Clustering of Data in Distributed Data Stores using Secure Multiparty Computation

M. Arathi[1] Ch. Nancy Paulina[2]

[1]Associate Professor, Department of Computer Networks & Information Security, School of Information Technology -JNTUH, Village Kukatpally, Mandal Hyderabad, District Hyderabad, Telangana, India

[2]M.Tech Student, Computer Networks & Information Security, School of Information Technology JNTUH, Village Kukatpally, Mandal Hyderabad, District Hyderabad, Telangana, India

Abstract

With advancements in the technologies used for data acquisition via sensors, online applications, it resulted in enormous amount of data. This data endures in different business places or organizations which results in a distributed environment and is then clustered to obtain desired results. In order to create clusters one must share their data, during this process the organization loses the privacy of data, for which many organizations adopt cryptographic method in order to protect its privacy, where data is first encrypted and then again decrypted while clustering. This results in an overall performance decrease as the cryptographic algorithms are applied to huge data. Hence, as a solution to this we have introduced a methodology which doesn't affect the data but provides privacy for the group of people who are involved in clustering of their data. "Secure Multiparty Computation" is used to provide privacy in this system and "Bisecting K-means" for clustering of the data.

Keywords: **Bisecting K-Means Clustering, Secure Multiparty Computation.**

1. Introduction

Over a couple of years, the data being generated through sensors and online applications has grown in its size, also has shifted from centralized processing to distributed processing. The data generated must be processed to extract used information from it. Processing of such huge data cannot be done as a single unit which can lead to an undesirable result and so the data is divided into "data sets". This "data sets" can be distributed and then processed. Different data mining techniques are then used to process the data. Clustering is one useful technique among these to process the data and get some useful information. Clustering involves partitioning the datasets based on the commonness or similarity between the objects or the data. For instance, to know the average salary of people according to their age, data must be collected from all age levels and then records with similar ages are grouped and clustered. Another practical scenario may consist of different telephone companies that need to set up towers in a common region they acquired. The optimum location of these common towers can be found through clustering algorithms so as to maximize the signal strength for each network's users. In social network analysis, clustering users' publications may help in recognizing communities with dense friendships internally and sparse friendships externally. Social network clustering can help in designing marketing plans, identifying terrorist cells and other useful applications. Hence, putting data sets got from different social networks together should deliver more value to the analysis. Further, the effective integration of data mining has given rise to privacy issues surrounding disclosing personal private data during analysis process in distributed environments. In fact, personal opinions, political interests, healthcare records and other private data are being shared between institutions and service providers to improve accuracy of the clustering task.

2. Literature Survey

In this section a literature survey is done on the Secure Multiparty Computation techniques and also about the K-means Clustering and its privacy issues.

2.1.Secure Multiparty Computation (SMC)

In secure multiparty computation (SMC), data is distributed among distinct yet connected parties. All parties cooperate to safely compute the final end result without revealing their private records to any party.SMC can be used as a cryptographic alternate to a trusted third party (TTP) as TTP is expensive and is hard to find one. The idea of secure two-party computation was initiated by Yao[2].It was later extended to fit into multiple parties.

Privacy and correctness are the vital necessities of any secure computation. Privacy holds when parties know only their output nothing outside what is undeniably required. When each party obtains its correct output, correctness holds good. The secure multiparty computations uses cryptographic or randomization method. In randomization, noise is added to the data. This prevents identifying the real data.

Few important techniques in secure multiparty computations:

1. Homomorphic Schemes: The homomorphic property allows operating on cipher values and obtains cipher value as results. In privacy preserving clustering technique, homomorphic public key cryptosystem is used to securely compute the distances matrix and cluster centers.

2. Secret Sharing: Secret sharing is another very popular homomorphic scheme used in clustering distributed datasets. Here the secret of one party is distributed amongst other parties in such a way that recovering the secret by one party all alone is not possible

3. Circuit Evaluation: The concept by Yao [2] is the fundamental work for evaluation circuit. Here a scrambled Boolean circuit consisting of encryption values is used for function evaluation. The input is divided between the parties. The main advantage of this circuit is that the parties cannot learn anything apart from the result. As each bit requires encryption, this method is expensive. This approach is widely used in k- means clustering algorithm to securely find the distance.

2.2.K- Means Clustering

The standard k-means algorithm is a repetitive refinement approach that minimizes the sum of squared distances between each point and its assigned cluster center. Assume that m points are to clustered into k clusters, all the m points have to go through assignment step, where they are assigned to the nearest cluster. The assignment step costs O(mk). For applications with large mk, the assignment step in exact k-means becomes prohibitively expensive. Hence, various approaches have been proposed for approximate k-means in large-scale applications.

2.3. Privacy in K-means Clustering

Companies in different sector prefer to outsource their data to multiple systems as huge amount of data is collected. While outsourcing the data to different systems, they must ensure privacy and security of the data. Dataset consists of sensitive which should not be disclosed to the third party. On one hand data owner encrypts the data before launching it to the distributed environment. On the other hand, several methods or techniques are used to securely query the data in the cloud storage by preserving the privacy of the dataset. With the rapid growth of data in its volume clustering methods are implemented that adheres to these features. For example to predict the investment of person requires clustering over his financial records, his savings etc. that contains sensitive information. In the multiparty scheme, cryptographic primitives are used that are expensive as it includes homomorphism encryption and transfer of data.

2.4. Privacy issues in K means Clustering

The integration of huge data mining has given rise to privacy issues surrounding disclosing personal private data during analysis process in distributed environments. In fact, personal opinions, political interests, healthcare records and other private data are being shared between institutions and service providers to improve accuracy of the clustering task.

When k-means algorithm is executed on a dataset, the distance computation itself does not violate privacy because each party holds all the components of an entity. But the problem arises when computing intermediate cluster centers, in this case, the entities of same cluster may come from several parties. This step also requires knowledge of the number of entities in each cluster; this number is extra information that should not be revealed to different parties during the execution of the protocol.

Another privacy issue arises in data distribution, while the random selection of k first centers while clustering.

2.5. Algorithms to obtain privacy while clustering

Many algorithms were proposed to overcome the issue with the privacy of the datasets in the process of clustering.

Jha [3] have proposed two protocols for the preservation of privacy in k-means algorithm on two parties only. Security primitives are used for centers computation in order to preserve the privacy of entities of each party. The first protocol called OPE is based on oblivious polynomial evaluation given by Naor and Pinkas[4]. The second protocol named DPE is based on homomorphic encryption schemes. The homomorphic encryption scheme is more efficient that

OPE for the two parameters: computing and communication cost, but their solution cannot be extended to several parties.

Samet[5] proposed a protocol which uses a secure method of division that protects the entities of each party and prevents the revelation of the number of entities in each cluster. The protocol is also applicable in a multi-party environment, but the intermediate centers are always revealed.

3. Proposed System

In this paper, the main focus is to provide, a secure way for the different participants involved in clustering. The data of the participant is only disclosed after the participant is verified as a group member. This is determined by using a key(smcKey) generated using Secure Multiparty Computation technique. Before determining the key, each participant has to choose a value (private key) known only to him. These private keys are then used to compute the smcKey, which is then distributed to all the participants. When a participant wants to cluster his data, he has to provide the smcKey; else he will not be allowed to do clustering. Once the key is verified, Bisecting K-means algorithm is used to create clusters. Using this algorithm the similarity between the points can be more compared to that while clustered using K-means. In the 4thth section of this paper the comparisons are made.

ALGORITHMS USED:

We have used two main algorithms. One based on Secure Multiparty computation for Key generation and Bisecting K-means algorithm for clustering of the data. The later algorithm includes the K-means algorithm.

Key Generation using Secure Multiparty Computation:

1. Let p represent the private value of each participant.

2. SmcKey= R+ Σp_i

where i= 1 to n

n= number of participants.

p_i= Private key of the ith participant

R= a random number

Every time a new participant joins the group the SmcKey is regenerated. A registered participant is given the SmcKey only after he enters his secret value (Private Key) correctly.

Since the key is computed using all the private values, it is hard for a participant to determine the key. A participant cannot find the private value of other participants as the key is regenerated every time a new participant enters or an existing person leaves.

The random value used to calculate the key makes it difficult for an attacker to determine the private values of the participants, as a different random value is used every time the key is computed.

Clustering Algorithm:

Bisecting K-means Clustering:

An implementation of the Bisecting K-means algorithm.
The Bisecting K-Means algorithm is a variation of the regular K-Means algorithms.
It consists of the following steps:

1. First pick a cluster,
2. Find 2-subclusters using the basic K-Means algorithm(This is the bisecting step)
3. Repeat the above step, for a fixed number of times(determined previously) and take the split that produces the clustering.
4. Repeat the first three steps until the desired number of clusters is reached.

In this implementation, the Squared Sum of Errors (SSE) is calculated to determine if a split is good. Moreover, we always choose the largest cluster for splitting.

Squared Sum of Errors:

SSE is the sum of the squared differences between each observation and its group's mean. It is used as a measure of variation within a cluster. The lesser the value the more the objects in the clusters are similarity.

K-means Clustering:

An implementation of the K-means algorithm

1. Choose the required number of clusters, k.
2. Generate k clusters randomly and determine the centers of the clusters, or generate random points, k, naming centers.
3. Allocate points to the cluster containing, cluster center near to the point.
4. Recalculate the new centers.
5. Repeat steps 3,4 until the allocation hasn't changed.

Two Distance functions are used for calculation in the Clustering algorithm.

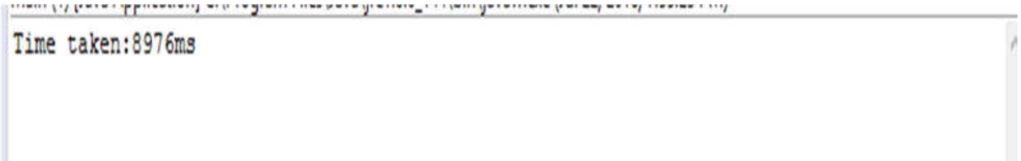
Euclidean: $d= \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$

Cosine: $d= \dfrac{\Sigma (x_i y_i)}{\sqrt{\Sigma (x_i)^2}\sqrt{\Sigma (y_i)^2}}$

4. Experimental Results

The implementation was done using java and Eclipse Oxygen was used as an IDE. The time period is divided into two. One from the time of login till the clustering begins, the other is the time taken for clustering of the data.

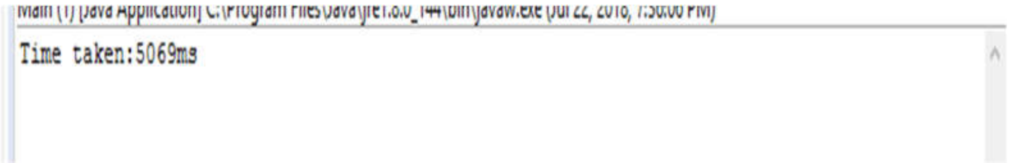
The time taken for the participant to login and get the key was around 8.976s (as shown in fig below)



```
Time taken:8976ms
```

Figure 1.a. Time taken for know the key

But if the participant already knows the key then the time taken to login and reach clustering tab was 5.069s (as shown in the Figure 1.b).



```
Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (Jul 22, 2018, 7:50:00 PM)
Time taken:5069ms
```

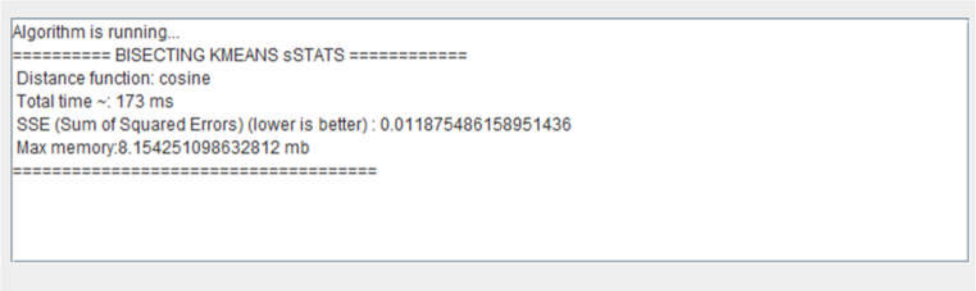
Figure 1.b. Time taken for complete login

The later time periods are of two clustering algorithms Bisecting K-means and K-means with combinations of two distance funtions Euclidean and cosine. As a total 4 time periods were observed.

A file consisting of 296 rows and 2 columns of numerical values was used as a sample.

Using Bisecting k-means(Cosine)

Time taken:173ms

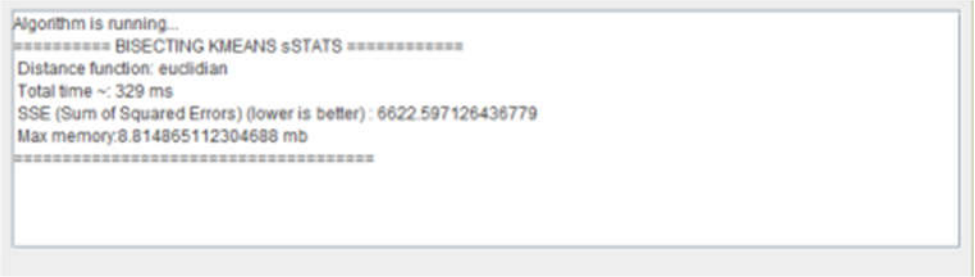


```
Algorithm is running...
========== BISECTING KMEANS sSTATS ============
 Distance function: cosine
 Total time ~: 173 ms
 SSE (Sum of Squared Errors) (lower is better) : 0.011875486158951436
 Max memory:8.154251098632812 mb
=======================================
```

Figure 2.Bisecting k-means using cosine

Using Bisecting k-means (Euclidean)

Time taken: 329ms

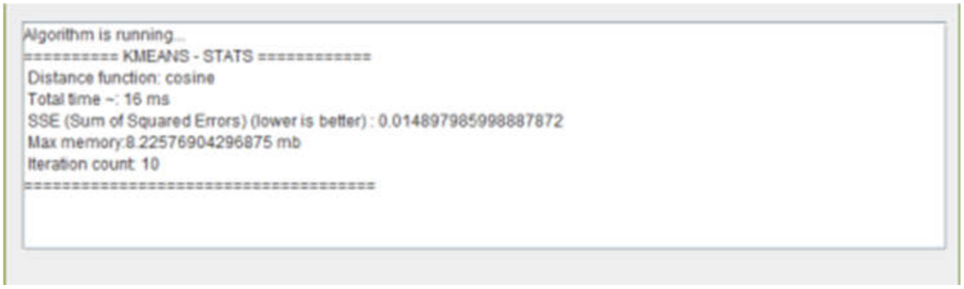


```
Algorithm is running...
========== BISECTING KMEANS sSTATS ============
 Distance function: euclidian
 Total time ~: 329 ms
 SSE (Sum of Squared Errors) (lower is better) : 6622.597126436779
 Max memory:8.814865112304688 mb
=======================================
```

Figure 3.Bisecting k-means using Euclidean

Using K-means (Cosine)

Time taken: 16ms

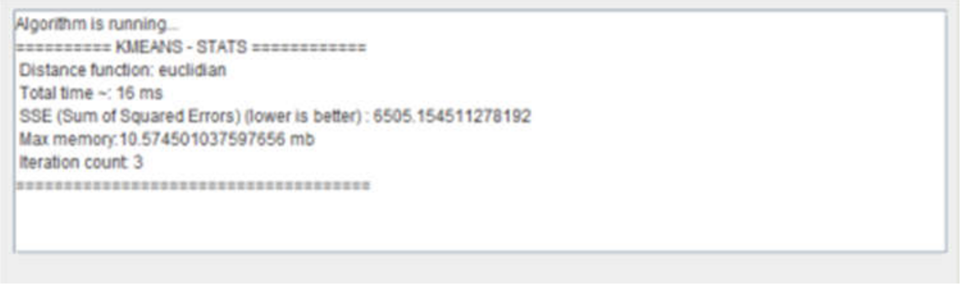


```
Algorithm is running...
========== KMEANS - STATS ============
 Distance function: cosine
 Total time ~: 16 ms
 SSE (Sum of Squared Errors) (lower is better) : 0.014897985998887872
 Max memory:8.22576904296875 mb
 Iteration count: 10
=======================================
```

Figure 4.K-means using cosine

Using K-means (Euclidean)

Time taken: 16ms



```
Algorithm is running...
========== KMEANS - STATS ============
 Distance function: euclidian
 Total time ~: 16 ms
 SSE (Sum of Squared Errors) (lower is better) : 6505.154511278192
 Max memory:10.574501037597656 mb
 Iteration count: 3
=====================================
```

Figure 5. K-means using Euclidean

All the observed time periods are then tabularized (Table 1.) and compared.

Table 1. Time taken for clustering

Algorithm	Distance Function	Time taken(ms)	Sum of Summed errors(SSE)
Bisecting k-means	Cosine	173	0.0118
Bisecting k-means	Euclidean	329	6622.597
K-means	Cosine	16	0.0149
K-means	Euclidean	16	6505.15

From the above table Bisecting k-means with Cosine as distance funtion has minimum round time.

5. Conclusion

This paper shows how data residing with different participants can be clustered without sharing their information. For providing security a key is used which doesn't deal with the data at all. Although data with different variety cannot be clustered using this method, still can be used to perform clustering most of the numerical data. Only the people who know the key can be allowed to perform clustering. An attacker cannot compute the key even if he knows the previous key, as it is computed everything a new participant enters or leaves the group. The bisecting k-means algorithm also results in a quick clustering when compared to the traditional k-means algorithm. Hence this paper provides a result of clustering the data without encryption and still preserving the privacy.

References

[1]Zakaria Gheid, Yacine Challal “Efficient and privacy preserving k-means clustering For Big Data Mining.

[2].Yao, Andrew C. "Protocols for secure computations." Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on. IEEE, 1982.

[3]Jha S.,Kruger L., and Mc-Daniel P., “Privacy- Preserving Clustering,” in Proceedings of European Symposium on Research in Computer Security, pp 397-417,2005.

[4] Naor M. and Pinkas B.,” Oblivious Transfer and Polynomial Evaluation,” in Proceedings of the 31st ACM Symposium on Theory of Computaing, Atlanta USA,pp.245-254,1999.

[5]Samet S., Miri A., and Orozco-Barbosa L., “Privacy – Preserving K-Means Clustering in Multi-Party environment” in Proceedings of International Conference on Security and Cryptography, Barcelona, Spain pp. 523-531, 2007.