ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2018.0252

PRIVACY PRESERVING DATA MINING USING THRESHOLD BASED FUZZY C-MEANS CLUSTERING

V. Manikandan, V. Porkodi, Amin Salih Mohammed and M. Sivaram
Department of Information Technology, Lebanese French University, Erbil, Iraqi Kurdistan

Abstract

Privacy preserving is critical in the field of where data mining is transformed into cooperative task among individuals. In data mining, clustering algorithms are most skilled and frequently used frameworks. In this paper, we propose a privacy-preserving threshold clustering that uses code based technique with threshold estimation for sharing of secret data in privacy-preserving mechanism. The process includes code based methodology which enables the information to be partitioned into numerous shares and handled independently at various servers. The proposed method takes less number of iterations in comparison with existing methods that does not require any trust among the clients or servers. The paper additionally provides experimental results on security and computational efficiency of proposed method.

Keywords:

Privacy Preserving, Data Mining, Threshold Cryptography, Fuzzy C-means Clustering, Vandermonde Matrix, Secure Multiparty Computation

1. INTRODUCTION

Advances in networking and data acquisition strategies have empowered the collection and capacity of huge storage of data. This data is of no utilization until the point when it is analysed and changed over into valuable data. The data is first clustered and afterward examined to discover patterns. To get more exact data patterns, organizations share their data, which can trade off the security of clients and their data. The development of numerous techniques guarantee data security and protection of data. A few cryptographic systems are, for example, homomorphic encryption, secure computation and threshold cryptographic strategies. As an answer for the protection issues in distributed and privacy-preserving data mining is seen in [1]. Privacy-preservation distributed data mining is the agreeable calculation of data that is disseminated between numerous different parties without uncovering any of their private data items.

The security of data is reasonably characterized as the proper utilization of data. Anchoring sensitive data is generally known as data security and as a rule alluded to as the accessibility, secrecy and respectability of data. Data security ensures that the data is right, trustworthy and open when those with allowed get to require it. Organizations need to underwrite an approach of data security for the single motivation behind guarantying data protection or the security of their customer data, especially when it is being used. One technique for ensuring the security of the individual records is to bother the first data. Data annoyance methods are measurably based systems that attempt to guarantee mystery data by adding arbitrary clamor to private, numerical characteristics, in this way protecting the first data.

1.1 SECURITY PRESERVING DATA MINING

Consider a situation in which in excess of two parties having sensitive data mean to forms a computation on the blend of their contributions without revealing any unfortunate data. In the perfect situation every member sends their contributions to the ordered party, who next procedures the limit and sends the correct outcomes to interchange party without losing security of individual data sources. Along these lines we can save security even within the sight of ill-disposed members that endeavor to accumulate data about the contributions of their parties. After [1] proposition on idea of secure calculation in the field of data mining, from that point forward, privacy-preservation distributed data mining has pulled in much consideration and many secure conventions have been proposed for particular data mining calculations.

2. RELATED WORK

Security saving data mining was proposed in [1]. The study of the blend of cryptography strategies and security saving data mining techniques might be found in [6]. Different methods like randomization for grouping utilizing privacy-preservation has been examined. Cryptography based procedures [7] offer protection at larger amount yet at the expense of high calculation and correspondence required in such cases. The rundown of the association among the fields of cryptography and PPDM might be found in [9]. In [12], displayed exhaustive and similar investigation of mystery sharing strategies for PPDM and Secure Multiparty Computation based systems and its proficiency.

In [20], the author proposes a privacy clustering based sharing technique for vertically allocated data using two non-plotting clients to compute group implies. In [13], the authors proposed a privacy-preservation estimation in data mining. The procedures of secure multiparty computation (SMC), homomorphic encryption (HE) and comparison has been helpful for techniques proposed in [10] [11].

2.1 OUR CONTRIBUTION

The past methodologies utilized Chinese Remainder Theorem for the data distribution and Secure Multiparty Computation, shamir threshold mystery sharing plan for privacy preserving data mining. In our development, we utilize the code based method that incorporates with Vandermondes grid for figuring offers of private data. Intelligent conventions are then intended to do the bunching calculation. In this work, we accomplish the security at the level of mystery sharing while at the same time keeping the correspondence expenses to a level like that of the past conventions and however we accomplish an effective bunching. Our methodology is more important in lessening computational expense as looked at in this paper. The benefit of our convention is that, it fundamentally diminishes the correspondence costs in this manner making privacy-preservation bunching functional.

3. PRELIMINARIES

3.1 CLUSTERING

The initial phase in data mining is grouping the data as indicated by some rule. Sharing finds the regular collection of unlabeled data. This dividing prompts the arrangement of significant sub-parties or sub-classes that are called clusters. Cluster investigation is a general undertaking to be settled and not one particular calculation. Cluster investigation calculations can be extensively delegated progressive and partition calculations. Dividing calculations group data as indicated by some standard and assess them. The work is centered on Fuzzy c-means clustering which is a partition calculation.

3.2 FUZZY C-MEANS ALGORITHM

The fuzzy c-means (FCM) algorithm is a grouping algorithm. It is helpful when the required number of clusters are pre-decided; along these lines, the algorithm attempts to put every one of the information focuses to one of the clusters. What makes FCM distinctive is that it does not choose the total participation of an information point to a given cluster; rather, it computes the probability that an information point will have a place with that cluster. Henceforth, contingent upon the exactness of the grouping that is required practically speaking, suitable resistance measures can be set up. Since the total enrollment has not ascertained, FCM can be quick due to number of cycles required to accomplish a particular clustering exercise compares to the required exactness.

3.2.1 Iterations:

In every cycle of the FCM calculation, the objective function J is minimized:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{C} \delta_{ij} \left\| x_i - c_i \right\|^2 \tag{1}$$

where, N is the number of data points, C is the number of clusters required, c_j is the centre vector for cluster j, and δ_{ij} is the degree of membership for the i^{th} data point x_i in cluster j.

The standard, $\|x_i - c_j\|$ measures the closeness of the data point x_i toward the middle vector c_j of cluster j. Note that, in every iteration, the calculation keeps up a middle vector for every one of the clusters. These data-points are computed as the weighted average of the data-points, where the weights are given by the degrees of membership.

3.2.2 Degree of Membership:

For a given data point x_i, the membership degree to cluster j is calculated as:

$$\delta_{ij} = \frac{1}{\sum_{k=1}^{C} \delta_{ij} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \tag{2}$$

where, m is the fuzziness coefficient and the centre vector c_j is calculated as:

$$c_j = \frac{\sum_{i=1}^{N} \delta_{ij}^{m} x_i}{\sum_{i=1}^{N} \delta_{ij}^{m}} \tag{3}$$

From above, δ_{ij} is the value of the membership degree from previous iteration. Note that, the membership degree for data point i to cluster j is initialised at the start of iteration with a random value θ_{ij}, $0 \le \theta_{ij} \le 1$, such that $\sum_{j=1}^{C} \delta_{ij} = 1$.

3.2.3 Fuzziness Coefficient:

In Eq.(2) and Eq.(3) the fuzziness coefficient m, where $1 < m < \infty$, measures the tolerance of the required clustering. This value determines how much the clusters can overlap with one another. The higher the value of m, the larger the overlap between clusters. In other words, the higher the fuzziness coefficient the algorithm uses, a larger number of data points will fall inside a fuzzy band where the degree of membership is neither 0 nor 1, but somewhere in between.

3.2.4 Termination Condition:

The required accuracy of the degree of membership determines the number of iterations completed by the FCM algorithm. This measure of accuracy is calculated using the degree of membership from one iteration to the next, taking the largest of these values across all data points considering all of the clusters. If we represent the measure of accuracy between iteration k and k+1 with ϵ, we calculate its value as follows:

$$\varepsilon = \Delta_i^N \Delta_j^C \left| \delta_{ij}^{k+1} - \delta_{ij}^{k} \right| \tag{4}$$

where, δ_{ij}^{k} and δ_{ij}^{k+1} are respectively the degree of membership at iteration k and k+1, and the operator Δ, when supplied a vector of values, returns the largest value in that vector.

3.3 THRESHOLD CRYPTOGRAPHY

Threshold cryptography is a secret cryptographic technique, which is used for appropriating a secret key among individuals in a way that an approved subset of individuals can especially reproduce the secret key and an unapproved subset can get no data about the secret. It is a specific methodology used as a piece of secure multiparty hashing, where diverse questioned individuals facilitate and lead count endeavors considering the private data they give. Secret sharing was at first proposed in [3]. The arrangement by Shamir relies upon the standard Lagrange polynomial expansion, while the arrangement by Blakley relies upon the geometric imagined that uses meeting hyper planes. Afterward, in [4] threshold sharing plans is proposed using Chinese leftover portion hypothesis. In [5] proposed secret sharing plans in view of direct and MDS codes.

3.4 DEFINITIONS AND GOALS

A threshold cryptosystem or signature scheme actualized by n players with threshold t is said to be secure if the perspective of the enemy that debases up to t players does not empower him to compute decodings or signatures alone. A threshold scheme is said to be vigorous if, regardless of what the corrupted t players do, the remaining (i.e. legitimate) players still yield a substantial decryption or signature.

A standard technique of demonstrating security of a threshold cryptosystem (or a signature scheme) is to display a reenactment calculation which, without access to any mystery data however with a prophet access to the single-server acknowledgment of the hidden cryptosystem outfits the attacker with the right perspective of the execution of threshold protocol. In this way, by displaying such test system, we lessen the security of the threshold version of a cryptosystem to the security of its single-server partner.

A relating standard technique for demonstrating robustness of an threshold scheme is to display an information extractor which fills the role of the legitimate players in the protocol, and on the off chance that the attacker prevails with regards to inciting the fair players into delivering an invalid yield, it separates from the attacker's conduct an answer for some difficult issue. Hence once more, by displaying such extractor, we lessen the strength of our threshold protocol to some standard hardness assumption.

3.4.1 Concurrent Adaptive Security with Committed Proofs:

Simultaneous versatile security with submitted proofs. Our first perception about the above thinking is that there may be no conflicting player amid the recreation by any stretch of the imagination, if the “trading off” share a_i can be deleted before the halfway outcome A_i is distributed. Since there would be no conflicting players, the test system could never need to rewind, and subsequently simultaneous executions of such threshold protocol can be recreated and along these lines demonstrated secure. Be that as it may, how might we accomplish power if a player is to eradicate its offer a_i before distributing A_i. We demonstrate that it is to be sure conceivable by concocting a novel apparatus of a submitted zero-information verification, where an explanation that should be demonstrated, e.g. A_i and $g^{a_i}h^{\bar{a}_i}$ contain a similar esteem a_i, is uncovered simply after the evidence closes. Specifically, it very well may be uncovered after the observer a_i expected to demonstrate the above explanation is eradicated. This submitted evidence method would thus be able to be connected to change, with immaterial increment in correspondence many-sided quality, the versatile DSS and RSA arrangements, and also different protocols like threshold ElGamal, to simultaneously anchor versatile arrangements. We further observe that by providing robustness while eliminating all inconsistent players in the above way, the committed proof technique can actually transform, in the erasure-enabled setting, a very general class of statically secure threshold protocols into adaptively and concurrently secure ones.

We additionally see that by giving strength while disposing of every conflicting player in the above way, the submitted confirmation procedure can really change, in the deletion empowered setting, an exceptionally broad class of statically secure threshold protocols into adaptively and simultaneously secure ones.

4. PROPOSED DISTRIBUTED FUZZY C-MEANS CLUSTERING ALGORITHM

In this section, we present our distributed privacy preservation using Fuzzy c-means Clustering protocol.

4.1 PROPOSED ALGORITHM

The proposed distributed threshold Fuzzy c-means algorithm is divided into three sub phases namely Initialization, Data storage and Fuzzy c-means clustering.

Phase-1: Initially, choose each entity in the data base X_1 which is one dimensional and choose l × m vandermonde matrix V from finite field of F_q. Shares X_1 l are calculated using original data D and V. Distribute shares X_1 l_j to j servers.

Phase-2: In this subsection we present the sub phases of Fuzzy c-means clustering algorithm. The first step of the algorithm is, the servers chooses k points in the R dimensional space as their initial cluster points. Next the servers create k cluster locations (centers) to the servers with the computation for the current iteration.

- **Step 1:** *Initialization*
- **Step 2.1:** *Cluster Assignment*: To assign data the clusters we calculate minimum distance using Euclidean distance and then apply secure addition protocol and secure comparison protocol techniques are used. The algorithm is as follows:
- **Step 2.2:** Updating cluster locations after having $D^l_{k_j}$ which is the distance, the each server interacts with the cluster centers to find the mean M_{k_j}. Finally assign the entity as new cluster center of cluster k which is close to the computed mean.

Phase-3:

- **Step 2.3:** *Checking Termination Criterion*: If no change in the cluster centers, then stop, else repeat Step 2.

Phase-4:

- **Step 3:** *Knowledge Revelation*: Cluster assignment is revealed to all the data owners.

5. EXPERIMENTAL RESULTS AND ANALYSIS

We have implemented our algorithm in C language. The experiments are conducted on Linux, Ubuntu 14.02 LTS version, Intel Core 2 Duo CPU with 8GB RAM and 2.93GHz speed. In this results we found that our protocol terminates means limited number of iterations and provide efficient clustering with respect to distributed Fuzzy c-means clustering algorithm even with or without privacy preserving mechanism.

5.1 SECURITY ANALYSIS

The total computation cost of the clustering is depends on the initial clusters and the number of iterations required for finding final clusters.

5.2 PRIVACY

Theorem 5.1: The privacy of the secret data can be achieved stated earlier is fulfilled.

Proof: As we have seen, the chosen codeword C, can be reconstructed by specifying any of its N components. In $[n,k,d]$ MDS code, message symbols are of any of k symbols are taken. Even out of n, if (k–1) servers are compromised even though secret cannot be reconstructed. This way we can achieve the privacy preserving of the data. Less than k symbols or an unauthorized set recovering probability of the secret is equal to same as that of the exhaustive search, which is 1-q.

Theorem 5.2: The Proposed PPDM protocol is efficient and ideal.

Proof: Initially, we distribute the secret data to each servers is given exactly one share. Also, the chosen secret data sets and the generated shares space is F_q. Shares are distributed uniquely and randomly to the servers efficiently. So, the proposed algorithm is ideal and efficient.

5.3 COMPUTATIONAL EFFICIENCY

Experimental comparisons showed that the performance of PPMF is superior to the algorithm in [2]. Use Java language to realize two algorithms. The experimental environment is: two computers whose configuration settings are 2.19GHz dual-core CPU, 2GB memory and 1.61GHz CPU for AMD, memory for 512MB. One of them is mining server and another only is parties. Use plant Cell Signaling data [5] as the experimental data sets. We took the 2000 row. Average arranges them in the two computers according to items. Since point protocols of the two algorithms have random numbers, it will influence the algorithm time. We carry out several experiments for each fixed support for every algorithm, and then take the average value. The Fig.3 shows the result with 2000 row. X-axis is the support and the y-axis is operation time which unit is millisecond (ms).

From the figures, we can see that the operating time of PPMF is superior to the algorithm of literature [2]. The smaller the support degree, PPMF algorithm produces higher efficiency. This is about the local calculation time, network transmission time and the complexity of scalar product, so we observe this question from these three points. Local computing time is mainly referring to generate local frequent itemsets and part of infrequent itemsets time. In PPMF, this time is about find all the maximal frequent itemsets time used. For [2], it is the time of Apriori algorithm. For the complexity of scalar product: [2] needs four steps to get the results and need to three transmission between two parties. The point product protocol of this paper needs three steps and needs to two transmission data among the two parties.

Table.1. Computational Efficiency Calculations

Support	**Time (ms)**	
	Proposed Privacy Preservation with Fuzzy c-means clustering	**Privacy Preservation with K-means clustering**
0.1	0.2	0.45
0.2	0.18	0.4
0.3	0.15	0.36
0.4	0.13	0.3
0.5	0.11	0.24
0.6	0.09	0.15

6. CONCLUSIONS

In this work, we propose a distributed threshold privacy preserving Fuzzy c-means clustering algorithm that use the code based threshold secret sharing scheme and secure addition and comparison protocols. We allow parties to collaboratively perform clustering and thus avoiding trusted third party. We compare our protocol with k-means based clustering proposed. Our algorithm does not require any trust among the servers or users and it provide perfect privacy preserving of user data.

REFERENCES

[1] R. Agrawal and R. Srikant, “Privacy Preserving Data Mining. ACM SIGMOD”, *Proceedings of International Conference on Management of Data*, pp. 439-450, 2000.

[2] Y. Lindell and Pinkas, “Privacy Preserving Data Mining”, *Journal of Cryptology*, Vol. 15, No. 3, pp. 177-183, 2002.

[3] A. Shamir, “*How to Share a Secret*”, Communications of the ACM, 1979.

[4] M. Mignotte. “How to Share a Secret”, *Proceedings of Workshop on Cryptography*, pp. 371-375, 1983.

[5] Josef Pieprzyk and Xian-Mo Zhang, “Ideal Threshold Schemes from MDS Codes”, *Proceedings of International Conference on Information Security and Cryptology*, pp. 253-263, 2003.

[6] B. Pinkas, “Cryptographic Techniques for Privacy-Preserving Data Mining”, Available at: http://www.pinkas.net/PAPERS/sigkdd.pdf.

[7] S. Verykios et al., “State of the-Art in Privacy Preserving Data Mining”, *ACM SIGMOD Record*, Vol. 33, No. 1, pp. 50-57, 2004.

[8] V Baby and Subhash N Chandra, “Privacy-Preserving Distributed Data Mining Techniques: A Survey”, *International Journal of Computer Applications*, Vol. 143, No. 10, pp. 37-41, 2016.

[9] J. Brickell and V. Shmatikov, “Privacy-Preserving Classifier Learning”, *Proceedings of 13th International Conference on Financial Cryptography and Data Security*, pp. 1-6, 2009.

[10] G. Jagannathan and R.N. Wright, “Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data”, *Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 593-599, 2005.

[11] P. Bunn and R. Ostrovsky, “Secure Two-Party K-Means Clustering”, *Proceedings of ACM International Conference on Computer and Communications Security*, pp. 486-497, 2007.

[12] M. Upmanyu, A.M. Namboodiri, K. Srinathan and C.V. Jawahar, “Efficient Privacy Preserving K-Means Clustering”, *Proceedings of Pacific-Asia Workshop on Intelligence and Security Informatics*, pp. 154-166, 2010.

[13] E. Bertino, I.N. Fovino and L.P. Provenza. “A Framework for Evaluating Privacy Preserving Data Mining Algorithms”, *Data Mining and Knowledge Discovery*, Vol. 11, No. 2, pp. 121-154, 2005.