations has been introduced to ensure privacy of categorical data in clustering. The misclassification errors obtained after applying the hybrid data transformation techniques for various noise levels are computed and they are found to be the least for a noise level of 75%. However, in the proposed data perturbation technique the misclassification error is always 0% as the clusters are displaced in such a way that there is no overlapping after the subsequent rotation process.

D. Data Usability

The term data usability refers to the ability of a data perturbation technique to provide accurate aggregate information. An ideal data perturbation technique is one that preserves both statistical as well as clustering information [4]. Data randomization techniques [2], which obfuscate data by the addition of random noise to the original data, can be tailored to preserve statistical information. However, the inherent clusters in the original data are distorted because of the addition of random noise.

The technique proposed in this paper perturbs data by isometric transforms such as cluster rotation. The misclassification error is zero as the original clusters are preserved. After perturbation process, the entire dataset is normalized by scaling it around its mean with $\frac{1}{\lambda}$ as the scaling factor. In this way, the cluster centroids are brought back to their original positions and the aggregate information of the clusters is preserved. Therefore, data perturbed by this technique is still usable for cluster analysis.

E. Data Privacy Attack Model

Existing work on the privacy analysis of data perturbation techniques has primarily considered a model where the adversary correlates perturbed data objects with data from other publicly accessible databases in order to reconstruct sensitive information of interest [4]. In this model, the worst-case scenario would be a brute force attack by an adversary that has *a priori* information about the steps involved in the data perturbation process, and some of the original data objects themselves. The adversary could then try to reverse engineer the entire dataset. The effectiveness of a data perturbation technique is measured by the ability of the perturbed dataset to withstand such an attack.

For a dataset perturbed by cluster rotation, every cluster is rotated by a specific set of angles. Therefore, in case of a brute force attack, even if the adversary can correlate an object in the original dataset with the corresponding one in the perturbed dataset, then the adversary would only be able to reconstruct the original objects within that cluster because every other cluster is perturbed with a different set of angles. Hence, it is extremely difficult to reverse engineer a dataset perturbed by this technique.

F. Experimental Results

We implemented the proposed technique for the standard k-means algorithm on an open source LINUX platform using C++. We tested the technique on standard datasets, each having 'm' n-dimensional tuples, and evaluated the values of Sec, and M_E. The number of clusters k was determined using the rule of thumb [17] given by $k = \sqrt{m/2}$.

Table I lists the experimental results for four sample datasets. The misclassification error for all datasets perturbed by cluster rotation is 0%. This is because the geometric transformations are applied to the clusters so as to preserve their object membership and the orientation of the objects within the clusters. Other methods for privacy preserving clustering like [4] and [9] offer only a non-zero misclassification error. Therefore, data perturbation by cluster rotation does not affect the quality of the data.

TABLE I. RESULTS FOR SAMPLE DATASETS

Dataset	m	N	k	HF	Sec (%)	M_E (%)
D_1	123	8	8	0	11.6826	0.0
D_2	569	30	17	0	2.6701	0.0
D_3	2100	19	33	0	15.2784	0.0
D_4	4601	58	48	0	8.53731	0.0

D1. Income Limit – 2008, California USA [Source: Department of Housing and Community Development, State of California, USA].

D2. Original Wisconsin Breast Cancer Database [Source: UCI Machine Learning Repository (MLR)].

D3. Image features extracted from a Corel image collection [Source: UCI MLR].

D4. Spam base classifying email as spam and non-spam [Source: UCI MLR].

Since every cluster is rotated about its centroid by pair-wise attribute distortion, no sensitive information is revealed to the data analyst. Hence, the hiding failure in this method is 0.

The values in Table I have been obtained after normalizing the perturbed dataset by scaling it around its mean by a factor of $\frac{1}{\lambda}$. That means the cluster centroids remain the same in the original dataset and the perturbed dataset. Thus, the statistical summary of the data within the clusters is preserved. However, this does not affect the degree of perturbation, as the value of Sec is still quite high.

CONCLUSION AND FURTHER WORK

The basic premise of privacy preserving data mining is that the data analyst need not be revealed the exact values of sensitive data objects. Instead, the data analyst is presented an altered view of the dataset. This paper proposed a technique for privacy preserving clustering by cluster rotation. The exact values of individual objects are not revealed and the privacy of individual objects is preserved but the perturbed dataset is still relevant for cluster analysis.

Further work lies in the development of suitable variations of the cluster rotation technique for privacy preservation in other partitional clustering methods and hierarchical clustering methods.

REFERENCES

[1] Vladimir Estivill-Castro and Chris Clifton, Preface: Proc. of the ICDM 2002 Workshop on Privacy, Security, and Data Mining.

[2] R. Agrawal and R. Srikant, "Privacy preserving data mining", ACM SIGMOD Conf. Management of Data, pp. 439-450, May 2000.

[3] Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin and Yannis Theodoridis: "State-of-

the-art in privacy preserving data mining", ACM SIGMOD Record, v.33 n.1, March 2004.

[4] Rupa Parameswaran and Douglas M. Blough, "Privacy preserving data obfuscation for inherently clustered data", International Journal of Information and Computer Security, v.2 n.1, p.4-26, January 2008.

[5] Paul Bunn and Rafail Ostrovsky, "Secure two-party k-means clustering", the 14th ACM conference on Computer and Communications Security, October 28-31, 2007, Alexandria, Virginia, USA.

[6] S. Oliveira and O. R. Za¨ıane, "Privacy preserving clustering by data transformation", the 18th Brazilian Symposium on Databases, pages 304–318, 2003.

[7] J. Vaidya and C. Clifton, "Privacy preserving k-means clustering over vertically partitioned data", the 9th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining. ACM Press, 2003.

[8] Stanley R.M. Oliveira and Osmar R. Za¨ıane "Achieving privacy preservation when sharing data for clustering", the Workshop on Secure Data Management in a Connected World (SDM'04) in conjunction with VLDB'2004, Toronto, Canada, pg. 67-82.

[9] R.R. Rajalaxmi and A.M. Natarajan, "An effective data transformation approach for privacy preserving clustering", Journal of Computer Science 4(4): 320-326, 2008, Science Publications.

[10] Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, pg. 402.

[11] Greg Hamerly, Charles Elkan, "Learning the k in k-means", the Seventeenth Annual Conference on Neural Information Processing Systems (NIPS), pg. 281-288, December 2003.

[12] Greg Hamerly, Charles Elkan, "Alternatives to the k-means algorithm that find better clusterings", the Eleventh International Conference on Information and Knowledge Management, November 04-09, 2002, McLean, Virginia, USA.

[13] Paul S. Bradley and Usama M. Fayyad, "Refining initial points for k-means clustering", the Fifteenth International Conference on Machine Learning, p.91-99, July 24-27, 1998.

[14] Mohammad Ali Kadampur, D.V.L.N Somayajulu, S.S. Shivaji Dhiraj and Shailesh G.P. Satyam, "Privacy preserving clustering by cluster bulging for information sustenance", of the 4th International Conference on Information and Automation for Sustainability(ICIAfS '08), Colombo, Sri Lanka, December 2008.

[15] H. T. Croft, K. J. Falconer and R. K. Guy, "Unsolved problems in geometry", v.2. New York: Springer-Verlag, 1991.

[16] Muralidhar, K., R. Parsa and R. Sarathy, "A general additive data perturbation method for database security", J. Mgmt. Science, 1999, Vol: 45, pp: 1399-1415.

[17] K.V. Mardia, J.T. Kent and J. M. Bibby, Multivariate Analysis, Academic Press, pp.365.

[18] Mohammed Ketel and Abdollah Homaifar, "Privacy-preserving mining by rotational data transformation", the 43rd ACM Southeast Conference, March 18-20, 2005, Kennesaw, GA, USA.

[19] Elisa Bertino, Dan Lin and Wei Jiang, "A survey of quantification of privacy preserving data mining algorithms", Privacy-Preserving Data Mining (Models and Algorithms), Charu C. Aggarwaland Philip S. Yu (Eds.), Springer-Verlag, 2008.

[20] S.R.M. Oliveira, O. Za¨ıane, "Privacy preserving frequent itemset mining", IEEE ICDM Workshop on Privacy, Security and Data Mining, vol. 14, pp. 43–54 (2002)