Poster: Privacy Preserving Collaborative Clustering with Hyper Parameter Recommendation

Maryam Ghasemian
Case Western Reserve University
maryam.ghasemian@case.edu

Erman Ayday
Case Western Reserve University
erman.ayday@case.edu

***Abstract*—Clustering is an unsupervised machine learning technique that creates groups of data that are similar. The collected data may contain sensitive or confidential information and privacy concerns may arise when the data is shared among multiple parties. In this paper, we investigated various clustering algorithms in the context of a collaborative data mining model, in which two data owners performed the clustering algorithm on their shared data. We attempted to address the issue of determining optimal clustering algorithm input parameters when two parties want to perform collaborative clustering on their combined data set with the assistance of a third party (server).**

I. Introduction

Similar data records are assigned to the same cluster, and the proximity measure heavily influences (dis)similarity. A clustering algorithm can work independently or in conjunction with another algorithm [1]: In the collaborative framework, the goal is that each local computation, quite possibly applied to distinct data sets, benefits from the work done by the other "collaborators". This can be accomplished by exchanging information about local data, current hypothesized local clustering, or algorithm parameters. In this paper, we specially focused on collaborative clustering with/without the help of a server. The goal is to find whether or not the input parameters provided by the server lead us to the best clustering results. To address this, we performed several experiments on different scenarios to calculate the input parameters and analyzed the results using clustering measures such as the Adjusted Rand index (ARI) and Silhouette Score. We selected one labeled data set with numeric data type for our experiments.

Four different types of clustering algorithms have been investigated in this work [2], Partitioning-based clustering, distribution-based clustering, density-based clustering and hierarchical clustering. In this work we focused on one of the most well-known clustering algorithms in each of the aforementioned clustering categories.

II. System and Threat Models

The main idea for collaborative clustering is when two or more data owners want to cluster their shared data set. They may outsource some of their data (with additional noise) to a server (AKA semi trusted party). The server will then assist them in selecting the best clustering algorithm based on the data they provide. It will also provide the hyper parameters to parties (data owners) in order for them to perform the clustering (i.e., the number of clusters for k-means, HC, and GMM clusterings, and the maximum distance between clusters in DBSCAN (*Eps*)). There are several techniques to find the optimum number of the clusters (k). In this work, we considered Elbow method and Silhouette Score method to find the optimum k as the primary input parameter for K-Means, HC and GMM algorithms. DBSCAN has two main input parameter,*Eps* and *minpoint*. The *Eps* value is computed based on *KNN*(k-Nearest Neighbors) algorithm. *minpoint* another input parameter for DBSCAN, can be obtained with [3] and [4] recommendations based on different data dimension. In our system model, the server was assumed to be semi-honest. As a result, the server may misbehave and attempt to extract sensitive information from each party's (data owner's) data set using metadata, workflow, or algorithm output. Membership inference, deanonymization attacks, and attribute inference are all known privacy attacks that use study results and/or partially provided data sets [5]. In membership inference attacks, an attacker may attempt to determine whether or not a desired record (victim) is part of the data set. In attribute inference attacks, the attacker's goal is to infer additional sensitive attributes of an individual from the observed ones. Because the identities of the individuals are concealed in our work, attribute inference becomes a viable attack scenario only after the attacker determines a victim's data set membership. The attacker's goal in deanonymization attacks is to use auxiliary information about the individual to link an individual's anonymized data to the individual's identity. Because of the metadata's shared partial data set, this may be possible. However, after applying noise to the each party data set, each party shares only a small portion of with the server. As a result, the most effective deanonymization attack can use the victim's partial data as auxiliary data, making deanonymization more complicated than membership inference attacks. Therefore, the most relevant attack for a misbehaving server in our scenario is membership inference. And since we assume that the parties trust each other in our proposed system model, there is no threat model from the data owners' perspective.

III. Proposed Framework

Our proposed system model and framework as shown in Figure 1 consist of five steps as follows: **Step 1:** Data owners make data set noisy using randomized response (RR). **Step 2:** Data owners send a portion of their noisy data to server. **Step 3:** Server applies some methods to find out the optimum

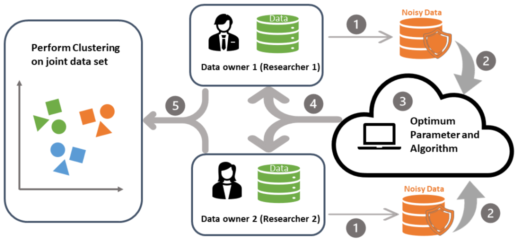

Perform Clustering on joint data set
Data
Data owner 1 (Researcher 1)
Data
Data owner 2 (Researcher 2)
1
Noisy Data
2
3
Optimum Parameter and Algorithm
4
5
Noisy Data
1
2

Fig. 1. Proposed system framework

Algorithm	Server Size	ϵ	K/ Eps	ARI	Silhouette
K-Means	0.1	0.01	7	1	0.44
K-Means	0.1	0.1	8	0.75	0.41
K-Means	0.1	1	8	0.75	0.41
K-Means	0.1	5	7	1	0.44
K-Means	0.1	0.1	8	0.75	0.41
K-Means	0.2	0.1	9	0.79	0.40
K-Means	0.3	0.1	8	0.75	0.41
K-Means	0.4	0.1	8	0.75	0.41

TABLE I
EFFECT OF PRIVACY BUDGET (ϵ) AND THE AMOUNT OF DATA SHARED WITH THE SERVER ON ARI AND SILHOUETTE VALUES IN THE K-MEANS CLUSTERING ALGORITHM WHEN PARTIES SHARE 10% OF THEIR NOISY DATA WITH THE SERVER OR WHEN THE PRIVACY BUDGET IS 10%. K/EPS: SERVER SUGGESTION OF THE HYPER PARAMETER (K: NUMBER OF THE CLUSTERS, EPS: MAXIMUM DISTANCE)

algorithm with its corresponding hyper parameter(s). **Step 4:** Server provides its outcome (algorithm and parameter) to the data owners. **Step 5:** Data owners perform clustering based on server suggestions on their combined data set.

IV. EXPERIMENTS AND RESULTS

The main input parameter for K-Means, HC and GMM clustering algorithms is the number of clusters (number of the components in GMM algorithm), k. In the DBSCAN algorithm, two main input parameters include the lowest cluster size (*minpoint*) and the maximum distance between the two clusters (*Eps*).

After applying Steps 1-4 of the III, effects of the privacy budget (ϵ), as well as the amount of data shared with the server, were analysed via running experiments and applying clustering algorithms based on the server suggestions of the input parameters. In addition, the power of the membership inference attack was obtained through experiments.

Effect of privacy budget and amount of the shared data: The effect of the privacy parameter and amount of shared data on the input parameters of clustering algorithms is discussed in this Section. The main question here is how the data perturbed with different amount of ϵ or different shared portion of data can lead the server to guess the input parameter of the specific clustering algorithm. To accomplish this, the same amount of data was shared with the server while changing the privacy parameter to perturb data (RR mechanism) to observe the effect of the ϵ. To observe the effect of the amount of shared data, the ϵ remained the same and different portions of the data shared with the server. In both approaches the server suggestion was used to evaluate the clustering results for the not noisy joint data set. Table I shows the a sample of the results of applying K-Means algorithm on joint data set when 10% of the data shared with the server to observe the effect of ϵ and when different portions of the data shared with the server while the ϵ was 0.1. In both approached it was observed that either ϵ or the portion of the shared data has little to no effect on the server suggestion of the input parameter.

*Membership Inference Attack:*A (misbehaving) server might try to determine whether a target victim is in the case group by computing the distance between the target victim's data points and the partial noisy data set of the case group's individuals which known as "hamming distance" attack. We used LRT (Likelihood Ratio Test) to quantify the membership inference risk due to shared data set and the hamming distance attack to quantify the membership inference risk due to the shared partial noisy data set. In this work, case group consists of the 150 individuals from one data owner that shared with the server and the control group consist of the data from the second data owner plus the data from first data owner that is not shared with the server. The results showed that the power of the membership inference attack increases for higher ϵs.

V. CONCLUSION

In this work, we obtained optimum input parameters for the four well-known clustering algorithms while two data owners want to perform the clustering collaboratively and a semi-trusted third party suggests them the optimum input parameter and the clustering algorithm with high utility. Results indicated that neither the amount of perturbed data shared with the third party (server), nor the privacy budget (ϵ), has any significant effect on the server suggestion. In addition, analysis of the membership inference attack showed that the power of the membership inference attack increases when ϵ increases.

ACKNOWLEDGEMENT

The work was partly supported by the National Library of Medicine of the National Institutes of Health under Award Number R01LM013429, the National Science Foundation (NSF) under grant numbers 2141622, 2050410, 2200255, and OAC-2112606, and Cisco Research.

REFERENCES

[1] A. Cornuéjols, C. Wemmert, P. Gancarski, and Y. Bennani, "Collaborative clustering: Why, when, what and how," *Information Fusion*, vol. 39, 04 2017.

[2] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[3] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, pp. 226–231, 1996.

[5] C. Dwork, A. Smith, T. Steinke, and J. Ullman, "Exposed! a survey of attacks on private data," *Annu. Rev. Stat. Appl*, vol. 4, no. 1, pp. 61–84, 2017.

Poster: Privacy Preserving Collaborative Clustering with Hyper Parameter Recommendation

Maryam Ghasemian, Erman Ayday
(maryam.ghasemian, erman.ayday) @case.edu

Introduction

What is clustering and collaborative clustering?

- Clustering is an unsupervised machine learning (ML) technique for detecting unknown patterns in unlabeled data.
- Clustering is made up of four parts: feature selection/normalization, a proximity measure to determine similarity/dissimilarity, a clustering algorithm, and an output evaluation.
- A clustering algorithm may operate independently or in collaboration with another algorithm (cooperative clustering, collaborative) [1]
- Types of clustering algorithms: partitioning-based, distribution-based , density-based and hierarchical [2]
- In this work, we have focused on one clustering algorithm in each clustering categories:
  - K-Means, Hierarchical Clustering (HC), Gaussian Mixture Models (GMM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN).
- We specially focused on collaborative clustering with the help of a server.

Our Goal?

- To find whether or not the input parameters provided by the server lead us to the best clustering results.
- Performed several experiments to calculate the input parameters and analyzed the results using clustering measures such as the Adjusted Rand index (ARI) and Silhouette Score.

System Model

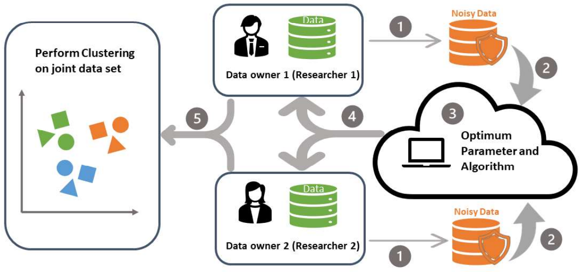

Perform Clustering on joint data set
Data
Data owner 1 (Researcher 1)
Noisy Data
Data
Data owner 2 (Researcher 2)
Noisy Data
Optimum Parameter and Algorithm
1
2
3
4
5
1
2

Fig. 1: Proposed System Model

- **Step 1:** Data owners make data set noisy using randomized response (RR).
- **Step 2:** Data owners send a portion of their noisy data to server.
- **Step 3:** Server apply some methods to find out the optimum algorithm with its corresponding hyper parameter(s).
- **Step 4:** Server provides its outcome (algorithm and parameter) to the data owners.
- **Step 5:** In this step Data owners will perform clustering based on server suggestions on their combined data set (this step is already done in some previous approaches by using some encryption techniques in distributed/collaborative clustering)

Methods to choose optimum input parameter

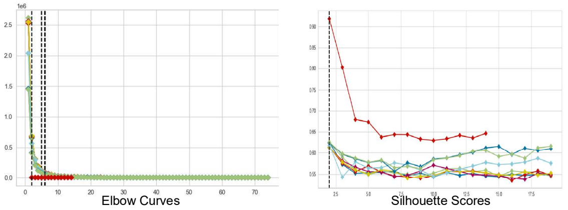


Fig. 2: Elbow vs Silhouette methods to choose optimum input parameter

Results and Evaluation

Optimum input parameter selection results on noisy data:

- Using the generalized RR mechanism, each party perturbed their own data.
- Data owners send a subset of their perturbed data to a semi-trusted server .
- The server attempts to identify the best input parameter for each clustering algorithm.[3,4]
- The server also, must determine which algorithm will provide data owners with the best clustering results.
- The server has its own selection mechanism.

Algorithm	Server Size	ϵ	K/ Eps	Silhouette Coef	CH
GMM	10%	0.1	8/-	0.34	301.30
DBSCAN	10%	0.1	10/1	-	-
K-Means	**10%**	**0.1**	**8/-**	**0.36**	**318.13**
HC	10%	0.1	8/-	0.31	237.61

Table 1. Server suggestions of the clustering input parameter for different clustering algorithms when parties shared 10% of their noisy data (when ϵ= 0.1) with it.

Algorithm	K/Eps	ARI	Homo	Comp	Silhouette	CH
GMM	8/-	0.21	0.34	0.36	-0.17	429.0
DBSCAN	10/1	0.02	0.12	0.39	-0.50	32.85
K-Means	**8/-**	**0.75**	**0.87**	**0.81**	**0.41**	**4768.90**
HC	8/-	0.48	0.58	0.77	0.39	2240.70

Table 2. clustering performance results on the combined not noisy data set using input parameters suggested by server from table 1.

Privacy Analysis:

- Effect of the privacy metric:
  - Same portion of the data shared with the server

Algorithm	Server Size	ϵ	K/ Eps	ARI	Silhouette
K-Means	0.1	0.01	7	1	0.44
K-Means	0.1	0.1	8	0.75	0.41
K-Means	0.1	1	8	0.75	0.41
K-Means	0.1	5	7	1	0.44
HC	0.1	0.01	8	0.481	0.39
HC	0.1	0.1	8	0.481	0.39
HC	0.1	1	7	0.482	0.41
HC	0.1	5	8	0.482	0.41
GMM	0.1	0.01	7	0.189	-0.035
GMM	0.1	0.1	6	0.185	-0.0143
GMM	0.1	1	8	0.2069	-0.072
GMM	0.1	5	6	0.2008	-0.007
DBSCAN	0.1	0.01	10/1	0.017	-0.504
DBSCAN	0.1	0.1	10/1	0.017	-0.504
DBSCAN	0.1	1	10/1	0.017	-0.504
DBSCAN	0.1	5	10/1	0.017	-0.504

Table 3. Effect of privacy budget (ϵ) for different clustering algorithms when parties shared 10% of their noisy data with the server.

- Effect of the shared data portion:
  - Data perturbed with same value of the privacy parameter

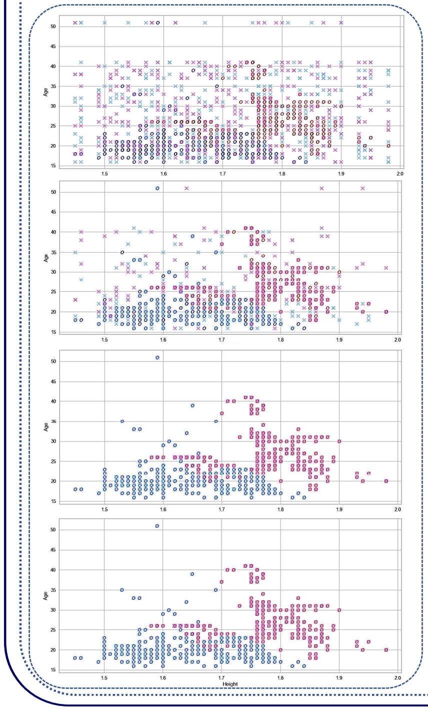
Algorithm	Server Size	ϵ	K/ Eps	ARI	Silhouette
K-Means	0.1	0.1	8	0.75	0.41
K-Means	0.2	0.1	9	0.79	0.40
K-Means	0.3	0.1	8	0.75	0.41
K-Means	0.4	0.1	8	0.75	0.41
HC	0.1	0.1	8	0.481	0.39
HC	0.2	0.1	8	0.481	0.39
HC	0.3	0.1	8	0.481	0.39
HC	0.4	0.1	8	0.0.481	0.39
GMM	0.1	0.1	6	0.185	-0.143
GMM	0.2	0.1	5	0.163	0.029
GMM	0.3	0.1	8	0.175	-0.111
GMM	0.4	0.1	5	0.169	-0.001
DBSCAN	0.1	0.1	10/1	0.017	-0.504
DBSCAN	0.2	0.1	10/1	0.017	-0.504
DBSCAN	0.3	0.1	10/1	0.017	-0.504
DBSCAN	0.4	0.1	10/1	0.017	-0.504

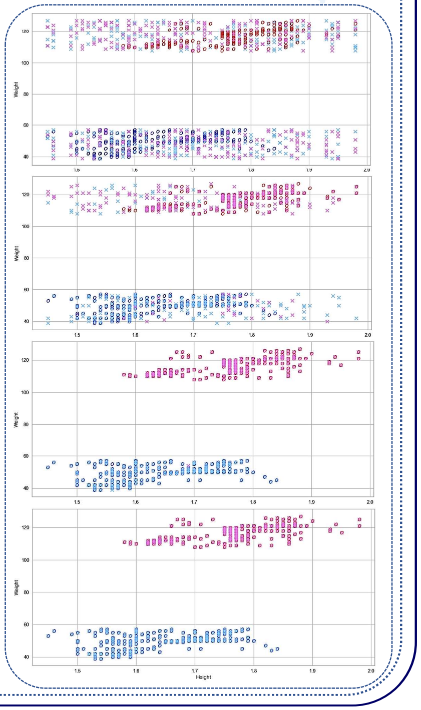
Table 4. Effect of amount of the data which shared with the server for different clustering algorithms when the privacy parameter (ϵ) = 0.1.

Why this happens?

- The RR mechanism preserves the gap between the clusters if there is any.

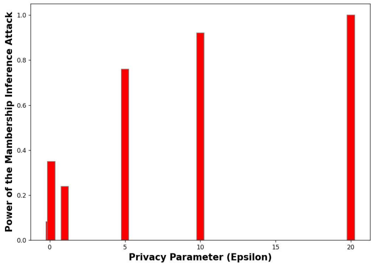
ϵ = {1, 5, 10, 20}
O = Original data points
X = Noisy data points





Membership Inference Attack

- A (misbehaving) server might try to determine whether a target victim is in the case group [5].
- By ”hamming distance” attack : Computing the distance between the target victim’s data points and the partial noisy data set of the case group’s individuals.
- LRT(Likelihood Ratio Test) was used to quantify the membership inference risk due to shared data set .
- The hamming distance attack was used to quantify the membership inference risk due to the shared partial noisy data set.
- Case group consists of the 150 individuals from one data owner that shared with the server.
- The results showed that the power of the membership inference attack increases for higher ϵs.



Conclusion

- Optimum input parameters for the four well-known clustering algorithms were identified while two data owners want to perform the clustering collaboratively.
- A semi-trusted third party suggested data owners the optimum input parameter and the clustering algorithm with high utility.
- The amount of perturbed data shared with the third party (server) and the privacy budget (ϵ) have no significant effect on the server suggestion.
- Analysis of membership inference attack has indicated that the power of the membership inference attack increases when ϵ increases.

References

[1] A. Cornu´ejols, C. Wemmert, P. Gancarski, and Y. Bennani, “Collaborative clustering: Why, when, what and how,” Information Fusion, vol. 39, 04 2017.
[2] R. Xu and D. Wunsch, “Survey of clustering algorithms,” IEEE Transactions
on Neural Networks, vol. 16, no. 3, pp. 645–678, 2005.
[3] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, “Density-based clustering in spatial databases: The algorithm gdbscan and its applications,” Data mining and knowledge discovery, vol. 2, no. 2, pp. 169–194, 1998.
[4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in kdd, vol. 96, pp. 226–231, 1996.
[5] C. Dwork, A. Smith, T. Steinke, and J. Ullman, “Exposed! a survey of attacks on private data,” Annu. Rev. Stat. Appl, vol. 4, no. 1, pp. 61–84, 2017.

Acknowledgement
The work was partly supported by the National Library of Medicine of the National Institutes of Health under Award Number R01LM013429, the National Science Foundation (NSF) under grant numbers 2141622, 2050410, 2200255, and OAC-2112606, and Cisco Research.