IOSR Journal of Computer Engineering (IOSR-JCE)
ISSN : 2278-0661, ISBN : 2278-8727, PP : 32-38
www.iosrjournals.org

Optimized Protocol for Privacy Preserving Clustering

Miss Mane P.B.[1] Mr Kadam S.R.[2] Mr Bugade V.M.[3]
[1,2,3] *(Dept of CSE, Shivaji University Kolhapur, India)*

Abstract
Data mining has been a popular research area for more than a decade due to its vast spectrum of applications. The aim of privacy preserving data mining researchers is to develop data mining techniques that could be applied on databases without violating the privacy of individuals. In this work, we propose methods for constructing the dissimilarity matrix of objects from different sites in a privacy preserving manner which can be used for privacy preserving clustering that require pair-wise comparison of individual private data objects horizontally distributed to multiple sites.

Keyword: *Privacy, Data Mining, Distributed Clustering, Security.*

I. INTRODUCTION

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). Achieving privacy preservation when sharing data for clustering is a challenging problem. To address this problem, data owners must not only meet privacy requirements but also guarantee valid clustering results. Let us consider a real-life motivating example where the sharing of data for clustering poses different constraints:

Suppose that a hospital shares some data for research purposes (e.g. group patients who have a similar disease). The hospital's security administrator may suppress some identifiers (e.g. name, address, phone number, etc) from patient records to meet privacy requirements. However, the released data may not be fully protected. A patient record may contain other information that can be linked with other datasets to re-identify individuals or entities. How can we identify groups of patients with a similar disease without revealing the values of the attributes associated with them?

We refer to the former as privacy preserving clustering (PPC) over centralized data, and the latter as PPC over vertically partitioned data. The problem of PPC over vertically and horizontally partitioned data has been addressed in [4, 5].

Privacy preserving techniques for clustering over vertically partitioned data was proposed by Vaidya and Clifton in [3].

In this paper, we propose a privacy preserving clustering technique on horizontally partitioned data. Our method is based on constructing the dissimilarity matrix of objects from different sites in a privacy preserving manner which can be used for privacy preserving clustering.

In Section 2, we provide the background and related work. Section 3 gives the basic concepts and problem formulation. Section 4 provides steps for protocol implementation that will be later used clustering.

II. BACKGROUND AND RELATED WORK

The aim of privacy preserving data mining is ensuring individual's privacy while maintaining the efficiency of data mining techniques. Mainly two approaches are employed: 1. data sanitization and 2. Secure multi-party computation. Data mining on sanitized data results in loss of accuracy, while secure multi-party computation protocols give accurate results at the expense of high computation or communication cost.

Agarwal & Srikant initiated research on privacy preserving data mining for constructing classification models while preserving privacy in [1]. Clifton and Vaidya propose a secure multi-party computation of k-means algorithm on vertically partitioned data in [3]. In Ref. [4] and Ref. [5], Oliveira and Zaïane focus on different transformation techniques that enable the data owner to share the mining data with another party who will cluster it. In Ref. [8], they propose new methods for clustering centralized data: dimensionality reduction and object similarity based representation. Methods in Ref. [8] are also applicable on vertically partitioned data, in which case each partition is transformed by its owner and joined by one of the involved parties who will construct a dissimilarity matrix to be input to hierarchical clustering algorithms. Ref. [6] proposes model-based solutions for the privacy preserving clustering problem.

Optimized Protocol for Privacy Preserving

The problem of privacy preserving clustering on distributed data by means of secure multi-party computation of the global dissimilarity matrix which can then be input to hierarchical clustering methods. There is no loss of accuracy while providing privacy to horizontal partitioned data and with the numeric attributes and categorical attributes can be used in clustering.

III. BASIC CONCEPTS AND PROBLEM DEFINITION

A. Data Matrix

A data matrix is an object-by-variable structure. Each row in the matrix represents an entity through values of its attributes stored in columns. An m×n data matrix is the data of m objects on n attributes as depicted in (1).

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

The attributes in a data matrix are sometimes normalized before being used. The main reason is that different attributes may be measured on different scales (e.g. centimeters and kilograms). For this reason, it is common to normalize the data so that all attributes are on the same scale.

B. Dissimilarity Matrix

A dissimilarity matrix is an object-by-object structure. An m×m dissimilarity matrix stores the distance or dissimilarity between each pair of objects as depicted in (1). Intuitively, the distance of an object to itself is 0. A dissimilarity matrix, d (i,j), is as shown in (2).

$$d = \begin{bmatrix} 0 & & & & \\ dist(2,1) & 0 & & & \\ dist(3,1) & dist(3,2) & 0 & & \\ \vdots & \vdots & \vdots & 0 & \\ dist(m,1) & dist(m,2) & \cdots & \cdots & 0 \end{bmatrix} \tag{2}$$

To calculate the dissimilarity between objects i and j one could use the most popular distance measure called Euclidean distance, or others. If i = (xi1, xi2, ..., xin) and j = (xj1, xj2, ..., xjn) are n-dimensional data objects, the Euclidean distance between i and j is given in (3).

$$d(i,j) = [\sum_{k=1}^{n} (x_{ik} - x_{jk})^2]^{1/2} \tag{3}$$

The attributes in a data matrix, D, in (1) are sometimes normalized before being used. The main reason is that different attributes may be measured on different scales (e.g. centimeters and kilograms). For this reason, it is common to standardize the data so that all attributes are on the same scale. We review Min-Max Normalization. Min-max normalization performs a linear transformation on the original data. Each attribute is normalized by scaling its values so that they fall within a small specific range, such as 0.0 and 1.0. Min-max normalization maps a value v of an attribute A to v' as follows:

$$v' = \frac{v - min_A}{max_A - min_A} \times (new\_max_A - new\_min_A) + new\_min_A \tag{4}$$

where min_A and max_A represent the minimum and maximum values of an attribute A, respectively, while new_ min_A and new_ max_A are the new range in which the normalized data will fall.

C. Problem Definition

We will approach the problem of PPC on horizontally partitioned data. There are k data holders, such that $k \geq 2$, each of which owns a horizontal partition of the data matrix D, denoted as D_k. These parties want to cluster their data by means of a third party so that the clustering results will be public to data holders at the end of the protocol. The third party, denoted as TP, does not have any data but serves as a means of computation power and storage space.

At the end of the protocol, the third party will have constructed the dissimilarity matrices for each attribute separately. Then the third party runs a hierarchical clustering algorithm on the final dissimilarity matrix and publishes the results.

IV. STEPS IN PROTOCOL IMPLEMENTATION

There are three participants of the protocol: **TP**- Third Party (Semi Honest Nature), $\mathbf{DH_j}$-Data Holder at Site 1, $\mathbf{DH_k}$-Data Holder at Site 2 as shown in Fig. 1.

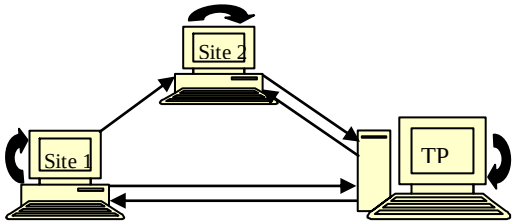


Fig. 1. Overall steps in protocol implementation

Third party's duty in the protocol is to govern the communication between data holders, construct the dissimilarity matrix and publish clustering results to data holders. Data holders are supposed to have agreed on the list of attributes that are going to be used for clustering beforehand. This attribute list is also shared with the third party so that TP can run appropriate comparison functions for different data types.

At the end of the protocol, the third party will have constructed the dissimilarity matrices for each attribute separately. These dissimilarity matrices are weighted using a weight vector sent by data holders such that the final dissimilarity matrix is built. Then the third party runs a hierarchical clustering algorithm on the final dissimilarity matrix and publishes the results.

Major steps for implementation of privacy preserving clustering as follows:

Step 1 - Suppress identifiers:

Attributes that are not subjected to clustering (e.g., address, phone, etc) are suppressed.

Step 2 -Normalize numerical attributes:

If the attributes subjected to clustering are numerical, they should be normalized using (4). Normalization helps prevent attributes with large ranges (e.g, salary) from outweighing attributes with smaller ranges (e.g., age). If the dataset contains mixed variables, there is no need for normalization.

Step 3 - Compute the Local dissimilarity matrix & send it to TP:

In the step, the pair wise distances between objects are computed. Euclidean distance is widely used for numerical attributes as given in (3).

Dissimilarity matrix, d, is an object-by-object structure in which d[i][j] is the distance between objects i and j. Now consider an entry in d, d[i][j]: If both objects are held by the same data holder, the third party need not intervene in the computation of the distance between objects i and j. The data holder that has these objects simply computes the distance between them and sends the result to the third party. However, if objects i and j are from different parties, a privacy

preserving comparison protocol must be employed between owners of these objects. Every data holder first builds his local dissimilarity matrix using the algorithm in Fig. 2 for numeric attribute and sends the result to the third party.

INPUT: Comparison function distance(), Data vector D_J

OUTPUT: Local dissimilarity matrix d

Begin

1. Initialize matrix of distance values:
   d = {D_J.Length × D_J.Length}

For m = 0 to D_J.Length-1

For n = 0 to m

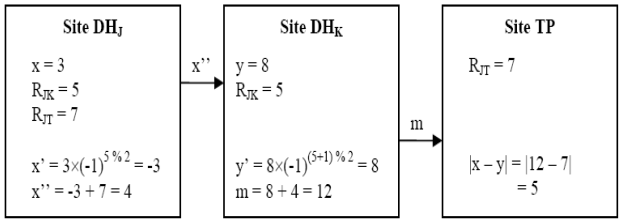
2. d[m] [n] = distance(D_J[m], D_J[n])

End

Fig. 2. Pseudocode for local dissimilarity matrix construction for numeric attributes

Step 4 – Encrypt Data Vector & send to Next site :

In this step, Site DH_J processes each value in DH_J (data vector at site j) using both rngJK and rngJT and sends the resulting vector to site DH_K as shown in Fig. 3. DH_J and DH_K share a secret number r_{JK} that will be used as the seed of a pseudo-random number generator, rngJK. Similarly DH_J and TP share a secret number r_{JT} that will be used as the seed of a pseudo-random number generator, rngJT.



Step 5 – Compute Comparison Matrix:

In this step, Site DH_K initializes a matrix of size DH_K.Length × DH_J.Length to be sent to site TP. This matrix is filled column by column, each column containing intermediary comparison values for an input from DH_J against all inputs from DH_K. At the end of each column, DH_K creates a new random number using rngJK to preserve consistency with the oddness/evenness of the random numbers generated at site DH_J.

The distance between two numeric attributes is simply the absolute value of the difference between them. Categorical attributes are only compared for equality so that any categorical value is equally distant to all other values but itself.

Pseudocode for the computing comparison matrix of integers is given in Fig. 4, 5 and 6.

INPUT: Pseudo-random number generator seeds r_{JK} and r_{JT}, Input data vector DH_J
OUTPUT: Data vector DH'_J
Begin
1. Initialize rngJK with seed r_{JK}
2. Initialize rngJT with seed r_{JT}
3. Initialize data vector: $DH'_J = \{DH_J.Length\}$
For m = 0 **to** DH_J.Length-1
4. Blind_JK=rngJK.Next()
5. Blind_JT= rngJT.Next()
6. $DH'_J[m] = Blind\_JT + DH_J[m] \times (-1)^{Blind\_JK\%2}$
7. Send DH'_J to site DH_K
End

Fig. 4. Pseudocode of numeric attributes comparison at site DH_J

INPUT: Pseudo-random number generator seed r_{JK}, Input data vectors DH_K and DH'_J
OUTPUT: Pair-wise comparison matrix s
Begin
1. Initialize rngJK with seed r_{JK}
2. Initialize comparison matrix:
s = {DH_K.Length × DH'_J.Length}
For n = 0 **to** DH'_J.Length-1
3. Blind_JK=rngJK.Next()
For m = 0 **to** DH_K.Length-1
4. $s[m][n] = DH'_J[n] + DH_K[m] \times (-1)^{(Blind\_JK + 1)\%2}$
5. Send s to site TP
End

Fig. 5. Pseudocode of numeric attributes comparison at site DH_K

INPUT: Pseudo-random number generator seed r_{JT}, Pair-wise comparison matrix s
OUTPUT: Dissimilarity matrix component for DH_J and DH_K, J_K
Begin
1. Initialize rngJT with seed r_{JT}
2. Initialize matrix of distance values:
J_K = {s.Length × s[0].Length}
For n = 0 **to** s[0].Length-1
3. Blind_JT= rngJT.Next()
For m = 0 **to** s.Length-1
4. J_K[m] [n] = | s[m] [n] – Blind_JT |
End

Fig. 6. Pseudocode of numeric attributes comparison at site TP

Step 6 – Transfer Comparison Matrix To TP :

In this step, Site DH_K sends intermediate comparison for each attribute to TP. Site TP, upon receiving this matrix, subtracts the random numbers generated by rngJT as shown in Fig. 2.

Step 7 – Calculate Global Dissimilarity Matrix :

Each party that holds a horizontal partition of a data matrix D can construct its local dissimilarity matrix as long as comparison functions for object attributes are public. However privacy preserving comparison protocols need to be employed in order to calculate an entry d[i][j] of dissimilarity matrix d, in(2),if objects i and j are not held by the same party.

Involved parties construct separate dissimilarity matrices for each attribute in our protocol. Then these matrices are merged into a single matrix using a weight function on the attributes. The dissimilarity matrix for attribute i is denoted as d_i. Finally the third party merges all portions of the dissimilarity matrix from local matrices and comparison protocol outputs. The process ends after a normalization step that scales distance values into [0, 1] interval. Details of the algorithm are given in Fig. 7.

INPUT: List of data holders DH
OUTPUT: Dissimilarity matrix d
Begin
For each Data holder DH_J ×DH
1. Request local dissimilarity matrix
For each Data holder DH_K ×DH and K > J
2. Run comparison protocol between DH_J and DH_K
3. Construct dissimilarity matrix d
4. Normalize d (d[m] [n] = d[m] [n] / maximum value in d)
End

Fig. 7. Pseudocode for Global dissimilarity matrix construction of numeric attributes

Upon the construction of dissimilarity matrices for all attributes, then dissimilarity matrices of attributes are merged into a final dissimilarity matrix, which will be input to the appropriate clustering algorithm.

Step 8 – Apply Clustering Algorithms:

Third party notifies the data holders, asking for their attribute weight vectors and hierarchical clustering algorithm choices. The main advantage of our method is its generality in applicability to different clustering methods such as hierarchical clustering.

Step 9 – Convey Results Back to Different Sites:

Dissimilarity matrices must be kept secret by the third party because data holder parties can use distance scores to infer private information since they have a portion of data. That's why clustering results are published as a list of objects of each cluster. The results of clustering are conveyed to data owners without any leakage of private information.

V. CONCLUSION

In this paper we proposed algorithms for privacy preserving clustering over horizontal partitioned data. Our method is based on the dissimilarity matrix construction using a secure comparison protocol for numerical data. The main advantage of our method is its generality in applicability to different clustering methods such as hierarchical clustering. Another major contribution is that quality of the resultant clusters can easily be measured and conveyed to data owners without any leakage of private information.

REFERENCES

[1] R. Agrawal and R. Srikant, "Privacy-Preserving Data Mining," *Proc. 19th ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, 2000, pp. 439-450.

[2] J. Han and M. Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann, 2001.

[3] J. Vaidya and C. Clifton. "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, August 2003.

[4] S. R. M. Oliveira, O. R. Zaïane, "Achieving Privacy Preservation When Sharing Data for Clustering," *Proceedings of the International Workshop on Secure Data Management in a Connected World* (2004) 67-82

[5] S. R. M. Oliveira, O. R. Zaïane, "Privacy Preserving Clustering By Data Transformation," *Proceedings of the 18th Brazilian Symposium on Databases* (2003) 304-318

[6].S. Meregu and J. Ghosh. "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, November 2003.

[7].J. Vaidya, C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2002) 639-644

[8].C. Clifton et al., "Tools for Privacy Preserving Distributed Data Mining," *SIGKDD Explorations*, vol. 4, no. 2, 2003, pp. 28-34.

[9].S. R. M. Oliveira, O. R. Zaïane, "Privacy Preserving Clustering By Object Similarity-Based Representation and Dimensionality Reduction Transformation," *Proceedings of the 2004 ICDM Workshop on Privacy and Security Aspects of Data Mining (*2004) 40-46