Density Based Distribute Data Stream Clustering Algorithm

Bing Gao
College of Computer Science and Technology,HEU,Harbin , China
Email: gaobing@neusoft.edu.cn

Jianpei Zhang
College of Computer Science and Technology,HEU,Harbin , China
Email: zhangjianpei@hrbeu.edu.cn

***Abstract*—To solve the problem of distributed data streams clustering, the algorithm DB-DDSC (Density-Based Distribute Data Stream Clustering) was proposed. The algorithm consisted of two stages. First presented the concept of circular-point based on the representative points and designed the iterative algorithm to find the density-connected circular-points, then generated the local model at the remote site. Second designed the algorithm to generate global clusters by combining the local models at coordinator site. The DB-DDSC algorithm can find the the clusters of different shapes under the distributed data stream environment, avoid frequently sending data by using the test-update algorithm, and reduce the data transmission. The experiments show that the DB-DDSC algorithm is feasible and scale expandable.**

***Index Terms*—data streams, data mining, clustering, distributed data stream**

I. INTRODUCTION

With rapid development of network communication technology, environmental monitoring, and sensor networks, a new data form named distributed data streams emerged. Data distribute in multiple concurrent data sources and each source generates its own data stream changing over time. For example, in network monitoring and communication, each site is a potential source of new data streams.

Clustering refers to the process that similar objects will be divided into one or more groups (called " clusters ") for a given data set. Elements are similar to each other in the same cluster, but different from those in the other cluster[1]. By combining these two aspects, we need to carefully consider how clustering in distributed data stream in order to identify the knowledge.

The simple and intuitive approach is that the distributed data streams are concentrated in the coordination site at first, and then to be clustering or do other operations. However, this method need to transfer all the data. There are some problems such as the long response time , the heavy traffic load , not suitable to resource-constrained environment , and privacy protection issues.

The current approach uses the in-network aggregation technology. The processing operations down to the network nodes, and each node computes the local summary information transmitted to the central node. Finally, the final clustering results are obtained on the secondary processing of summary information.

In literature [2] K-median clustering algorithm is applied to the distributed streams , for $(1 + \varepsilon)$ approximate clustering. In literature [3], the K-center clustering algorithm is applied to the distributed streams. Basing on the division method, the two algorithms can only find spherical clusters and need specify the number of clusters. In literature [4] using the expectation maximization(EM) algorithm for distributed data stream clustering , the algorithm is limited to specific models and is not able to deal with noises well.

In this paper, we propose the algorithm of DB-DDSC(Density Based Distribute Data Stream Clustering), a novel algorithm for discovering clusters of arbitrary shape in distribute data stream. Our algorithm consists of two stages , firstly, generates local clustering models - the collection of represent points at a remote site; secondly, generates global clusters by combining algorithm at the coordinator site.The algorithm's features are described below.

In this paper, we consider clustering over distributed data streams. We propose a suite of algorithms based on representative points. Our contributions are as follows:

(1) Proposed the concept of the circular point on the base of the representative point, analyzed the rationality that uses density connected circular points as local model, and designed the algorithm to get local model through iteratively searching the circular points.

(2) Designed the cluster algorithm based on density in the distributed data stream environment, which has two stages to generate global cluster model. The algorithm can find different shape and different size of clusters, avoid frequently sending data, and reduce the amount of transferred data through test-update local model algorithm

(3) The experiments have shown that the RB-DDSC algorithm has good cluster quality, faster processing speeds and less data traffic that could be adapted to

clustering requirements in distributed streaming environments.

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 gives the introduction about preliminaries. Section 4 presents the framework of DB-DDSC. Section 5 introduces the details of the generation and update local clustering model at remote sites. Section 6 describes the combining algorithm at coordinator site. Section 7 describes the performance study. Section 8 is a summary of the full text.

II. Relative Work

Several important data stream clustering algorithms have been proposed in recent. This includes CluStream[5] which Aggarwal proposed a data stream analysis frame. The CluStream is incremental and gives the different time granularity response. It uses the pyramidal time frame and divides the process into the on-line and off-line phrases. The on-line phrase uses the microcluster to calculate the statistical information. The off-line phrase carries on the macroclustering to provide the user different time granularity cluster result by using the pyramidal time frame.

In the following year, the HPStream algorithm framework was proposed in [6] and was for the high dimensional data streams. The main improvements include: first,HPStream introduced the projection clustering technique to handle high-dimensional problem. Secondly, HPStream uses the fading cluster structure to the preservation of historical data through the attenuation factor. The two above algorithms have a common disadvantage that it is difficult to find non-spherical clusters, and they all use the k-means algorithm that is not easy to deal with the noise.

The DenStream[7] algorithm expanded in the traditional data set cluster algorithm based on density method DBSCAN. The algorithm divides cluster analysis's process into p-micro-cluster and o-micro-cluster, then applies the concept of the nearest neighorhood to combine the points whose density are bigger than threshold, finally runs the DBSCAN algorithm handle the micro-cluster in order to get the cluster results. At the same time, it emphasized the isolated point examination question, separates out the isolated point and the normal data element. It focuses in the processing arbitrary shape clusters of the data stream.

But, the above algorithms are all based on a centralized, singular data stream environment and can not solve the problems in distributed conditions. So, there are some algorithms proposed in distributed environment as following.

Literature [8] based the Guha's clustering data stream algorithm, combined clusterings from each distributed site to find the global clustering with the same approximation factor. In many environments, the used k-median clustering are known to be less quality than those algorithm by other cluster techniques. At the same time, there are other clustering technology used over distributed streams.

DBDC[9] proposed a distributed density-based clustering algorithm. The algorithm first builds a local density-based cluster at each site, and then sends a summary of the clusters to a central site. The central site recontructs density-based clusters to find a global clustering based on the summaries come from all sites. The clustering result is returned back to the distributed sites and the sites update their local clusterings according to the discovered global clusterings. The transmission costs are a small number, because the local clusters only are part of origin data. But this approach is not able to process stream data.

EM-based[4] algorithm proposed expectation maximization method with Gaussian mixture model. This algorithm used a new test-update clustering idea to capture the distribution of the data stream, based on event-driven strategies to minimize the cost of communication in remote sites, as long as the remote site 's local distribution remain unchanged. The algorithm is suitable with incomplete or noisy data stream.

Literature [10] consider clustering parallel data streams. The authors used weighted exponential sliding windows to represent the data for finding relative streams when they arrived synchronously. The algorithm computes the discrete Fourier coefficient transform and perform k-Means clustering at the specific time. The method is suitable for online stream. After modified, the method can be applied to a distributed environment.

There are some other papers about distributed data streams, including outlier detection[11], top-k[12], frequent items [13,19], holistic aggregation[14], and sequential patterns [15]. Contrast with these basic statistics, Our algorithm focuses on the clustering is a more complex data mining problem.

III. Preliminaries

A. Distributed Data Stream Model

In order to modeling the above application, literature [16] first proposed the concept of the distributed data stream model, which refers to a number of data streams segmented by the level from multiple data sources, each point can be observed by its own data stream. At the coordinator site, the final statistical information or data mining is calculated on all the data streams. The structure of distributed data stream model is shown in figure 1 [6]. The model is composed of m remote sites and one coordinator site. Si(ts)is the data that arrive at timestamp ts on a remote sitei. Skti(ts) is the synopsis or local model on the sitei sent to the coordinator site at the timestamp ts. Skti(ts) can be constructed in different ways according to different algorithm. Remote sites may communicate each other, our paper concentrates on the case the remote site only communicate with coordinator site.

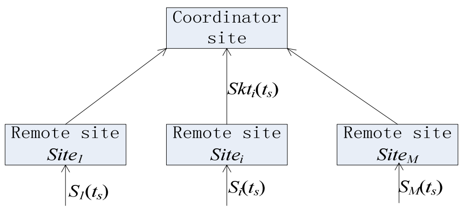


Figure.1 Distributed Data Stream Model

In a distributed data stream model, instead of submitting all data to a central site, the nodes perform calculations near the data, and when needed, send the local summaries to the coordinator site. The final results are calculated by the global algorithm at the coordinator site. Therefore, the model has three significant advantages:

(1) Processes the global tasks on multiple remote sites at the same time and improves computing speed.

(2) By sending the local model, this structure can reduce the traffic loads.

(3) The local model is not the original data, so can avoid privacy issue etc.

B. Synopsis and Local Model

In a distributed data stream model, the synopsis produced by the remote site is a summary of the data structure and represents a summary of the essential characteristics and basic information of the original data set. For a different problem you can use the different synopsis that needs to meet the following three basic properties:

(1) Compression: the synopsis should be smaller than the size of the original data, otherwise there is not necessary to generate synopsis.

(2) Combination : the synopsis of the set P can be calculated by combining the synopsis of the divided set of P, namely:

SM(P) = f(SM(P1), SM(P2))

(3) Error boundary : Because the generated synopsis of the original data is a summary of the information, the approximate solution of the synopsis should ensure $1+\varepsilon$ approximate(ε is the approximate factor).

For some issues such as cluster analysis in data mining, there is the need to handle the synopsis in order to generate local model, then send the local model to the coordinator site. For local model we need to consider three issues. First is the ability to accurately represent the original data. It is clear that the more information the model stores, the higher the degree of precision is. Second is to minimize the data transmission. The local model requires as much as possible to save space, as the representative of the original data. And finally, the global clusters or model can be obtained through computing and merging the local model. It can be drawn that the two previous issues are one contradiction, and we do the process of trade-offs dealing with specific problem.

C. Density-Based Clustering Algorithm DBSCAN

DBSCAN is a density-based clustering algorithm (Density-Based Spatial Clustering of Applications with Noise). The algorithm defines the clusters as the largest set of connected points about density. The basic idea involves a few of definitions as following[1].

Definition 1($\varepsilon-$ neighborhood). A given area within a $\varepsilon-$ radius around the object as the object of the neighborhood.

Definition 2(core object). The object whose neighborhood of a given radius (ε) has to contain at least a minimum number(*MinPts*) of objects.

Definition 3 (directly density-reachable). An object *p* is *directly density-reachable* from an object *q* wrt. ε and *MinPts* in the set of objects *D* if p is in the $\varepsilon-$ neighborhood of q and q is a core object.

Definition 4 (density-reachable). An object *p* is *density-reachable* from an object *q* wrt. ε and *MinPts* in the set of objects *D*, if there is a chain of objects $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that p_i and p_i+1 is directly density-reachable from p_i wrt.ε and *MinPts*.

Definition 5 (density-connected). An object *p* is *density-connected* to an object *q* wrt. ε and *MinPts* in the set of objects *D* if there is an object $o \in D$, both *p* and *q* are density-reachable from o wrt. ε and MinPts in D.

A cluster is defined as a set of density connected objects which is maximal density-reachability. And the noise is the set of objects not contained in any cluster.

The algorithm DBSCAN defined a cluster as density connected objects, and discovered a cluster through the process based on such fact that a cluster can be determined by one core object. Its basic philosophy is: select one point a in the database, if a is the core point, obtain this point's neighborhood through the region inquiry. The points in the neighborhood and the a belong to one cluster. Expand them to be the same cluster through the unceasingly region inquiry, until find one complete cluster. Then, seek for other clusters according to this process. Finally the left points not belong to any kind of cluster are noises.

The purpose of the DBSCAN algorithm is to filter low-density area, find the density sample points. Contrast with the traditional cluster based hierarchical clustering and partition clustering, the algorithm can find clusters of arbitrary shape. It has the following advantages:

With the K-means, you do not have to enter the number of clusters you have to divide;

The shape of the clusters is not bias;

Filter noise parameters can be entered when needed.

IV. Structure of DB-DDSC

This section gives the description of a distributed data stream clustering algorithm, referred to as DB-DDSC(Density Based Distribute Data Stream Clustering), the structure of the algorithm is divided into two phrases, the process on the remote sites and the process on the coordinator site. Brief description of algorithm DB-DDSC is shown as follows:

DB-DDSC_Remotei

1. Generate local model composed of represent points

2. If time changes, maintain local clusters CLUi
3. $\triangle$CLUi(tcurrent)$\leftarrow$CLUi(tcurrent)- CLUi(tcurrent-1)
4. If $\triangle$CLUi(tcurrent)>0 Send($\triangle$CLUi(tcurrent))
DB-DDSC_Coord
1. GLO0 $\leftarrow \cup$ i$\triangle$CLUi(tcurrent)
2. Maintain global clusters.

First,at the each remote site, the synopsis is generated as representative points. The set of reprentative points forms the local model which sent to the coordinator site. And when the new points arrive, the local model will be updated and sent. Second, at the coordinator site, when the local models are received from all remote sites, the calculation is carried out on the local models in order to generate the global cluster model. The global cluster is updated when receives the updated local model. In the follows, we will introduce the details of the two phrases.

V. Remote Site Processing

A. Representative Point

On a remote site, the core points of the data set can be generated by running the DBSCAN algorithm directly. The core points themseves represent at least MinPts points of $\varepsilon -$ neighborhood, and they can be sent to the coordinator site as local model generated by remote site. However, the number of core points may be very numerous, especially in the dense area of the clusters. Simply sending the core points will result in large transmission and long time of transmission. It is not feasible in a distributed situation by using DBSCAN algorithm to generate and transmit the core points.

Because the core points may be density , we do not need to transfer all of the core points. We can select the subset from the core points to represent all core points, which gives the definition of representative points.

Definition 6 (representative point) [17]**.** Let C be a cluster of the dataset wrt. ε and MinPts. $Cor_C \subseteq C$ is a set of core points belonging to this cluster. Then $Rcor_C \subseteq C$ is called a set of representative points of C iff the following three conditions are true.

$$Rcor_C \subseteq Cor_C \quad (1)$$

$$\forall s_i, s_j \in Rcor_C, s_i \neq s_j \Rightarrow s_i \notin N_\varepsilon(s_j) \quad (2)$$

$$\forall c \in Cor_C, \exists s \in Rcor_C, c \in N_\varepsilon(s) \quad (3)$$

In Figure 2. For instance, a, b and c are all the core points of cluster C. If a is a point in a representative point set, i.e. a$\in$ $Rcor_C$ and object b locates within $\varepsilon-$neighborhood of a. Object b can not be included in $Rcor_C$. Object c may be included in $Rcor_C$ that it doesn't locate in the a's $\varepsilon-$neighborhood. Another situation, if b is an element of the set of represents points, a and c will not be in $Rcor_C$ because they locate in the b's $\varepsilon-$ neighborhood. So there may be several different collections, which all meeting the definition of 6. The concrete set of representative points are determined by the actual processing order of the objects.

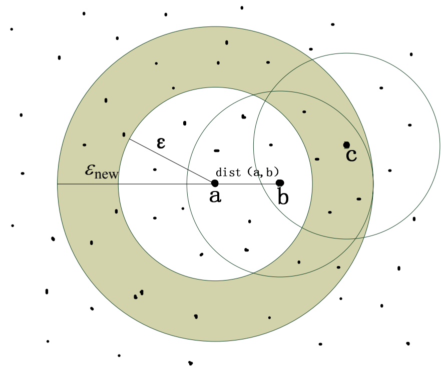


Figure. 2. Local model(representative point and ε_{new})

B. Local Model

If we already obtained n clusters as $C_1,..,C_n$, at a remote site k, we use the set of representative points $Rcor_{C_i}$ to represent every cluster. The local model is the union of different sets $Rcor_{C_i}$.

Using density based clustering method, there are often a number of core points located in the other core point's $\varepsilon-$ neighborhood. We need to consider the domain that representative point's denotation. In figure 3, if select the core point a as the representative point, we have $a \in$ Rcor and $b \notin$ Rcor according to condition 2 in Definition 6. If like this, object a will not only represent the data points of itself neighborhood, but also the data points of the b's $\varepsilon-$neighborhood, so a will represent the whole data points as $N_\varepsilon(a) \cup N_\varepsilon(b)$. We need to reallocate a new range $\varepsilon_a = \varepsilon + dist(a,b)$ for a, in the formula, $dist(a,b) < \varepsilon$ is the distance of a and b. Of course, the different representative point has different ε_i .

We must give the representative point of a new range ε to indicate the scope of data that can be represented. If $Rcor_C \subseteq C$ is the set of representative points, for every $S \in Rcor$ we specify a new ε_s to indicate the scope that the s represent.

$$\varepsilon_s = \varepsilon + \max\{dist(s,s_i) \mid s_i \in Cor \Lambda s_i \in N_\varepsilon(s)\}$$
(4)

Property 1. ε_s is less than or equal to 2ε, which is the max scope indicated by representative point s, i.e. ε_s<=2 ε .

It is obvious according to the formula (3) and (4).

Definition 7 (circular point). The point located between ε and ε_{new} is called circular point, such as the point c located in the gray part of the figure 3.

Property 2. The representative point in the same cluster must be circular point.

Proof: According to Definition 6, if the point a is representative point, the points in $N_{\varepsilon}(a)$ must not be representative points; The circular point satisfies the condition of formula (2); In the same cluster, the range of the representative point must be connected. So, there is the conclusion.

We designed the algorithm to generate local model. The algorithm iterately searches the circular point to find representative points, saves representative point and its corresponding ε_{new} as local model. LocalModel algorithm is shown as follows:

```
LocalModel(data )
1: for every point p do
2:   if p.unClassified {
3:       if expandCluster(p)   clusterID++;
     }
expandCluster(p)
4: if p is representative {
5:   save(p, $\varepsilon_p$ );
6:   getListofCircularPoints(p);
7: }
8:for every point q in ListofCircularPoints(p) {
9:   if q is core point {
10:       set q as representative;
11:       save(q, $\varepsilon_q$ );
12:       theListofCircular = addListofCircularPoints(q);
13:   }
14:   else { set q as noise; remove q; } }
15:}
```

Every representative point and its corresponding ε_s are saved as a tuple. All tuples form the set of local model on remote site k. The local model and noises are sent to the coordinator site for the second phase of the clustering.

Property 3. Suppose we have found n clusters $C_1, .., C_n$ on a remote site j, for the cluster C_i, $|Rcor_{C_i}|$ is the number of representative points, so the total number m of representative points which sent to the coordinator site is:

$$m_{remote_j} = \sum_{i=1..n} |Rcor_{C_i}|$$

C. Test-Update Algorithm

The local model and noises are saved at the remote site. When new data point arrives, the method first judges whether the point belongs to the local model. If the point belongs to the local model, we keep the model unchanged. Otherwise we need to consider to generate a new cluster by combining the point with noise or save the point as noise. If generated a new cluster, we need to merge the cluster with the existed model or let the cluster independent. In both cases, the local model was changed, so we update the local model and send it to coordinator site, shown as follows:

```
TestAndUpdate(LocalModel, data )
1: for every point p do
2:   if ( FitModel(localMode, p ) )
3:       return;
4: else{
5:     if( isNoise(p) )
6:         save p as noise;
7:     else{
8:         newCluster = p $\cup$ noise;
9:         if( newCluster $\cap$ localModel != $\phi$ )
10:            mergeModel(localMode, newCluster);
11:        else
12:            localModel = localModel $\cup$ newCluster;
13:        Send(localMode);
14: }}
```

According to the algorithm , the new coming data point is tested by the current local model. Because of the characteristics which the data arrives continuously in the data stream, the algorithm has good response time. First we test whether the point belongs to the existing model, and we don't update the model if the point belongs to the model. When the local model is changed, we need to send data to coordinator site and only send the affected data. Therefore, we avoid the frequent updating local model and sending data to coordinator site.

VI. Coordinate Site Processing

The representative point is composed of the tuple $<r, \varepsilon_r>$, represents the all data points wrt. ε_r and $N_{\varepsilon_r}(r)$, so the data points in $N_{\varepsilon_r}(r)$ belong to the same cluster. Therefore, the representative point is a cluster of itself. The global cluster should be generated by combining the representative points satisfied the conditions from different sites. The final global clustering model consists of merged representative points, sporadic representative points and noises, shown in Figure 3.

Property 4. Suppose we have m sites, on the coordinate site, the total number of representative points can be computed through the following formula according to the Property 3.

$$num = \sum_{j=1..m} m_{remote_j} = \sum_{j=1..m} \sum_{i=1..n} |Rcor_{C_i}|$$

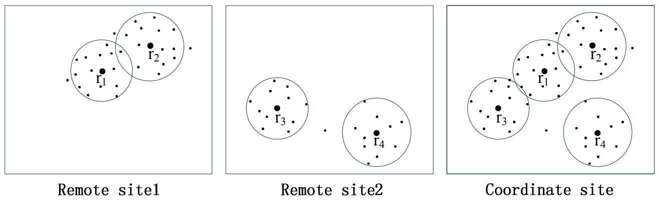


Figure.3. a global model

Because the representative point is already a cluster of itself, our algorithm can simplify the problem of the combining process. We can combine two representative points when their distance is less than the sum of their respective radius, i.e. $dist(r_1, r_2) < \varepsilon_{r_1} + \varepsilon_{r_2}$. When get the final global model, for every noise, we need to judge whether the noise locates in the scope of arbitrary representative point. If not, we save it as the final noise. Global clustering algorithm is shown as follows:

```
GlobalModel( Rcor )
1: gathering represent from cluster at every sites
2: for every represent r1 do
3:   compute dist(r1,r2); /* r2 come from another site*/
4:   if (dist(r1,r2)< $\varepsilon_{r_1} + \varepsilon_{r_2}$ ) {
5:     merge (r1,r2 );
6:   }
7: for each noise n1 do
8:   if( n1 not locate in arbitrary r )
9:       save n1 as noise;
```

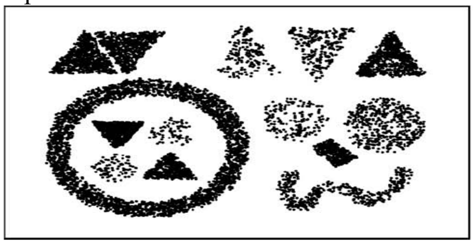
The method of global clustering is based the units consisting of representative points, not the local model. That is, when we judge whether needs to merge two local model coming from different site, we use the method that we merge two local models into a global model as long as there are two representative points whose distance is less than 2ε, because it indicates that two local models have overlap parts. Thus repeat this process, we can get all the global model.

When the change of local model (including the noises) is transferred to the coordinate site, we need to deal with the problem of cluster updating. Through the previously analysis of updating local model, we only consider the affection imposed on the existing model. If the change of the local model intersections with the existed model, we need to combine cluster, otherwise, a new cluster is generated. Global clustering algorithm is shown as follows:

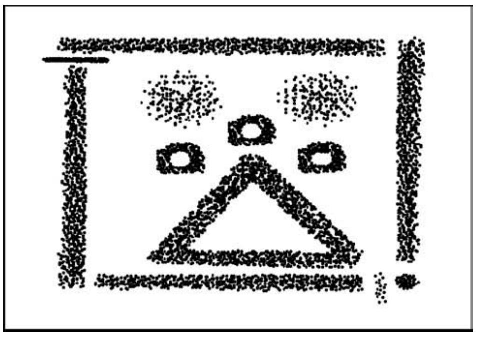
```
GlobalUpdate( Rcor )
1: when new changed local models arrive
2:for every changed represent r1 do
3:   if ( r1 $\cap$ globalModel != $\phi$ )
4:     merge( r1, globalModel);
5:   else
6:     save r1 as global Model
7:for each noise n1 do
8:   if( n1 not locate in globalModel )
9:       save n1 as noise;
```

VII. Experimental Evaluation

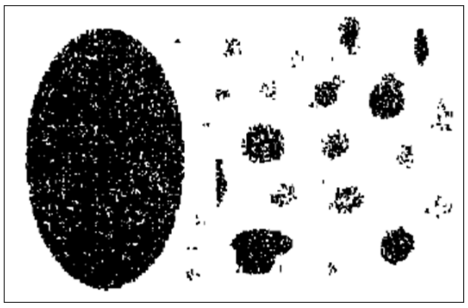
Here we evaluate the performance of DB-DDSC and compare it with DBSCAN. We implemented DB-DDSC algorithm and modified DBSCAN with Java under the software Weka 3.6.0. All experiments have been completed on a PC with a P2 CPU (1GHz) , 1G memory. We have used both synthetic datasets and the popular KDD-Cup-99 dataset.



(a) Db1



(a) Db2



(a) Db3

Figure 4 Three synthetic dataset

We evaluated the quality of our approach based on three different 2-dimensional synthetic datasets, two of which were also used in Ref [8] for testing DBSCAN algorithm, and one of which was used in Ref [9]. The three synthetic datasets are shown in Figure 4. Typical experimental results are given in Table 1. The results show that the representative points can indicate the correct number of clusters and find the clusters with arbitrary shape. On the db1 and db2, the number of clusters is less than the actual number and the reason is that the representative point has a larger ε_{new} that makes the close clusters combined. In the figure 4(a), the two triangle clusters are combined on the upper left corner. In the figure 4(b), the three clusters are combined on the upper left corner. There is a better quality of the clustering, because the clusters on the db3 have further distance each other.

TABLE 1.

PERFORMANCE COMPARISON(ε =9)

	Db1	Db2	Db3
Point number	9153	5458	20000
Cluster number	14	13	25
DB-DDSC	13	11	25

Our algorithm needs two parameters: ε and minPts which are the same to DBSCAN algorithm. The parameter ε_{new} is computed according the ε. Here, we test the affection of parameter ε. The results are shown in Figure 5. if ε is a smaller value, we can get the more clusters. If ε is a larger value, we can get the less clusters. So ε 's value is between 6 and 9, the quality of our algorithm is stable.

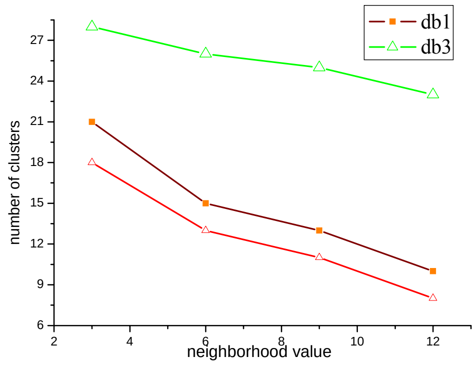


Figure 5 The affection of parameters

We evaluated the efficiency and transmission of our approach based on the synthetic dataset and the KDD-Cup-99 dataset. We use the 10% sub-sampled set containing 500,000 points and 41 dimensions. Typical experimental results are given in Figure 6 and Figure 7. When we test the runtime of our algorithm, we randomly divided the KDD-Cup-99 dataset into a number of sites. Figure 6 shows that with the number of sites increases, the runtime of our algorithm significantly decreases. Figure 7 shows the number of the representative points is far less than core points of DBSCAN with the increasing data. Furthermore, by comparing with (a) and (b), we can see our algorithm will have better adaptability with the data volume grow.

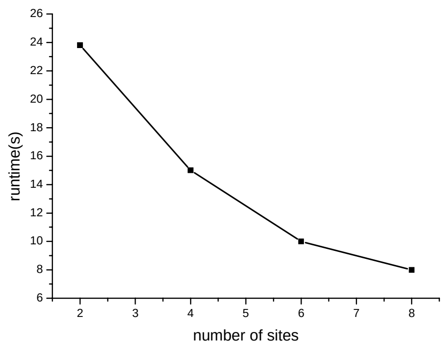
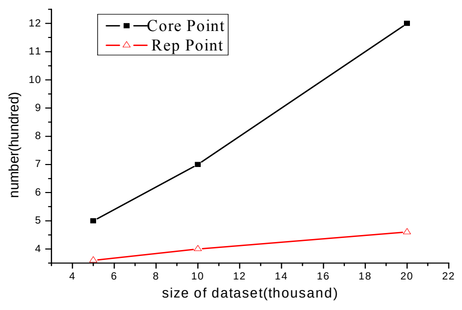
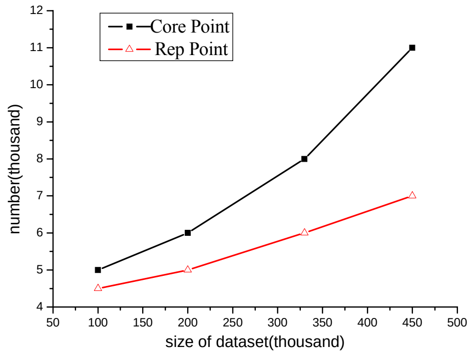


Figure 6 Efficiency



(a)on synthetic dataset(db3)



(b)on the KDD-Cup-99 dataset

Figure 7. Transmission volume:DBSCAN vs DB-DDSC (MinPts:20)

VIII. Conclusions

In this paper, we design the algorithm DB-DDSC, which transmits and uses the representative points to clustering. Proved by experiments, our algorithm can find the clusters with arbitrary shape, has the less runtime and transmission.

Our algorithm is suitable the landmark window model of data stream. We will improve our algorithm by intoducing the time factor to adapt the change.

We will do further experiments, focusing on comparing cluster quality and dealing with noise point. And then improve the algorithm to handle the temporal feature of data.

References

[1] Han J, Kamber M. Data Mining: Concepts and Techniques (Second Edition). Morgan Kaufmann , Elsevier Inc, 2006, 467-589.

[2] Qi Zhang, Jinze Liu, Wei Wang.Approximate Clustering on Distributed Data Streams[C]. Proc of the 2008 IEEE 24th International Conference on Data Engineering.2008: 1131-1139.

[3] Cormode G.,Muthukrishnan S,Wei Zhuang.Conquering the Divide:Continuous Clustering of Distributed Data Streams[C]. IEEE 23rd International Conference on Data Engineering.2007: 1036-1045.

[4] Aoying Zhou,Feng Cao,et al. Distributed Data Stream Clustering:A Fast EM-based Approach[C]. Data Engineering (ICDE 2007), 2007 IEEE 23rd International Conference on; Istanbul,Turkey.

[5] C.C.Aggarwal,J.Han,J.Wang, and P.S.Yu. A framework for clustering evolving data streams. In Proc.VLDB, 2003,81-92.

[6] C.C. Aggarwal, J. Han, J. Wang, P.S. Yu. A framework for projected clustering of high dimensional data streams, In Proc.VLDB, 2004.

[7] F. Cao, M. Ester, W. Qian, A. Zhou. Density-based clustering over an evolving data stream with noise. Proceedings of the Sixth SIAM International Conference on Data Mining, 2006.

[8] A. Ghoting and S. Parthasarathy. Facilitating interactive distributed data stream processing and mining. In Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), 2004.

[9] E. Januzaj, H. Kriegel, and M. Pfeifle. DBDC: Density based distributed clustering. In Proceedings of the International Conference on Extending Data Base Technology (EDBT), 2004.
[10] J. Beringer and E. Hullermeier. Online clustering of parallel data streams. Data and Knowledge Engineering, 2005.
[11] M. Otey, A. Ghoting, and S. Parthasarathy. Fast distributed outlier detection in mixed attribute data sets. Data Mining and Knowledge Discovery Journal, 2006.
[12] B. Babcock and C. Olston. Distributed top-k monitoring. In Proc. of SIGMOD, 2003.
[13] Ling Chen, Yixin Chen, Li Tu. A Fast and Efficient Algorithm for Finding Frequent Items over Data Stream. Journal of Computers. 2012,7(7):1545-1554.
[14] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi. Holistic aggregates in a networked world: Distributed tracking of approximate quantiles. In Proc. of SIGMOD, 2005.
[15] Shih-Yang Yang, Ching-Ming Chao, Po-Zung Chen, Chu-Hao Sun. Incremental Mining of Across-streams Sequential Patterns in Multiple Data Streams. Journal of Computers. 2011,6(3):449-457.
[16] Gibbons P, Tirthapura S. Estimating simple functions on the union of data streams. In: Proc. of the ACM Symp. on Parallel Algorithms and Architectures. Crete Island, 2001. 281−291.
[17] S. Lühr and M. Lazarescu, "Incremental clustering of dynamic data streams using connectivity based representative points," Data & Knowledge Engineering, vol. 68, pp. 1-27.
[18] Shuigeng Zhou, Aoying Zhou, Wen Jin, Ye Fan, Weining Qian. FDBSCAN: A Fast DBSCAN Algorithm. Journal of Software. 2000,11(6):735-744.
[19] Keming Tang, Caiyan Dai, Ling Chen. A Novel Strategy for Mining Frequent Closed Itemsets in Data Streams. Journal of Computers. 2012,7(7):1564-1573.

Bing Gao Liaoning Province, China.Birthdate:September ,1976. is Computer Application Master.M., graduated from Dept. Computer Science and Information Technology Haerbin Engineering University. And research interests on Data Stream, Data Mining, Machine Learning.

Jianpei Zhang Heilongjiang Province, China. Birthdate: November, 1956. is Computer Software Ph.M., graduated from Dept. Computer Science and Information Technology Haerbin Engineering University. And research interests on Datahouse Theory, Data Mining,Software Theory. He is a professor of Dept. Computer Science and Information Technology Haerbin Engineering University.