Density Peaks Clustering with Differential Privacy

ABSTRACT

Density peaks clustering (DPC) is a latest and well-known density-based clustering algorithm which offers advantages for finding clusters of arbitrary shapes compared to others algorithm. However, the attacker can deduce sensitive points from the known point when the cluster centers and sizes are exactly released in the cluster analysis. To the best of our knowledge, this is the first time that privacy protection has been applied to DPC. In this paper, we provide density peaks clustering privacy protection(DPCP) model to obtain the clustering results without revealing the data via differential privacy protection, in which the privacy protection is achieved by add Laplace noise to local density ρ and distance δ. However, the computation complexity will reaches O(n) and have an inaccurate clustering results when adding noise to the data set directly. Therefore, we are inspired by the idea of divide and conquer algorithm. Firstly, we divide the data set into relatively independent groups by Voronoi diagram and then adding noises. We employ a parallel computing by MapReduce to improve the efficiency. Secondly, according to the principle that is the privacy budget can be superimposed in high dimensional data. We introduces $\epsilon_1+\epsilon_2$-differential privacy protection model and ensure the accuracy of the calculation via data replication and filter. Where ϵ_1 and ϵ_2 to protect ρ and δ respectively. Finally, through a lot of experiments, we also provide performance analysis and privacy proof of our solution.

CCS Concepts

•**Security and privacy** → *Domain-specific security and privacy architectures;*

Keywords

differential privacy; Voronoi diagram partition; clustering; privacy preserving; data mining

1. INTRODUCTION

WOODSTOCK '97 El Paso, Texas USA

© 2016 ACM. ISBN 123-4567-24-567/08/06…$15.00

DOI: 10.475/123_4

The cluster analysis is of great significance, which can obtain a great amount of valuable information. However, cluster analysis has its underlying risk, because some privacy information may be exposed in public [12]. For example, two organizations cluster their datasets with different attributes for a person to maximize portfolio income. The clustering results will be a person's complete information which reveals privacy. How to extract the knowledge from clustering data without revealing data each other is an important issue in the field of privacy preserving distributed data mining.

To this end, privacy-preserving clustering (PPC) model is proposed in the early period. Between 2003 and 2008, most works are oriented to the k-means algorithm by applying secure multi-party computation model on different data distributions that include vertically data [17], horizontally data [11] and arbitrary partitioned data [9, 2]. Yao protocol [22] and homomorphic cryptosystems are adopted in the above.

Interestingly, the research of secure function evaluation protocols almost completely ignored the question of which functionalities preserve privacy[1]. Since the differential privacy [7] was proposed and accepted by the database field, the privacy requirements of various jobs have implemented the transformation from syntactic model to a more rigorous differential privacy model. Clustering under differential privacy data analysis [1, 16, 17, 23] become more important. There are three state of the interactive algorithms. The first is the differentially private version of the Lloyd algorithm. The second algorithm is implemented in the GUPT system. The third one is PrivGene. The experiments show that the first method is the best [21].

Typical partitioning-based clustering algorithms (the most common is k-means) are not able to detect non-spherical clusters. But the density-based clustering can be done. For the DBSCAN[8] that is the classical density-based clustering algorithm. There are several privacy-preserving algorithms. Such as, Kumar et al. [13] discussed both horizontally and vertically partitioned data. Jinfei et al. [14] oriented to horizontally, vertically and arbitrarily partitioned data and designed a Multiplication Protocol based on Pailler's Additive Homomorphic cryptosystem.

The big data era has been an enormous increase in the multi-dimensional data and divesification data. We need a simple and fast clustering algorithm which can be applied to data sets with various types and shapes. For the above problems, Alex Rodriguez and Alessandro Laio [19] propose an alternative approach. The algorithm has its basis in the assumptions that cluster centers are surrounded by neighbors with lower local density and that they are at a relatively large distance from any points with a higher local density. For each data point i, they compute two quantities: its local density ρ_i and its distance δ_i from points of higher density. To the best of our knowledge, this is the first time that privacy protection has been applied to this clustering process. In this paper, we study the density peaks clusters under differential privacy protection. For instance, it requires to measure distance between any point of objects when computing ρ and δ value for each data. Additional, if we directly add noise to the raw data. The model's efficiency and scalability will be limits especially for high-dimensional data.

Therefore, we are inspired by the idea of divide and conquer algorithm[15] and the principle that is the privacy budget can be superimposed in high dimensional data[3]. Our main contributions are summarized as follows:

1) We introduce the idea of Voronoi-diagram partitioning. The original data set is divided into relatively independent grouping. Meanwhile, in order to prevent errors in the calculation of ρ and δ, we use the idea of replication and filtering.

2) We introduce $\epsilon = \epsilon_1 + \epsilon_2$-differential privacy protection, in which ϵ_1 and ϵ_2 to protect ρ and δ, respectively. Because the clustering is determined by parameters ρ and δ, and these two parameters are all operated on the original data.

3) We conduct extensive experiments on three data sets with different dimensions and levels. The experimental results show that our algorithm is effective and accurate.

2. RELATED WORK

2.1 Differential Privacy

Differential privacy[6] is based on a very strict attack model, which guarantees that an adversary cannot infer an individual's presence in a dataset from the randomized output, despite having knowledge of all remaining individuals.

Definition 1. (ε-differential privacy)Given any pair of neighboring databases D and D' that differ only in one individual record, a randomized algorithm A is ε-differentially private iff for any S $\in$ Rang(A): $Pr[A(D) = S] \leq e^{z} * Pr[A(D') = S]$

The two neighboring data sets D and D' meet D=D'+t or D'=D+t, where D+t represents a tuple t add to the database. We use D $\cong$ D' to denotes this. Adding or deleting any tuple leads to a change in the output result will satisfy a certain probability distribution, which can protect any tuple. In previous work, there are a variety of methods used to realize ε-differential privacy, including the Laplace mechanism [7] and the exponential mechanism [18]. In this paper, we use Laplace Mechanism.

Laplace Mechanism: Laplace mechanism perform function g operations on data sets D, by adding a random noise g(D), the added size depends on the GS_g (overall sensitivity). Its random function A_g(D):

$$\begin{cases} A_g(D) = g(D) + Lap(\dfrac{GS_g}{\varepsilon}) \\ GS_g = max|g(D) - g(D')|, Pr[Lap(\beta) = x] = \dfrac{1}{2\beta} e^{\frac{-|x|}{\beta}} \end{cases}$$

Lap(β) represents a random variable selected from the Laplace distribution of the parameter β.

2.2 Density Peaks Clustering

We briefly review the DPC algorithm. Details are described in [19]. DPC is a density-based algorithm which can detect arbitrary shaped clusters. The key idea is that cluster centers are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. Therefore,cluster centers are decided by **local density** ρ and **density distance** σ. We illustrate several definitions and properties in DPC algorithm [19] that will be used in the next Section.

Definition 2. (**local density**) The local density of point x_i(denote as ρ_{x_i}) w.r.t. the number of points that are closer than d_c to point x_i. Let $\chi(x)$ is a function, $\chi(x)$=1 if x<0, otherwise $\chi(x)$=0 . Let $d_{x_i x_j}$ is a distance that from point x_j to point x_i. Let d_c is a cutoff distance. Thus $\rho_{x_i} = \sum_{x_j} \chi(d_{x_i x_j} - d_c)$.

Definition 3. (**density distance**) The density distance of point x_i(denote as δ_{x_i}) w.r.t. the typical nearest neighbor distance only for points that are local or global maxima in the density. Thus $\delta_{x_i} = \min_{x_j : \rho_{x_j} > \rho_{x_i}} (d_{x_i x_j})$. For the point with highest density, we conventionally take $\delta_{x_i} = max_j(d_{x_i x_j})$.

Definition 4. (**dependent point**) The dependent point of point x_i(denote as σ_{x_i}) w.r.t. the nearest point x_j from the point x_i in those point that local density more than the ρ_{x_i}. Thus $\sigma_{x_i} = \underset{x_j : \rho_{x_j} > \rho_{x_i}}{argmin}$ $(d_{x_i x_j})$.

Property 1. (**dependency**) According to Definition 4, for point x_i, it belongs to the cluster that include point σ_{x_i} if x_i depends on σ_{x_i}(denote as $\psi_{x_i \sigma_{x_i}}$ or $\psi_{x_i x_j}$). That is a dependency betweeen point x_i and σ_{x_i}.

Property 2. (**correlation between δ_{x_i} and σ_{x_i}**) According to Definition 3,4, for point x_i, the lower δ_{x_i} is, the closer σ_{x_i} is, and the stronger $\psi_{x_i \sigma_{x_i}}$ will be, vice versa.

Property 3. (**correlation between ρ_{x_i} and σ_{x_i}**) According to Definition 2-4, for point $x_i \in C_i$ (a cluster),it may be a dependent point σ_{x_j} for point x_j in cluster C_j if ρ_{x_i}>min$\rho(C_j)$=min$\{\rho_{x_j} | \forall x_j \in C_j\}$.

Definition 5. (**cluster**) Assuming that D is a data point set. A cluster C is a non-empty subset of D w.r.t. ρ_{x_i}, δ_{x_i}, σ_{x_i} and $\psi_{x_i x_j}$ satisfying the following conditions:

1. For point x_i, it is a cluster center, if and only if $\forall x_i$ $\in$ D, both ρ_{x_i} and δ_{x_i} are higher.
2. For point x_i, σ_{x_i} $\in$ C, x_i,x_j $\exists$ $\psi_{x_i x_j}$, then x_i $\in$ C.

3. DIFFERENTIAL PRIVACY PRESERVING DPC ALGORITHM

From the section2.2, we can know that the cluster centers and cluster sizes are determined by the ρ_i and δ_i. In order to prevent the leakage of privacy, we have to protect them in the clustering process. Here, we use the ϵ-differential privacy to protect them.

However, there are two problems when directly add Laplace noise in the process of calculating them. Firstly, the calculation of ρ_i and δ_i will be related to the distance between two points, which makes the algorithm high complexity. When dealing with massive high dimensional data, computing costs will be greater. Secondly, for differential privacy protection, the smaller the privacy budget is, the greater the added noise will be. So, the greater data set is, the greater the degree of deviation from the true value is, the worse the data availability will be.

To this end, we use Voronoi-based diagram partitioning to solve the above two problems. Firstly, the original data set is divided into M groups according to Voronoi diagram. Secondly, ρ_i and δ_i are calculated separately for each point in disjoint local groups. Grouping can be performed in parallel, such as using MapReduce and other strategies to avoid mass computational overhead. In addition, because the clustering is determined by the parameters ρ_i and δ_i, and they are both associated and need to be calculated separately, so we introduce two parameters ϵ_1 and ϵ_2 to protect ρ_i and δ_i respectively, and ϵ=ϵ_1+ϵ_2. Show as in *M* 1.

M 1. (**$\epsilon = \epsilon_1 + \epsilon_2$-differential privacy**)For each point $x_i \in C$, introduce two parameters ϵ_1 and ϵ_2 to protect ρ_{x_i} and δ_{x_i} respectively, such as $\rho'_{x_i} = \rho_{x_i} + Lap(\epsilon_1)$, $\delta''_{x_i} = \delta_{x_i} + Lap(\epsilon_2)$, in which $\epsilon = \epsilon_1 + \epsilon_2$, then $\gamma = \rho'_{x_i}\delta''_{x_i}$ satisfying $(\epsilon = \epsilon_1 + \epsilon_2)$-differential privacy, as show in Figure 1.

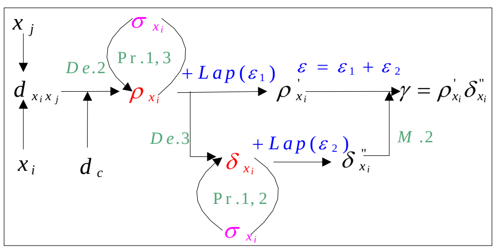


Figure 1: $(\epsilon = \epsilon_1 + \epsilon_2)$-differential privacy model

3.1 Pretreatment

In order to improve the efficiency and avoid reducing the data availability, we need to group data objects, and also need to determine an important parameter d$_c$. The parameter d$_c$ is calculated by empirical value estimation method[19], that is, the distance between all points is sorted in descending order, and select from 1%-2% of the sorted. The method of grouping is based on the Voronoi diagram. First,we illustrate symbols used in the follwing. D:data set, $\forall x \in D$, C: all clusters, S: initial center point set, s∈S, S_i: grouping of initial center point s_i, $C_i = S_i + R_i$, R_i: point set to be copied, l:boundary of Voronoi diagram. Next, we briefly introduce the Voronoi diagram, as show in Definition 6.

Definition 6. (**Voronoi diagram**) For dataset D, we select M points as the initial center point. The data set D is divided into M disjoint groups according to the vertical line dividing line between two points. Each point in the data set D is divided into a group, which is the shortest distance from the point to the initial center point.

Figure 2 shows an example of a Voronoi diagram partitioning the data set into 5 groups. Therefore, our preprocessing operation is grouping the data sets, by Voronoi diagram partitioning method, of course, first of all, we need to determine the initial center point. We pick the initial center point via the reservoir sampling algorithm [15] by MapReduce, and then calculate the distance between each data point x_i and the initial center point s_i. Comparing the distance between x_i and s_i, we choose s_i, so that $|x_i s_i|$ is a shortest distance.

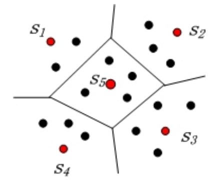


Figure 2: Voronoi diagram

After grouping, the whole data point set is divided into a number of disjoint groups. At the same time, we also use the reservoir sampling method to sample the distance between any two points, calculate the distance and sort them, and then select the appropriate d$_c$.

3.2 Calculation Local Density ρ

Now, firstly, we create a MapReduce job to group data. Grouping is based on Definition 4, wherein, dependent point σ_{x_i} from initial point set S, point x_i from the point to be processed. Secondly, we calculate ρ_{x_i} for each point x_i. Because each group is isolated with each other after grouping. So, for each point within the group, ρ_{x_i} may be a wrong value. As shown in Figure 3, in group S_j, the attribute ρ of the point x_j is 9, while the actual value is 13.

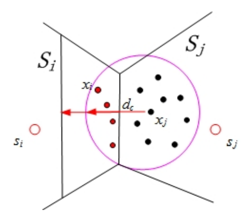


Figure 3: The process of data replication.

In order to get the correct ρ, we need to copy the four points from group S$_i$ to the S$_j$. Therefore, the point in each cluster C_i not only contains a set of points obtained from the Voronoi diagram, but also contains the R_{x_i} of all the points in this group. That is, C_i=$S_i \bigcup_{\forall x_i \in S_i} \mathrm{R}_{x_i}$,where in R_{x_i}={o|∀ o ∈ D,|o,x$_i$|<d$_c$}. As shown in Figure 3, all points that satisfy the distance to the edge of this group are less than dc will be copied to this group. After such grouping, data within each group contains two types, one is initial point set obtained by the Voronoi-diagram segmentation method, and another is the set of points that are replicated by other groups, in order to calculate the attribute values ρ of the data points.

Next, we add noise to points set, as show in equation(1).

$$\begin{cases} \rho' = \rho + Lap(\beta), \quad \rho = \sum_{x_j} \chi(d_{x_i x_j} - d_c) \\ Lap(\beta) = exp(-|x|/\beta), \beta = GS_g/\varepsilon_1 \end{cases} \tag{1}$$

The correctness of privacy protection for copy operation is ensured by Theorem 1.

THEOREM 1. *The condition of ρ' satisfying ϵ-differential privacy is that*

$$d_c > \frac{|x_i, s_j|^2 - |x_i, s_i|^2}{2|s_i, s_j|}, x_i, s_i \in S_i, s_j \in S_j.$$

Proofs in this section are deferred to the appendix.

3.3 Calculation Distance δ

Now, we calculate δ for each data point x_i. The calculation δ is limited because δ is also calculated within the group .Thus the calculated δ' is not true value, but slightly higher than the true value δ, as shown in Figure 4.

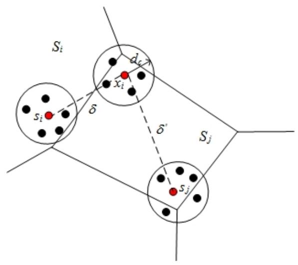


Figure 4: The relation between δ and δ'.

From Figure 5, for point x_i,we can see δ' significantly higher than δ. The reason for this error is that x_i and s_i are located in different group, while x_i and s_j are located in the same group. So x_i regard s_j as dependent point according to Definition 4. Therefore, we should take the distance between x_i and s_i as δ of x_i. Further more, there is $|x_i,s_i|\leq \delta'$ according to Property2 and Definition 3.

From the above analysis, it is known that x_i should be copied from S_j to S_i, copy conditions is $\rho_{s_i} > \min\{\rho_{x_i}|\forall x_i \in S_j\}$ according to Property 3. Obviously, the replicated group is $C_i = S_i \bigcup \sigma_{x_i}, \forall x_i \in S_i$. However, it will undoubtedly generate a lot of redundancy dependent point. To this end, according to Definition 2-4 and Property 2,3 we give the filter redudancy attachment model as show in M 2.

M 2. (**copy model**) Let δ^s denote the second max δ' in the grouping, and S_i, S_j denote the initial group. s_i, x_i, $x_m \in S_i$; $s_j, s_k, s_{x_i}, s_{x_m} \in S_j$; $S_i \neq S_j$, $x_i, s_{x_i} \exists \psi_{x_i,s_{x_i}}$, $x_m, s_{x_m} \exists \psi_{x_m,s_{x_m}}$ .Thus the copy condition of the dependent point follow in equation (2).

$$\begin{cases} \rho_{x_m} = max\rho(S_i), \quad \rho_{s_{x_m}} > \rho_{x_m} \\ |s_{x_m}, s_i| \leq \theta_2 = min\{2|x_m, s_i| + |s_j, s_k| + |s_j, s_i|\} \end{cases} \tag{2a}$$

$$\begin{cases} \rho_{x_i} \neq max\rho(S_i), \quad \rho_{s_{x_i}} > min\rho(S_i) \\ |s_{x_i}, s_i| \leq \theta_1 = max\{|x_i, s_i|\} + \delta^s(S_i) \end{cases} \tag{2b}$$

Now, we add noise to point set, as show in equation (3).

$$\begin{cases} \delta'' = \delta + Lap(\beta), \delta = \min\limits_{x_j:\rho_{x_j}>\rho_{x_i}} (d_{x_i x_j}) or \max\limits_{j}(d_{x_i x_j}) \\ Lap(\beta) = exp(-|x|/\beta), \beta = GS_g/\varepsilon_2 \end{cases} \tag{3}$$

The correctness of privacy protection for copy operation is ensured by Theorem 2.

THEOREM 2. *The condition of δ'' satisfying ϵ-differential privacy is that $|x_j,\ s_i|\leq|s_j, s_i|$-$\theta$, $\forall x_j, s_j \in S_j, s_i \in S_i$.*

Proofs in this section are deferred to the appendix.

4. EXPERIMENTAL ANALYSIS

In the experiments we used UCI dataset(http://archive.ics.uci.edu/ml) and KDD datasets(http:// osmot.cornerll.edu /kddcup/datasets.html), which shown in Table1. We select k-means [1, 16] and DBSCAN [14] as basedline methods. And we employ 3 evaluation metrics:(1) the metric of accuracy of clustering results that include purity, entropy, dunn and DBI. (2) the balance metric on accuracy of clustering results and degree of privacy protection, which include Calinski-Harbasez(CH) and F-measure. (3) the metric of privacy, communication and quality cost for all methods, which are deferred to the appendix because the limited page. For convenience, we simplify referred all methods as following:(1)ε_1+ε_2-DPCP denote DPCP under ε=ε_1+ε_2. (2)ε-DPCP denote DPCP under ε=ε_1=ε_2. (3)PPDBSCAN denote privacy preserving DBSCAN algorithm. (4)PPk-means denote privacy preserving k-means algorithm.

Table 1: Experimental data information

Dataset	Alias	Attribute number	Record number
Haberman	D1	4	306
Waveform Database	D2	40	5000
Biology Dataset	D3	74	145751

4.1 The accuracy validate of DPCP algorithm

Firstly, in order to validate the accuracy of DPCP algorithm for clustering, we use the following four measures.

(A)The **purity**[20] metric is calculated by equation $Purity = \frac{1}{n}\sum_{q=1}^{k} \max\limits_{1\leq j\leq l} n_q^j$. where, n is the total number of samples; l is the number of categories, n_q^j is the number of samples in cluster q that belongs to the original class j($1\leq j \leq l$).

It is a fitness function and also the goodness of formed clusters. A large purity is desired for a good clustering.

(B)The **entropy**[10] is calculated using the equation $e = -\sum_{j=1}^{K} \frac{m_i}{m} \cdot e_i, \quad e_i = -\sum_{L}^{j=1} p_{ij} \cdot log_2(\frac{m_{ij}}{m_i})$. where, e is the total entropy for a set of clusters, L is the number of classes, K is the number of clusters and m is the total number of data points. p_{ij} denotes the probability of a member of cluster i belongs the class j, m_i is the number of objects in cluster i, m_{ij} is the number of objects of class j in cluster i.

It is the degree to which each cluster consists of objects of a single class.

(C)The **dunn** index [5] is calculated using the equation $min_i\{min_j(\frac{min_{x\in C_i, y\in C_j} d(x,y)}{max_k\{max_{x,y\in C_k} d(x,y)\}})\}$. where, C_i is the i-th cluster; n_i is the number of objects in C_i; d(x,y) is the distance between x and y.

It is based on the minimum pairwise distance between objects in different clusters as the inter-cluster separation and the maximum diameter among all clusters as the intra-cluster compactness. The larger value of Dunn means better cluster configuration.

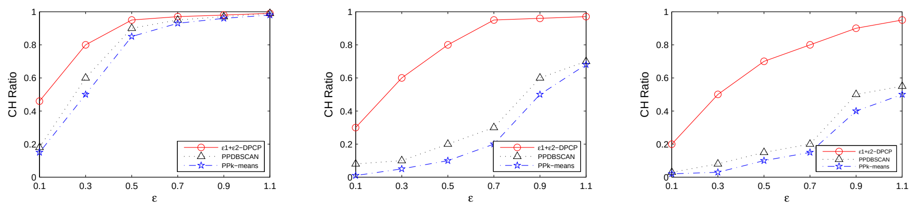
(D)The **DBI**[4] is calculated using the equation DBI= $\frac{1}{NC}\sum_i max_{j,j\neq i} \frac{[\frac{1}{n_i}\sum\limits_{x\in C_i} d(x,c_i) + \frac{1}{n_j}\sum\limits_{x\in C_j} d(x,c_j)]}{d(c_i,c_j)}$. where, c is the center of data set; NC is the number of clusters; C_i is the i-th cluster; n_i is the number of objects in C_i; c_i is the center of C_i; d(x,y) is the distance between x and y.

It calculates the similarities between each cluster C and other clusters, and the highest value is assigned to C as its cluster similarity. As the clusters should be compacted and separated, the lower DBI means better clustering.

Results are show in Table2. Because of the page limit,

Table 2: **Validate the accuracy of DPCP algorithm for clustering.**

Measure	Cluster-in-Cluster (K=3)	Cluster-in-Cluster (K=5)	Pin Wheel	Semi Circular (K=4)	Aggregation	Outlier	Compound
No.of Cluster	38	42	11	21	32	7	56
Purity(%)	99.8	99.294	98.93	99.4	92.413	99.81	98.15
Entropy	0.0818	0.1204	0.1285	0.1101	0.4507	0.0685	0.1748
Dunn(max)	1.3618	1.4102	1.5008	1.3109	2.1304	1.4002	1.4002
DBI(min)	1.04405	1.04505	1.05795	1.06105	1.25415	1.04605	1.07905



(a) Data set D1 (b) Data set D2 (c) Data set D3

Figure 5: CH ratio chart

two baseline methods are deferred to the appendix.

Table 2 shows the results over DPCP method, we can see the accuracy of DPCP algorithm is better. The achieved accuracy of DPCP in the Purity ranges around 99.2% and 99.3%, by the Entropy ranges around 0.069 and 0.4507, by the Dunn ranging around 1.31 and 1.5, by the DBI ranging around 1.04 and 1.25.

4.2 Result Analysis of CH Index

Then we evaluate other metrics CH for clustering results under privacy preserving. It computes a weighted ratio between the within-group scatter and the between group scatter. Well separated and compact clusters should maximize this ratio, as show in equation(4).Therefore, the larger the CH ratio is, the better the effectiveness of clustering will be.

$$CH(K) = \frac{\frac{1}{K-1} \times \sum_{i=1}^{K} N_i * d^2(c_i, c)}{\frac{1}{n-k} \times \sum_{i=1}^{K} \sum_{x \in C_i} d^2(x, c_i)} \tag{4}$$

Among them, let K is the number of subsets, thus, D={x_1, x_2,...,x_n} ={C_1, C_2,..., C_K} , C_i is a sub cluster of data set D, N_i is points number of C_i, c_i is the center point in C_i, c is the center in D, d(x,y) is a distance between x and y. Due to the random nature of the added noise, we perform the DPCP, PPDBSCAN, PPk-means algorithm many times on datasets D1,D2,D3, and then report the average of CH values as shown in Figure 5.

Usually use the parameter ε to measure the level of privacy protection, the smaller ε, the larger noise, thus the more powerful privacy protection can be achieved. So, CH ratio closter to 1 indicates that the clustering efficiency of the two clustering algorithms is more similar.

By observing the experimental results, we have the following conclusions. First, for three data sets, ε_1+ε_2-DPCP algorithm has the best performance in most cases. There are two possible reasons. Based on the Voroni diagram partition, the effect of privacy protection can be achieved by adding a small amount of noise. And the clustering results are close to the results of the original clustering algorithm. The second is that we introduce two parameters ε_1 and $\varepsilon_2$to protect ρ and δ. While PPk-means algorithm will add more noise with the increase of the number of iterations. In addition, we can measure the level of privacy protection by controlling ε value. By comparing the effect of privacy preserving clustering algorithm in each data set, we found that the clustering validity of large data sets is higher than K-means and DBSCAN.

4.3 Result Analysis of F-measure

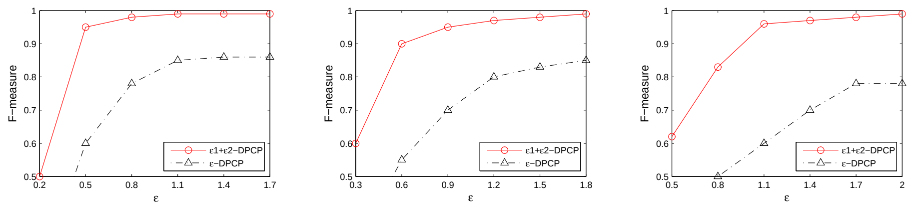
Finally, we evaluate other metrics F-measure for clustering results under privacy preserving. It is an external indicator to evaluate the effectiveness of clusters. The greater the calculated F-measure, the more similar the two algorithms are, that is, the effect of the difference privacy on the accuracy of the clustering results is small. The calculation method is as follows:

We use C to represent the results of DPC clustering on data sets, and use C_p to represent the clustering results of DPCP algorithm. X_i represents a cluster in C, Y_j represents a cluster in C_p, n_{ij}={X_i∩ Y_j} , |N| represents the number of data sets, According to equation (5), F(C_p) is the result to be calculated.

$$\begin{cases} recall(X_i, Y_j) = \dfrac{n_{ij}}{|X_i|}, \quad percision(X_i, Y_j) = \dfrac{n_{ij}}{Y_j} \\ F(X_i, Y_j) = \dfrac{2 \times recall(X_i, Y_j) \times precision(X_i, Y_j)}{recall(X_i, Y_j) + precision(X_i, Y_j)} \\ F(C_p) = \sum\limits_{X_i \in C} \dfrac{|X_i|}{|N|} \max\limits_{Y_j \in C_p} \{F(X_i, Y_j)\} \end{cases} \tag{5}$$

In order to measure the difference between the effect of ε-differential privacy protection and ε_1+ε_2- differential privacy protection in the same privacy budget, we perform the ε-DPCP algorithm and the ε_1+ε_2-DPCP algorithm on the data set D1, D2 and D3 respectively. When ε=1, there is ε_1=0.5, ε_2=0.5. Similarly, for each data set, we run the two algorithms several times, and then take the average value to draw the F-measure curve. The experimental results are shown in Figure 6.

By observing the experimental results of ε_1+ε_2-DPCP al-



(a) Data set D1 (b) Data set D2 (c) Data set D3

Figure 6: F-measure curves

gorithm and ε-DPCP algorithm, we can find that, under the same ε value, the results of F-measure have been greatly improved. This shows that the clustering validity of our algorithm is higher, at the same level of privacy protection. In addition, the experimental results also show that for large data sets D3, we can get better clustering results than ε-DPCP at a higher level of privacy protection. This is because after grouping, the added noise is less affected by the size of the data set. So, at the same level of privacy, the availability of large data sets clustering is even greater.

5. CONCLUSION

In this paper, we study the privacy preserving clustering problem and provide DPCP algorithm. We have provided $\epsilon = \epsilon_1 + \epsilon_2$-differential privacy preserving model. We provided performance analysis and privacy proof of our solution. The proposed approaches are evaluated through extecsive experiments, and we found that the density center clustering algorithm with differential privacy protection can obtain clustering solution close to the original algorithm, even in case of adding a small amount of noise.

6. REFERENCES

[1] A. Blum, C. Dwork, and F. Mcsherry. Practical privacy:the sulq framework. In *ACM Sigact-Sigmod-Sigart Symposium on Principles of Database Systems*, pages 128–138, 2005.

[2] P. Bunn and R. Ostrovsky. Secure two-party k-means clustering. In *ACM Conference on Computer and Communications Security(CCS)*, pages 486–497, 2007.

[3] R. Chen, Q. Xiao, Y. Zhang, and J. Xu. Differentially private high-dimensional data publication via sampling-based inference. In *The ACM SIGKDD International Conference*, pages 129–138, 2015.

[4] D. L. Davies and D. W. Bouldin. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):224–227, 1979.

[5] J. C. DunnâĂă. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.

[6] C. Dwork. Differential privacy. *Lecture Notes in Computer Science*, pages 1–12, 2006.

[7] C. Dwork, F. Mcsherry, and K. Nissim. Calibrating noise to sensitivity in private data analysis. In *Conference on Theory of Cryptography*, pages 265–284, 2006.

[8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, pages 226–231, 2008.

[9] G. Jagannathan and R. N. Wright. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In *Eleventh ACM SIGKDD*, pages 593–599, 2005.

[10] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[11] S. Jha, L. Kruger, and P. Mcdaniel. Privacy preserving clustering. *Lecture Notes in Computer Science*, 3679(4):397–417, 2005.

[12] J. J. KantarcioÇğlu, Murat and C. Clifton. When do data mining results violate privacy? In *Proceedings of the tenth ACM SIGKDD*, pages 599–604, 2004.

[13] K. A. Kumar and C. P. Rangan. *Privacy Preserving DBSCAN Algorithm for Clustering*. Springer Berlin Heidelberg, 2007.

[14] J. Liu, L. Xiong, J. Luo, and J. Z. Huang. Privacy preserving distributed dbscan clustering. *Transactions on Data Privacy*, 6(1):69–85, 2012.

[15] W. Lu, Y. Shen, S. Chen, and B. C. Ooi. Efficient processing of k nearest neighbor joins using mapreduce. *Proceedings of the VLDB Endowment*, 5(10):1016–1027, 2012.

[16] F. D. Mcsherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *ACM SIGMOD*, pages 19–30, 2009.

[17] P. Mohan, A. Thakurta, and E. Shi. Gupt: Privacy preserving data analysis made easy. In *Proceedings of the 2012 ACM SIGMOD*, pages 349–360, 2012.

[18] M. Redmond. Mechanism design via differential privacy. *Foundations of Computer Science Annual Symposium on*, pages 94–103, 2007.

[19] A. Rodriguez and A. Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[20] S. A. Salem and A. K. Nandi. New assessment criteria for clustering algorithms. In *Machine Learning for Signal Processing*, pages 285–290. IEEE, 2005.

[21] D. Su, J. Cao, N. Li, and E. Bertino. Differentially private k-means clustering. In *ACM on Conference on Data and Application Security and Privacy*, 2015.

[22] C. C. Yao. How to generate and exchange secrets. *Foundations of Computer Science Annual Symposium on*, 10:162–167, 1986.

[23] J. Zhang, X. Xiao, Y. Yang, and Zhang. Privgene: differentially private model fitting using genetic algorithms. In *ACM SIGMOD*, pages 665–676, 2013.