International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 4, June - 2012

Clustering Data Stream for Point Density with Mobile Agent

Monali Patil, Vidya Chitre, Dipti Patil

Department of Information Technology, K. C. COE, Thane
Department of Computer Engineering, Bharati Vidyapidth, Navi Mumbai
Department of Information Technology, PIIT, New Panvel
Mumbai University, Maharashtra, India

.

***Abstract*— The tremendous amount of data produced now a days in various application domains such as molecular biology or geography can only be fully exploited by efficient and effective data mining tools. One of the primary data mining tasks is clustering, which is the task of partitioning points of a data set into distinct groups (clusters) such that two points from one cluster are similar to each other whereas two points from distinct clusters are not. The detection of clusters in a given dataset is important for data analysis. This paper presents a possible DBSCAN clustering algorithm implementation. DBSCAN algorithm is based on density reachable and density connected point. Adding mobile Agent technique to density clustering algorithm we can improve clustering results as online and offline processing becomes parallel.**

***Keywords*— Clustering, knowledge discovery, dataPoint, DBSCAN, density-reachable, density- connected.**

I. INTRODUCTION

Due to the enormous amount of data in various application domains, the requirements of database systems have changed. Techniques to analyze the given information and find so far hidden knowledge are mandatory to draw maximum benefit from the collected data. Knowledge Discovery in Databases (KDD) is an interdisciplinary field, aimed at extracting valuable knowledge from large databases [6]. At the core of the KDD process is the Data Mining step which embraces many data mining methods, one of them is clustering.

Clustering approaches can be classified into partitioning methods, and hierarchical methods. Partitioning clustering algorithms compute a "flat" partition of the data into a given number of clusters, i.e. a unique assignment of each data object to a cluster. The number of clusters k is often a user specified parameter. There are several types of partitioning methods, optimization Based Methods, Distribution- (or Model-) Based Methods, Graph Theory Based Methods, Density-Based Methods. In this paper we are using Density-Based DBSCAN method [4].

The density-based notion is a common approach for clustering, used by various algorithms such as DBSCAN, DBCLASD, DENCLUE, and OPTICS [5]. All these methods search for regions of high density in a feature space that are separated by regions of lower density. DBSCAN was the first density-based spatial clustering method proposed to define a new cluster or to extend an existing cluster, a neighbourhood around a point of a given radius (Eps) must contain at least a minimum number of points (MinPts), the minimum density for the neighbourhood. DBSCAN starts from an arbitrary point q. It begins by performing a region query, which finds the neighbourhood of point q. If the neighbourhood is sparsely populated, i.e., it contains fewer than MinPts points, then point q is labeled as noise. Otherwise, a cluster is created and all points in q's neighbourhood are placed in this cluster. Then the neighbourhood of each of q's neighbours is examined to see if it can be added to the cluster. If so, the process is repeated for every point in this neighbourhood, and so on. If a cluster cannot be expanded further, DBSCAN chooses another arbitrary unlabelled point and repeats the process. This procedure is iterated until all points in the dataset have been placed in clusters or labelled as noise.

II. THE CONCEPTS AND DEFINITIONS

When looking at the sample sets of points depicted in figure 1, we can easily and unambiguously detect clusters of points and noise points not belonging to any of those clusters.

Fig. 1 Sample Databases

The main reason why we recognize the clusters is that within each cluster we have a typical density of points which is considerably higher than outside of the cluster. Furthermore, the density within the areas of noise is lower than the density in any of the clusters [1].

In our implementation, we try to formalize this intuitive notion of "clusters" and "noise" in DataPoint elements and

space (distance) is specified by matrix. The key idea is that for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points, i.e. the density in the neighborhood has to exceed some threshold. The shape of a neighborhood is determined by the choice of a distance function for two points p and q, denoted by *dist(p,q)*.

Definition 1: (Eps-neighborhood of a point) The *Epsneighborhood* of a point p, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p) = \{q \in D \mid dist(p,q) \leq Eps\}$.

A naive approach could require for each point in a cluster that there are at least a minimum number (*MinPts*) of points in an Eps-neighborhood of that point. However, this approach fails because there are two kinds of points in a cluster, points inside of the cluster (*core points*) and points on the border of the cluster (*border points*). In general, an Eps-neighborhood of a border point contains significantly less points than an Eps-neighborhood of a core point. Therefore, we would have to set the minimum number of points to a relatively low value in order to include all points belonging to the same cluster. This value, however, will not be characteristic for the respective cluster - particularly in the presence of noise. Therefore, we require that for every point p in a cluster C there is a point q in C so that p is inside of the Eps-neighborhood of q and $N_{Eps}(q)$ contains at least MinPts points. This definition is elaborated in the following.

Definition 2: (directly density-reachable) A point p is *directly density-reachable* from a point q wrt. Eps, MinPts if

1) $p \in N_{Eps}(q)$ and

2) $|N_{Eps}(q)| \geq$ MinPts (core point condition).

Obviously, directly density-reachable is symmetric for pairs of core points. In general, however, it is not symmetric if one core point and one border point are involved. Figure 2 shows the asymmetric case [1].

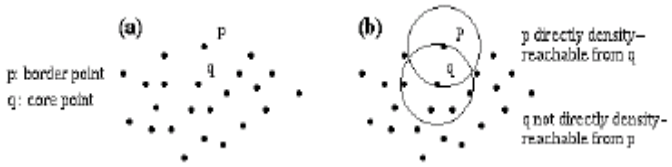

Fig. 2 Core points and border points

Definition 3: (density-reachable) A point p is *density reachable* from a point q wrt. Eps and MinPts if there is a chain of points $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

Density-reachability is a canonical extension of direct density-reachability. This relation is transitive, but it is not symmetric. Figure 3 depicts the relations of some sample points and, in particular, the asymmetric case. Although not symmetric in general, it is obvious that density-reachability is symmetric for core points. Two border points of the same cluster C are possibly not density reachable from each other because the core point condition might not hold for both of them. However, there must be a core point in C from which both border points of C are density-reachable. Therefore, we introduce the notion of density-connectivity which covers this relation of border points.

Definition 4: (density-connected) A point p is *density connected* to a point q wrt. Eps and MinPts if there is a point o such that both, p and q are density-reachable from o wrt.Eps and MinPts.

Density-connectivity is a symmetric relation. For density reachable points, the relation of density-connectivity is also reflexive (c.f. figure 3).

Now, we are able to define our density-based notion of a cluster. Intuitively, a cluster is defined to be a set of density connected points which is maximal with respect to density-reachability. Noise will be defined relative to a given set of clusters. Noise is simply the set of points in *D* not belonging to any of its clusters.

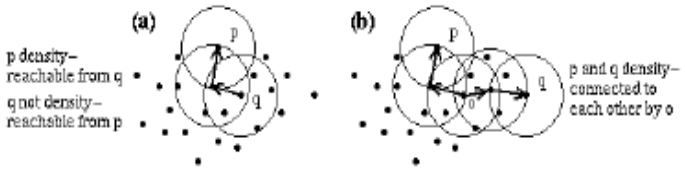

Fig. 3 Density-reachability and density-connectivity

Definition 5: (cluster) Let *D* be a database of points. A *cluster C* wrt. Eps and MinPts is a non-empty subset of *D* satisfying the following conditions:

1) $\forall$ p, q: if $p \in C$ and q is density-reachable from p wrt. Eps and MinPts, then $q \in C$. (Maximality)

2) $\forall$ p, $q \in C$: p is density-connected to q wrt. EPS and MinPts. (Connectivity)

Definition 6: (noise) Let C1 ,. . ., Ck be the clusters of the database D wrt. parameters Epsi and MinPtsi, i = 1, . . ., k. Then we define the noise as the set of points in the database D not belonging to any cluster Ci , i.e. noise = $\{p \in D \mid \forall i: p \notin Ci\}$.

Note that a cluster C wrt. Eps and MinPts contain at least MinPts points because of the following reasons. Since C contains at least one point p, p must be density-connected to itself via some point o (which may be equal to p). Thus, at least o has to satisfy the core point condition and, consequently, the Eps-Neighborhood of o contains at least MinPts points.

DBSCAN can be viewed as a heuristic method that uses a depth-first local spanning search. It randomly selects the first point, saying p, finds its neighbourhood, and checks whether p and its neighbours cover the whole cluster. If not, it picks a neighbour of p, called it q, adds it to the set, and checks its neighbours. If q is a border point, the next selected point is another neighbour of p. If q is a core point, the next point will be one of q's neighbours. The process continues until the whole cluster has been covered. The selected points may not be skeletal points, but together they form a cover for the corresponding neighbourhood graph.

Lemma 1: A density-based cluster corresponds to a connected neighborhood sub-graph with density-reachable used as the neighbor relation.

From Lemma 1, given n points, the clustering process of DBSCAN can be viewed abstractly as constructing neighbourhood graphs. Each time a core point is found, the

algorithm finds the directly density-reachable relation between the core point and each of its neighbours. The directly density-reachable relation holding for the two points can be viewed as the directed edge between the two corresponding vertices in the neighbourhood graph. Each cluster in the dataset is constructed as a connected neighbourhood sub-graph. Without considering noise, if a dataset has k clusters, then its corresponding neighbourhood graph will have k connected sub-graphs.

For example, suppose the nine points in Figure 4(A) are in one cluster. We assume MinPts is 3. DBSCAN is applied with Point 1 arbitrarily selected as the initial point [3]. The region query for Point 1 finds that Points 2, 3, 4, 5 and 6 are Point 1's neighbours. These points are shown inside the circle centered on Point 1 in Figure 4(A).

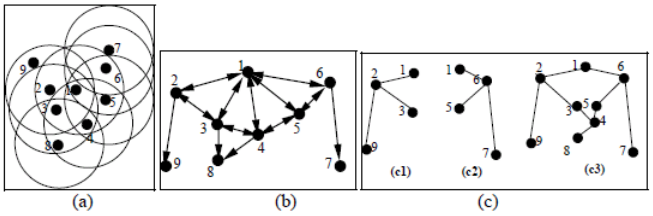

Fig. 4 (A) EXAMPLE CLUSTER; (B) STRONGLY CONNECTED NEIGHBOURHOOD GRAPH; (C) CONNECTED NEIGHBOURHOOD GRAPH

So edges from 1 to its neighbours are inserted in the neighbourhood graph. Points 2, 3, 4, 5 and 6 are organized in a list and checked for neighbours one by one, and so on for their neighbours. When DBSCAN terminates, the neighbourhood graph is connected, as shown in Figure 4(B).

Lemma 2: If the density-reachable relation is the neighbor relation, DBSCAN's clustering process corresponds to constructing the strongly connected neighborhood graph.

In Figure 4(B), for any two points if one point is density-reachable from the other, then a directed path connects them. So, Figure 4(B) shows a strongly connected neighborhood graph.

Lemma 3: A purity-density-based cluster correspond to a connected neighborhood graph with PD-reachable used as the neighbor relation.

III. DBSCAN ALGORITHM

```
DBSCAN(D, eps, MinPts)
  C = 0
  for each unvisited point P in dataset D
    mark P as visited
    N = getNeighbors (P, eps)
    if sizeof(N) < MinPts
      mark P as NOISE
    else
expandCluster(P, N, C, eps, MinPts)
  add P to cluster C
  for each point P' in N
    if P' is not visited
      mark P' as visited
      N' = getNeighbors(P', eps)
      if sizeof(N') >= MinPts
        N = N joined with N'
    if P' is not yet member of any cluster
      add P' to cluster C
  C = next cluster
  expandCluster(P, N, C, eps, MinPts)
```

DBSCAN visits each point of the database, possibly multiple times (e.g., as candidates to different clusters). For practical considerations, however, the time complexity is mostly governed by the number of getNeighbors queries. DBSCAN executes exactly one such query for each point, and if an indexing structure is used that executes such a neighbourhood query in O(log n), an overall runtime complexity of O(n.log n) is obtained. Without the use of an accelerating index structure, the run time complexity is $O(n^2)$. Often the distance matrix of size $(n^2 - n) / 2$ is materialized to avoid distance recomputations. This however also needs $O(n^2)$ memory.

IV. MOBILE AGENT TECHNIQUES

An agent is a program, which can act independently and autonomously. A mobile agent is an agent, which can move in various networks under their own control, migrating from one host to another host and interacting with other agents and resources on each. When a mobile agent's task is done, it can return to its home or accept other arrangement. The structures of the mobile agent are different for the different system. However, there are two parts generally speaking, which are MA (Mobile Agent) and MAE (Mobile Agent Environment). MAE realizes the migration of the mobile agents among the hosts using agent transfer protocol and distributes the executing environment and service interface to them [1]. It also controls the security, communication, basic service and so on. MA is above the MAE and it can move into another MAE. MA can communicate with other MA using agent communication language.

V. THE NEW FRAMEWORK BASED ON MOBILE AGENT

We propose a new framework based on the mobile agent to gather and cluster the data. The Fig.2 shows that the structure of the framework based on mobile agent. There are three sorts of the mobile agents: the main agent, the subagent and the result agent. The main agent is in the online component and it gathers the data in every gap. It distributes the each dimension data to the right subagent. If we have d-dimensional data, we need about d subagents. However, during clustering the data, if the data is not changed in a certain dimension, the main agent will distribute another dimension data to this subagent. This method can save the space.

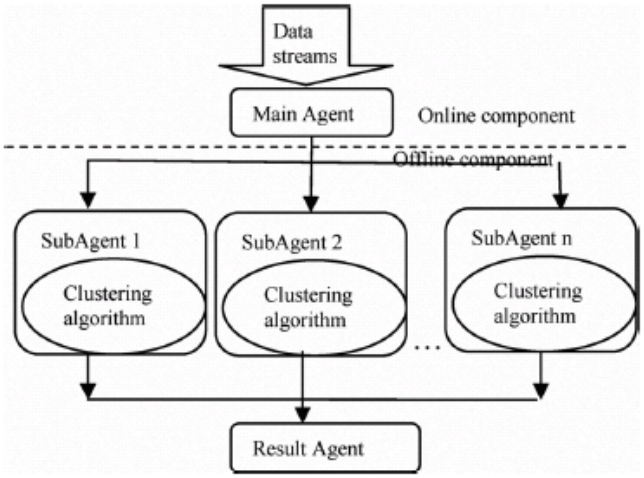

Fig. 2 The structure of the data streams clustering based on mobile agent

Every subagent contains a clustering algorithm model. When a subagent receives the data from the main agent, it detects whether there is change of the data. If there is no change in this dimension, it will send the message to the result agent to use the last time result. Then send the message to the main agent, the main agent will distribute another dimension data to it. If there is change in this dimension at this gap, the subagent creates new grids to cluster the data. This method avoids creating large number grids every time. All subagents execute clustering task parallel. So if each dimension is divided into 20 regions, each subagent only creates 20 grids so that it reduces the complexity of the computation and overcome the challenge of the large number of the grids. The result agent obtains all the clustering results of the subagents. The CLIQUE algorithm clustering the spatial data can use Apriori property that is if a k-dimension is dense, its projection on the (k-l)-dimension is also dense. So the result agent can get the final result through finding the common clustering parts of all the dimensions. The advantages of the structure are: The offline component is very flexible since the mobile agent can move and communicate with others. If subagent has some error, another unoccupied subagent can instead of it quickly. The programming is parallel and the clustering algorithm uses the DNA computing techniques, which is also parallel. This method can save much executing time. Each subagent only divides one dimension into grids, which can reduce the complexity of the computation. If a subagent detect there is no change at this interval time, it will not work or require to another dimension clustering, which can save the space [9].

VI. PROCESS OF THE CLUSTERING ALGORITHM

The initial clustering part is not changed and we improve the clustering part and add the programming of the mobile agent including the main agent, the subagent and the result agent. The clustering algorithm is described as follow:

Step l: Main agent gets the data in a gap time.

Step2: Main agent distributes the data into the subagents. Each subagent gets one dimension data. If the data is d-dimension, we need d subagents.

Step3: The subagent detects the data. If there is no change compared with the last gap, go to step5. Or else go to step4.

Step4: The subagent creates new grids for the new data.

Step5: The subagent sends the result to the result agent and sends the message to the main agent to require the new task.

Step6: All the results of the subagent are collected in the result agent. The result agent computes these results and gets the final result. Go to step 1.

VII. CONCLUSIONS

Clustering algorithms are attractive for the task of class identification in spatial databases. However, the well-known algorithms suffer from severe drawbacks when applied to large spatial databases. In this paper, we presented the clustering algorithm DBSCAN which relies on a density-based notion of clusters. It requires only one input parameter and supports the user in determining an appropriate value for it.

This DBSCAN is extremely good and is efficient in many datasets clustering spatial data in the knowledge discovery. However, by adding mobile Agent Technique we can work for multi dimensional data map it to grid using online and offline component and increase the efficiency of clustering method.

REFERENCES

[1] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, Published in Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining.

[2] Liadan O'Callaghan , Nina Mishra, Adam Meyerson, “Streaming-Data Algorithms For High-Quality Clustering “.

[3] Xin Wang, Howard J. Hamilton, “ A Comparative Study of Two Density-Based Spatial Clustering Algorithms for Very Large Datasets”.

[4] Xin Wang, “Density-Based Spatial Clustering Methods for Very Large Datasets”, December 2006.

[5] Karin Kailing Tag, “New Techniques for Clustering Complex Objects”, 2004.

[6] S. Guha, R. Rastogi, and K. Shim, “An efficient clustering algorithm for large databases”, In Proc. SIGMOD, pages 73{84, 1998.

[7] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, ”Clustering data streams”, In Proc. FOCS, pages 359 366, 2000.

[8] Jiaowei Tang, “An Algorithm for Streaming Clustering”, Examensarbete 30 hp Mars 2011.

[9] Zhang Hongyan, Liu Xiyu, “ A Data Streams Clustering Algorithm Using DNA Computing Techniques Based on Mobile Agent”, IEEE, 2009, Science and Technology Project of Shandong Education Bureau.