Information Technology Journal 8 (4): 476-485, 2009
ISSN 1812-5638

Clustering Large Spatial Data with Local-density and its Application

Guiyi Wei, Haiping Liu and Mande Xie
College of Computer Science and Information Engineering, Zhejiang Gongshang University,
Xuezheng Street No. 18, Hangzhou, Zhejiang, People's Republic of China

Abstract: In this study, a new algorithm LD-BSCA is proposed with introducing the concept of local MinPts (a minimum number of points) and the new cluster expanding condition: ExpandConClId (Expanding Condition of ClId-th Cluster). We minimize the algorithm input down to only one parameter and let the local MinPts diversified as clusters change from one to another simultaneously. Experiments show LD-BSCA algorithm is powerful to discover all clusters in gradient distributing databases. In addition, we introduce an efficient searching method to reduce the runtime of present algorithm. Using several databases, we demonstrate the high quality of the proposed algorithm in clustering the implicit knowledge in asymmetric distribution databases.

Key words: Spatial data mining, clustering expanding condition, minimum number of points, border points processing

INTRODUCTION

As larger and larger amounts of spatial data being collected and stored in databases, the demand of effectively analyzing and utilizing knowledge implicitly contained in the data is increasing. Today, clustering is one of the major data mining methods for knowledge discovery in the large spatial databases. Usually, it is also a preprocessing step for other mining methods. It is a generic term in the process of groups similar objects into a meaningful set (cluster) in which objects exhibit certain degree of similarities. It also separates dissimilar objects into different clusters and helps users to understand the natural grouping or structure in a data set. Therefore, the development of improved clustering algorithms has got a lot of attention in the last few years.

Clustering algorithms are very useful to the geographical and medical communities and they can be categorized into five main types (Han and Kamber, 2001) partitioning, hierarchical, density-based (Chehreghani *et al*., 2008; Rezaie *et al*., 2007; Ding and Yuan, 2008), grid-based and model-based clustering algorithms. Partitioning methods like k-means (MacQueen, 1967), k-medoid (Vinod, 1969; Jim and Liaw, 2008) which make use of a technique called iterative reallocation to improve the clustering quality based on initial partitioning solutions. Hierarchical clustering algorithms mend the membership of data object and cluster when the object is allocated to a cluster, such as DIANA (Kaufman and Rousseeuw, 2005) and BIRCH (Zhang *et al*., 1996). Instead of using distance to judge the membership of a data object, density-based clustering algorithms like DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ankerst *et al*., 1999) utilizes the density of data points within a region to discover clusters. To define the density around the region of a data object, DBSCAN requires inputting two parameters. Only the points within dense region will form clusters. But these density-based methods like DBSCAN and OPTICS are index-based which will face efficiency breakdown when the number of dimensions is high. The grid-based clustering methods, such as STING (Wang *et al*., 1997) and WaveCluster (Sheikholeslami *et al*., 1988, 2000), approximate the dense regions of the clustering space by quantizing it into a finite number of cells and identifying cells that contain more than one point dense number. Clusters are then formed by connecting these dense cells. The model-based clustering methods try to optimize the fitness between the given data and certain mathematics model. Each clustering method has its own advantage. Although a grid-based approach is usually more efficient than a density-based approach, the use of summarized information causes it to lose effectiveness as the number of dimensions increases. Therefore, in this study we will scrutinize and improve the density-based clustering method (Chehreghani *et al*., 2008).

Density-based clustering algorithms concern for class identification in spatial data mining. Clusters are formed as regions in the data space in which the objects are of similar density and which are separated by regions

Corresponding Author: Guiyi Wei, College of Computer Science and Information Engineering, Zhejiang Gongshang University, Xuezheng Street No. 18, Hangzhou, Zhejiang, People's Republic of China
Tel:+86-136-0661-9504 Fax: +86-571-2800-8303

with different object density. These regions may have an arbitrary shape and the points inside a region may be arbitrarily distributed.

To find high-density regions in the data space, a quick method (Jain and Dubes, 1998) based on grid cell densities is proposed. It constructs a histogram by partitioning the data space into a number of non-overlapping cells. These cells containing a relatively large number of objects are potential cluster centers and the boundaries between clusters fall within the valleys of the histogram. However, the size of the cells which must be assigned by the user has a significant influence on the outcome of this clustering method. Cells of too small volume will give a very noisy estimate of the density, whereas too large cells tend to overly smooth the density estimate.

As a density-based clustering method, DBSCAN does not need partition the data space into cells. The key idea of this algorithm is that for each point of a cluster the neighborhood of a given radius ε has to contain at least a minimum number of points (MinPts). In other words, the cardinality of the neighborhood of each point has to exceed the threshold. It requires the input of ε and MinPts. The result is very sensitive to these parameters.

Another algorithm OPTICS creates an expanded order of the database representing its density-based clustering structure. This cluster order contains information which is equivalent to the density-based clustering corresponding to a broad range of parameter settings. However, it only generates the clusters whose local-density exceeds some threshold instead of similar local-density clusters and does not produce clusters of a data set explicitly.

The density- and grid-based algorithm WaveCluster applies wavelet transformation to the feature space (Li *et al*., 2003; Li and Xu, 2001). The algorithm can detect clusters of arbitrary shape at different scales with complexity of O(n).

The density-based algorithm DENCLUE (Hinneburg and Keim, 1998) uses a grid which is very efficient, since it only keeps information about grid cells that do actually contain data points and manages these cells in a tree-based access structure. This algorithm generalizes some other clustering approaches which, however, result in a large number of input parameters.

The density and grid-based clustering technique CLIQUE (Agrawal *et al*., 1998) has been proposed for data mining in high-dimensional data space. Input parameters are the size of the grid and a global density threshold for clusters. The major difference between this algorithm and all other clustering approaches is that this method also detects subspaces of the highest dimensionality as high-density clusters existed in those subspaces.

The clustering algorithm CURD (Ma *et al*., 2003) captures the shape and extent of a cluster by references and then analyzes the data based on the references. Targeted mining very large databases, CURD can discover clusters with arbitrary shape, but it is insensitive to noise data. Experiments show there is also some problem. Especially, effectiveness of it is not high enough.

A new shifting grid clustering algorithm (Ma *et al*., 2004) uses the concept of shifting grid. The algorithm is a nonparametric type, which does not require users inputting parameters. It divides each dimension of the data space into certain intervals to form a grid structure in the data space. It clusters data in terms of cell rather than points.

The density-grid based clustering GCHL (Pilevar and Sukumar, 2005) uses axis-parallel partitioning strategy with a hybrid technique to identify areas of high density in the input data space. This algorithm works well in the feature space of any data set and operates on a limited memory buffer and requires at most a single scan through the data. It is certainly effective to discover concave/deeper and convex/higher regions, their robustness to outlier and noise.

A local-density based spatial clustering algorithm with noise relying on a local-density-based notion of clusters is proposed by Duan *et al*. (2007). It takes the advantage of the LOF (Local Outlier Factor) to detect the noises comparing with other density-based clustering algorithms. It can discover different density cluster existing in different regions of data space. But it needs a user to input three parameters which are hard to determine but have a significant influence on the clustering result.

As clustering is an unsupervised learning process, ideally it should be a non-parametric procedure, which does not require users to input parameters. Many clustering algorithms, however, require users to input several parameters, such as the number of cluster and the initial location of each cluster center. These parameters always pose significant influences on the clustering results. Usually, user does not know enough information to determine these input parameters. Therefore, obtaining reasonable clustering results requires testing large different initializations. Minimizing input parameters will certainly be very useful in reducing the errors introduced by human interference. In this study, we want to reduce the number of conventional algorithms' input parameters. Based on DBSCAN algorithm, we improve it and propose LD-BSCA which needs only one input parameter. When users change this parameter into different values, the fluctuation of local density is lower than DBSCAN clustering method which is needed to determine two input parameters. In DBSCAN, users should peg one of the parameters and change another for discovering clusters more effectively. Using LD-BSCA, users only need alter the only global parameter radius value, meanwhile the pivotal variable local MinPts is adjusted automatically. Therefore, the clustering result is exacter than DBSCAN. Another advantage of LD-BSCA method is that we import a more efficient expanding method which is described in lemma 2.

BASIC NOTIONS

Problems of existing clustering algorithms: An important property of most real data sets is that their intrinsic cluster structures are unable to be characterized by a global density parameter. As a result, very different local densities may be needed to reveal clustering in different regions of the data space. The fundamental idea of the algorithm DBSCAN is that for each point of a cluster the neighborhood of a given radius ε has to contain at least a MinPts points where ε and MinPts are user input parameters. Otherwise, DBSCAN can not generate the clusters accurately.

For example, in the data set shown in Fig. 1, it is impossible to detect the clusters A, B, C_1, C_2 and C_3 simultaneously using one global density parameter. A global density-based decomposition would be needed for the clusters A, B and C, or C_1, C_2 and C_3. In the second case, the objects from A and B may be noise.

There is also another problem existing in DBSCAN. As a matter of fact, in Fig. 2, data sets A and B may be in dissimilar clusters, but using a globally given radius ε may be clustered them into a similar cluster.

About the above first existing problem, the clustering algorithm OPTICS can resolve it completely. However, the algorithm presents a new drawback. It only generates the clusters whose local-density exceeds some threshold instead of similar local-density clusters and does not produce clusters of a data set explicitly.

How to perfectly figure out the problems described above in DBSCAN and OPTICS is a focus discussing issue in the density-based clustering algorithm. LD-BSCA, proposed in this study, is an efficient and effective algorithm to solve the problems existing in DBSCAN and OPTICS.

Basic concepts: In present studies, we choose DBSCAN algorithm as an investigative basis, because it has the ability in discovering clusters with arbitrary shape such as linear, concave, oval, etc. Furthermore, in contrast to others clustering algorithms, it does not require the predetermination of the number of clusters. DBSCAN has been proven capable for processing very large databases (Zhou and Zhou, 2000; Ester *et al*., 1998). To improve DBSCAN effectiveness, we define LD-BSCA similar to definition of DBSCAN. The formal definitions of clustering algorithm LD-BSCA are shortly introduced in the following:

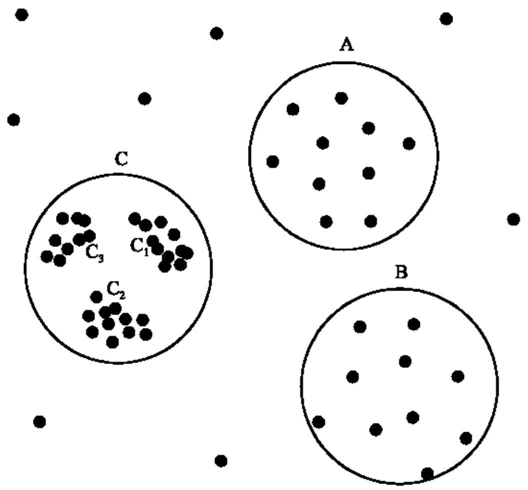

Fig. 1: An example spatial data set for DBSCAN

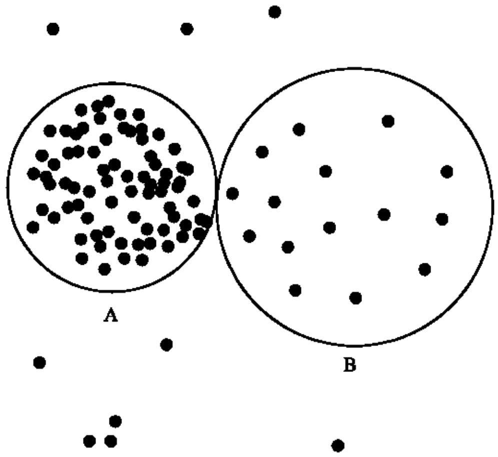

Fig. 2: Another example spatial data set for DBSCAN

Definition 1: (Core object): A core object is an unclassified point.

Any point which is not labeled can be selected as a core object. Given a radius, we can expand a new cluster around a core object. In the LD-BSCA algorithm, the global parameter MinPts is not defined and the local MinPts is variable parameter which changes as the core point selected disparately.

In this study, the definition of local MinPts is different from the definition by Ertöz *et al*. (2003) and Angiulli (2006). The local MinPts is changed while the core point is different.

Definition 2: (ε-neighborhood of a point): ε-neighborhood of a point p, denoted by $N_\varepsilon(p)$, is defined as $N_\varepsilon(p) = \{q \in D | dist(p, q) \le \varepsilon\}$.

Definition 3: (The number of ε-neighborhood of a point): The number of ε-neighborhood of a point p is represented as $CN_\varepsilon(p)$.

When a core point p is selected as a new clustering base point, the local MinPts which at least contains a local minimum number of points act as the lowest threshold of this cluster expanding. Apparently, the number of the ε-neighborhood of a core object p is denoted by $CN_{\varepsilon\text{-MinPtsClId}}(p)$ which is equivalent to local MinPts.

Definition 4: (Directly local-density-reachable): An object q is directly local-density-reachable from an object p if $q \in N_\varepsilon(p)$ and p is a core object.

Definition 5: (ExpandConClId): Let p be a core point of the ClId-th cluster C. The cluster C can be expanded if one of the objects q satisfies the following condition:

$$q \in \{q \mid q \in N_\varepsilon(p), q \ne p\} \tag{1}$$

$$CN_{\varepsilon\text{-MinPtsClid}}(p) \le CN_\varepsilon(q) \le CN_{\varepsilon\text{-MinPtsClid}}(p)(1+(CN_{\varepsilon\text{-MinPtsClid}}(p))^{-1/2}) \tag{2}$$

In the process of creating clusters, for each core point, it has a corresponding variable $CN_{\varepsilon\text{-MinPtsClid}}$ and its value should be set. Additionally, we should setup an upper limit for each cluster in order to determine when the cluster creating process will stop. The upper limit is also used to distinguish the cluster from other clusters with dissimilar density. Because there are different lower limits of different clusters, we define the upper limit for a cluster according to its lower threshold.

Definition 6: (Local-density-reachable): An object p is local-density-reachable from the object q with respect to ε and ExpandConClId if there is a chain of objects $p_1, \cdots, p_n$, $p_1 = p$ an $p_n = q$ such that p_{i+1} is directly density-reachable from p_i with respect to ε and ExpandConClId, for $1 \le i \le n$, $p_i \in D$ (Fig. 3a).

Definition 7: (Local-density-connected): An object p is local-density-connected to an object q with respect to ε and ExpandConClId if there is an object $o \in D$ such that both p and q are density-reachable from o with respect to ε and ExpandConClId.

It is obvious that local-density-connectivity is a symmetric relation as described earlier (Fig. 3b).

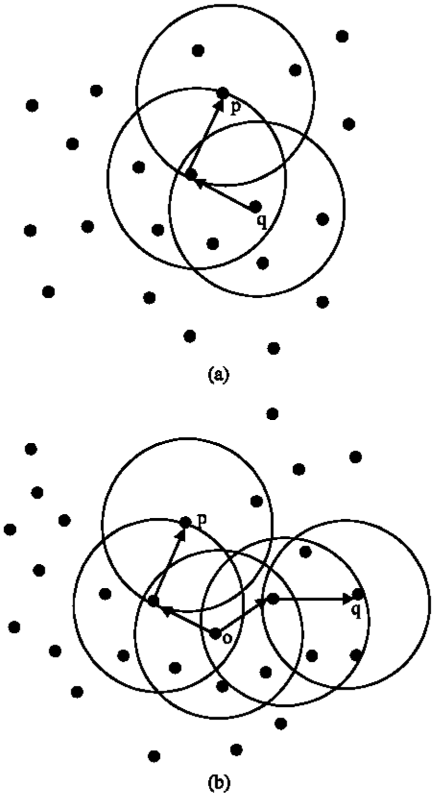

Fig. 3: (a) Local-density-reachable and (b) local-density-connectivity of object p

Definition 8: (Cluster): Let D be a database of points, a cluster C with respect to ε and ExpandConClId is a non-empty subset of D satisfying the following conditions:

- **Maximality condition:** ∀p, q: if $p \in C$ and q is local-density-reachable from p with respect to ε and ExpandConClId, then $q \in C$
- **Connectivity condition:** ∀p, $q \in C$: q is local-density-connected to p with respect to ε and ExpandConClId

Based on analyzing the searching method of DBSCAN, we can find that it need do area query, in other words, query every points in the ε-neighborhood of a point. Actually, many of these ε-neighbor of a point are overlapping in certain degrees. In order to reduce the number of area searching, we introduce a new searching method. The formal definitions for this notion of a clustering are shortly introduced in the following two lemmas.

Lemma 1: Let p be a point in database D. If p is a core point, then all the objects in the ε-neighborhood of p are directly local-density-reachable from the core point.

In other words, all the objects are in the same cluster as the core point p.

Lemma 2: Let p and q be two core points in database D and they satisfy the same cluster expanding conditions ExpandConClId. If some of their ε-neighborhoods superpose and there is a core point o satisfying the same cluster expanding conditions ExpandConClId as p and q in these overlapping regions, then the whole points in the ε-neighborhoods of p and q are local density-connected. According to this analysis, we obtain the knowledge that all the points in the ε-neighborhoods of p and q are in the same cluster.

LD-BSCA ALGORITHM

Here, a new clustering algorithm LD-BSCA relying on notions of clusters, local MinPts and ExpandConClId, are designed to discover the clusters in a spatial database. Many algorithms based DBSCAN need users input more than two parameters which may be difficult to determine and users who want to obtain reasonable clustering result may require testing large different initializations. Obviously, reducing input parameters is a very effective way to decrease the errors introduced by human interference. Using LD-BSCA algorithm, it only need one user input parameter, a global parameter radius ε.

Algorithm design: Suppose there is a database D includes n data objects. First, the LD-BSCA algorithm reset the given database to an increasing order by degrees in an appointed dimension. It begins with the smallest point p and retrieves all objects local-density-reachable from p with respect to ε and ExpandConClId. If p is unclassified, then it is a core point, then LD-BSCA creates a new cluster. According to definition 8, lemma 1 and 2, this algorithm works as following:

```
LD-BSCA (SetOfPoints,ε) // SetOfPoints is UNCLASSIFIED
Sort(SetOfPoints);
ClusterId := 0;
FOR i FROM 1 TO SetOfPoints.size DO
    Point := SetOfPoints.get(i);
    Seeds :=SetOfPoints.regionQuery(Point, ε);
    IF Point.ClId = UNCLASSIFIED THEN
      OldClusterId :=GetfirstcoreId(seeds);
      IF OldClusterId= UNCLASSIFIED THEN
            MinPtsClId :=Count(seeds);
//Defining the Local
//Minimum Number of Point
            ClusterId := ClusterId+1;
            SetOfPoints.changeClIds(seeds,ClId);
      ELSE
        ExpandCluster(SetOfPoints, seeds,
        OldClusterId, ε, MinPtsClId);
      END IF
    END IF
END FOR
ReOrganize(SetOfPoints);
END; // LD-BSCA
```

SetOfPoints is either the whole database or a discovered cluster from a earlier algorithm run. The parameter ε is a global density parameter determined manually by users. The function Sort() is used to sort database orderly in an appointed dimension. The function SetOfPoints.get(i) returns the i-th element of SetOfPoints. The function GetfirstcoreId() returns index minimum core point which is signed in a certain set or UNCLASSIFIED if all points are unsigned. The function Count() takes count of the local minimum value of a cluster based on the current core point. In addition, the intention of ReOrganize() is to coordinate the clustering result and clear up excrescent cluster signs. The function ExpandCluster() is particularly showing in the following:

```
ExpandCluster (SetOfPoints, seeds, OldClusterId, ε, MinPtsClId)
WHILE seeds < > Empty
  currentP :=seeds.first();
  result := SetOfPoints.regionQuery(currentP, ε);
    IF SetOfPoints IsCorePoint(result.size, MinPtsClId)
     THEN
     SetOfPoints MergeCluster(currentP,ClId);
   ELSE
     SetOfPoints.changeClId(resultP,ClId);
   END IF
  seeds.delete(currentP);
END WHILE
END;// ExpandCluster
```

The function IsCorePoint() judges whether a point is a core point or not (as described in algorithm 2). The action of function MergeCluster() is used to combine two or more subset into a cluster. The function delete() is used to remove the classified point from unclassified points set.

```
IsCorePoint(result.size, MinPtsClId) : Boolean
  IF result.size>= MinPtsClId and
result.size <= MinPtsClId +(MinptsClId)$^{-1/2}$
     THEN
      RETURN True;
ELSE
```

```
        RETURN False;
    END IF
END; // IsCorePoint
```

Dealing with border points: If two clusters C_1 and C_2 are very close to each other, a point o may belong to both C_1 and C_2. In this case, the point o must be a border point between C_1 and C_2. As the matter of fact, in the runtime of the algorithm LD-BSCA, it will come across the following scene. Let p as the core point of cluster C_1 and q as the core point of cluster C_2.

This algorithm performs expanding the cluster C_1, when $Cn_\varepsilon(q)<CN_{\varepsilon\text{-MinptsClId}}$ or $CN_\varepsilon(q)>CN_{\varepsilon\text{-MinptsClId}}(p)(1+(CN_{\varepsilon\text{-MinptsClId}}(p))^{-1/2})$, then start another cluster based on a core point q. In this process, there may be existent a point o which is clustered into C_1 and $o \in N_\varepsilon(q)$, then it also can be classified into C_2. The point o belongs to which cluster will be more appropriate? In this study, we should compare the value $CN_\varepsilon(o)$ with $CN_{\varepsilon\text{-MinptsClId}}(p)$ and $CN_{\varepsilon\text{-MinptsClId}}(q)$, finding which one the value $CN_\varepsilon(o)$ is more closed to, the point o, therefore, belongs to this cluster. All the border points can be managed in this way.

Parameter ε: ε is an important parameter of algorithm LD-BSCA. How to determine the global parameter radius ε is a crucial problem. If a user has enough knowledge or can empirically evaluate this parameter, he can input this parameter use traditional method as described by Ester *et al*. (1998). Otherwise, we should be cautious to set value to the parameter radius ε, because it directed influences the clustering result. Compared with DBSCAN which need confirm two parameters, the global parameter radius ε and the global minimum number of points (MinPts), LD-BSCA algorithm need confirm only one single parameter. If the object radius ε is too big, the clusters may not be distinguished in detail. In order to obtain more precise clusters, we can try several times. Taking into account a core point p, as the parameter ε enhances, the region acreage of $N_\varepsilon(p)$ is increasing in ε^2 series and the value $CN_{\varepsilon\text{-MinptsClId}}(p)$ is non-descending. In the dense region, this value increases very fast. But due to that $CN_{\varepsilon\text{-MinptsClId}}(p)$ is changed following ε, the change of ε can not influence our clustering result very much. In order to cluster more exact, we introduce a single method to confirm ε. For example, we can fetch the first value as 1 mm and choose a bigger value as 5 mm, then select the median between 1 and 5 (Fig. 4). With respect to these three input parameter values, we separately execute LD-BSCA algorithm at the base of definition 8 and lemma 1 and 2 to elicit three distinct clustering results. Observe these three results with intuitionistic graphs and compare these graphs to find which two are more similar, then fetch the median between these inputting parameter values as a new inputting radius. It does not stop this analogical process until an optimal clustering result is obtained.

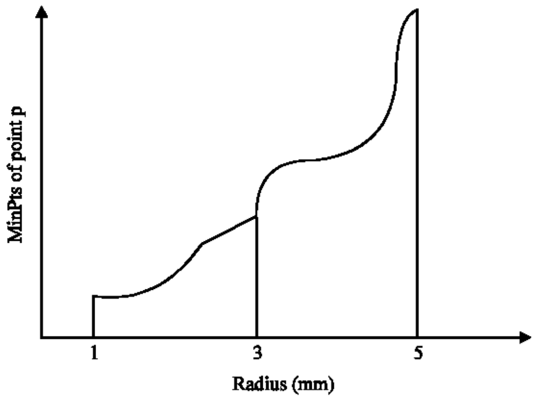

Fig. 4: Determine the valid value of ε

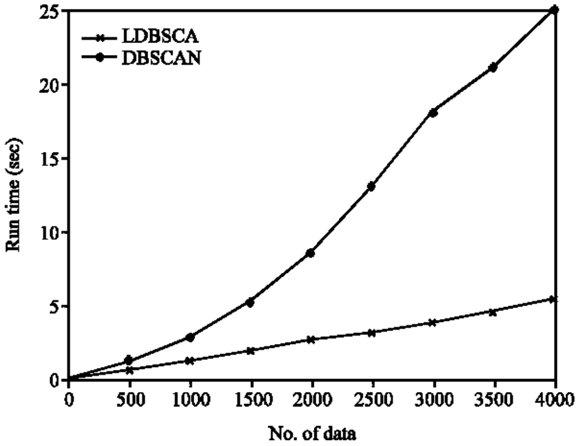

Fig. 5: The clustering runtimes of LD-BSCA and DBSCAN in different numbers of data

Time complexity analysis: The most of execution time of algorithm LD-BSCA is consumed when performing the function SetOfPoints.regionQuery(). In this experiment, the time consumed in other functions is ignored because it is much less than that of function SetOfPoints.regionQuery(). Using the house property database of Hangzhou city we carry out experiments with DBSCAN and LD-BSCA, respectively. The applicability of the proposed algorithm LD-BSCA is greatly comprehensive. For instance, we can use this algorithm to cluster the presold houses of a city in order to provide the concretely distributing status for the purchasers, or cluster the sold houses of a city with different price to provide land agents with some information. As shown in Fig. 5, the time complexity of LD-BSCA clustering in database D with n objects is much less than that of DBSCAN (O(n log n)). In general, the algorithm LD-BSCA is clustering several-fold faster than DBSCAN. Especially when the number of data becomes extremely great, the reduced runtime is much more and the advantage of LD-BSCA presents more evidently.

EXPERIMENTS

In order to compare with representative algorithm DBSCAN, we carried out some experiments to conduct a sensitivity analysis and evaluate our algorithm, especially, evaluate the effect of the regularization parameter ε. We elect different databases to prove the effectiveness of our algorithm.

Sample data 1: The experiments use a synthetic sample spatial database including 478 points. The spatial distribution of the dataset is shown in Fig. 6 visually.

The DBSCAN and LD-BSCA algorithms are implemented in Java based on the synthetic sample data in Fig. 6. All experiments have been conducted with a Pentium IV PC with 1G main memory and a 120 G fixed disk, running on Windows XP Professional.

To compare LD-BSCA with DBSCAN in effectiveness (accuracy), we apply two algorithms to the same database and input the same parameter radius ε. First, we give a test value to ε and prepare several different MinPts. Using these values, we carry out DBSCAN and LD-BSCA algorithm. In order to find out quantitative measure of the classification accuracy, results are denoted by visual inspection.

To test DBSCAN algorithm, we set the parameter radius ε to 2 mm and change another parameter MinPts from 3 to 15. With these parameters input, the results of DBSCAN clustering discovery are shown in Fig. 7. To show the results of both clustering algorithms, we visualize each cluster by different colors.

As shown in Fig. 7a, we can find that there are four clusters, labeled by color of red, blue, pink and orange paradise separately and a noise area smeared with gray color when ε = 2 and MinPts = 3. When MinPts is changed into 15, there are three clusters discovered by DBSCAN, labeled with color of red, forest green and aqua separately, while all other points labeled with grey color are noises in Fig. 7b. Apparently, DBSCAN is low effectiveness for clustering discovery of gradient distributing database. In Fig. 7a and b, there are some clusters lost in both clustering results. Compared to DBSCAN, the LD-BSCA algorithm discovers nine clusters. The result is shown in Fig. 8. Obviously, we can find that the cluster painted with fern green color can not be discovered no matter what value the parameter MinPts is set, but it can be found by LD-BSCA algorithm.

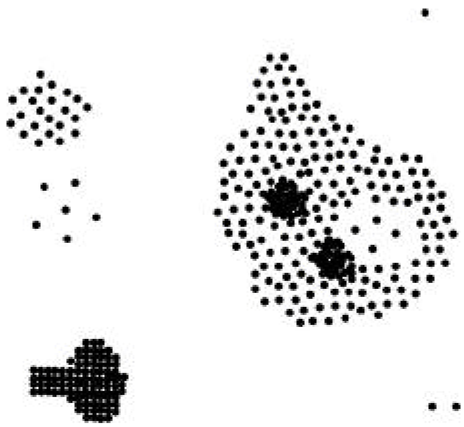

Fig. 6: Sample data 1

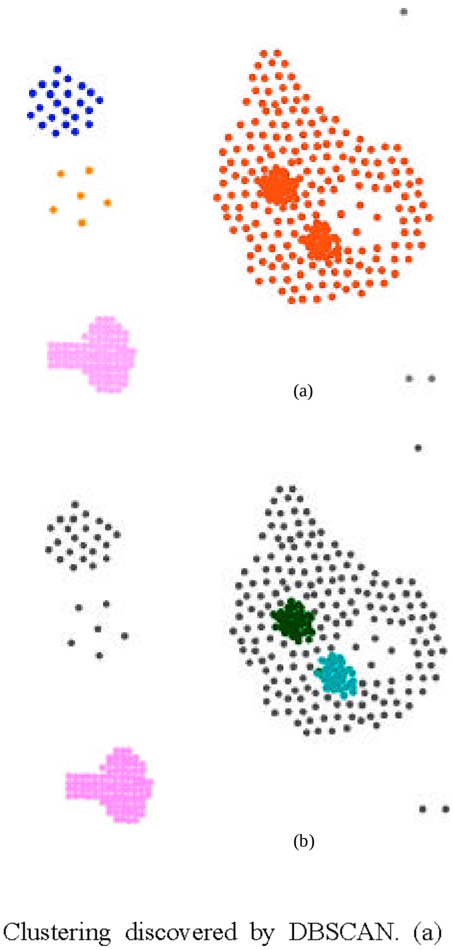

Fig. 7: Clustering discovered by DBSCAN. (a) ε = 2, MinPts = 3 and (b) ε = 2, MinPts = 15

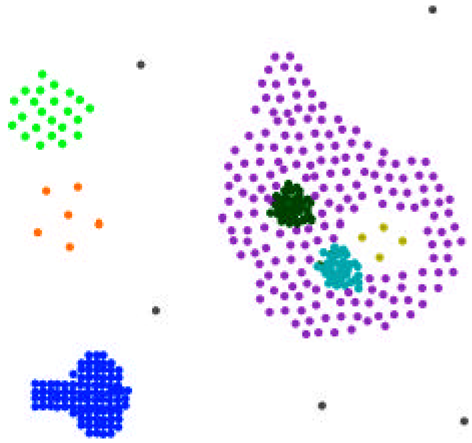

Fig. 8: Clustering discovered by LD-BSCA

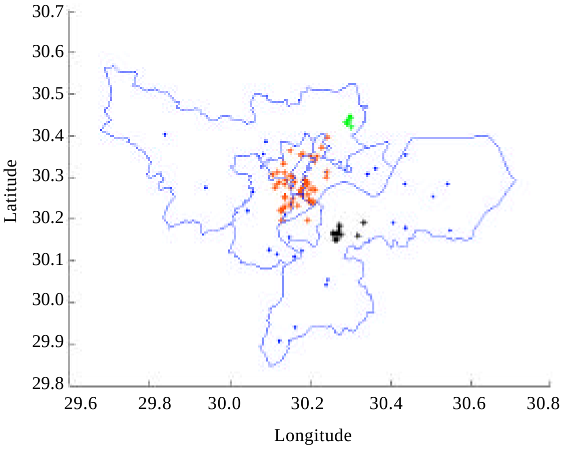

Fig. 10: DBSCAN discovers 3 clusters in park data (Scale 1:5000)

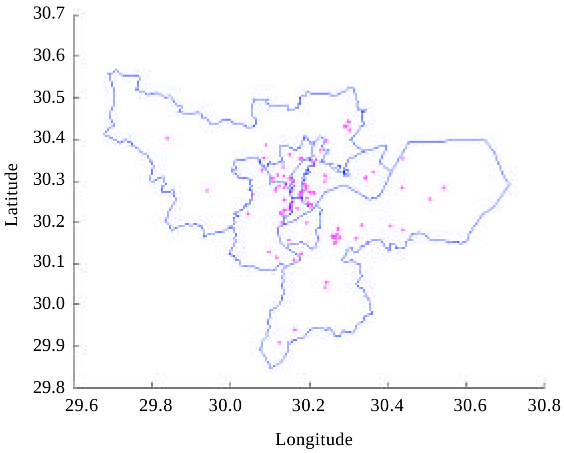

Fig. 9: The original Hang Zhou park data (Scale 1:5000)

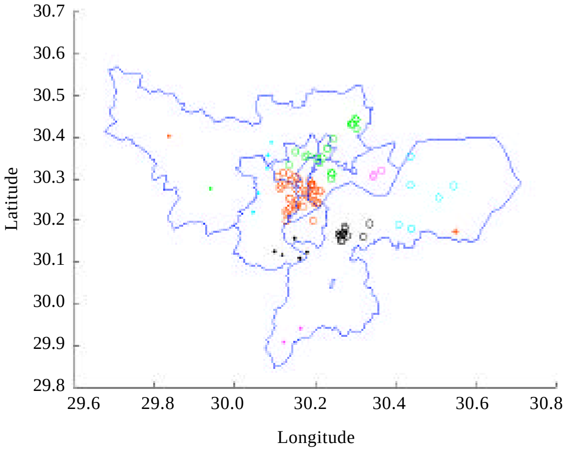

Fig. 11: LD-BSCA discovers 12 clusters in park data (Scale 1:5000)

Park data of hangzhou city: Here, we present experimental results from the park dataset of Hang Zhou which consists of almost all the real-world parks covering the whole Hang Zhou region. The original data is shown in Fig. 9. We can see that there are three clusters found by DBSCAN in Fig. 10. Different colors are used to signify the true clusters and the blue points represent noises. Note that, as expected, present algorithm can discover all twelve clusters (Fig. 11) without noise points excepted.

Comparing with OPTICS: In order to further test the effectiveness of the algorithm LD-BSCA, here we compare it with OPTICS on sample database 1 with 478 points as shown in Fig. 6. From Fig. 12, we can find that the main difference of OPTICS and LD-BSCA is the cluster by the red points. OPTICS discovers some overlap cluster, such as the cyan cluster E and the bottle green cluster F are folded to the cluster G containing all the cyan, bottle green and red points. Compared with Fig. 8, OPTICS cannot the olivine cluster H which LD-BSCA can find. Furthermore, LD-BSCA can obviously distinguish E, F and the red cluster, but OPTICS can't partition them in detail.

The results of this experiments show that present algorithm is more effective than the representative algorithm DBSCAN and OPTICS in clustering discovery of gradient distributing database.

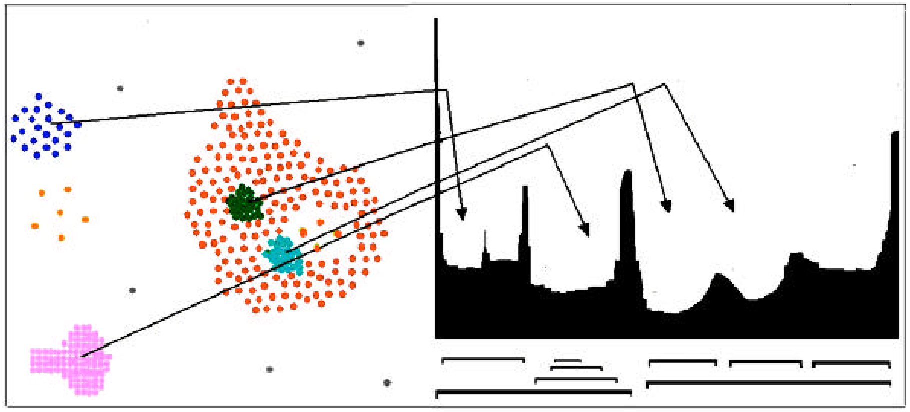

Fig. 12: Reachability-plots of OPTICS

CONCLUSION

Clustering is a main method in many fields, including data mining and knowledge discovery, statistics and machine learning. Among most clustering methods, density-based clustering algorithm is one of powerful tools for discovering arbitrary-shaped clusters in large spatial databases. In this research, present study presents a new density-based clustering algorithm LD-BSCA which is constructed by improving DBSCAN algorithm. By introducing a new concept local MinPts and a new cluster expanding condition ExpandConClId, the LD-BSCA algorithm only requires user input one parameter (radius ε) and supports user determining an appropriate value for it. Experiments results demonstrate that LD-BSCA algorithm is more effective and more powerful to discover all clusters than DBSCAN algorithm for mining gradient distributing spatial database.

REFERENCES

Agrawal, R., J. Gehrke, D. Gunopulos and P. Raghavan, 1998. Automatic subspace clustering of high dimensional data for data mining applications. Proceedings of ACM SIGMOD International Conference on Management of Data, Jun. 1-4, Seattle, WA, ACM Press, New York, pp: 94-105.

Angiulli, F., 2006. Clustering by exceptions. Proceedings of the 21st National Conference on Artificial Intelligence, Jul. 16-20, Boston, Massachusetts, AAAI Press, pp: 312-317.

Ankerst, M., M.M. Breunig, H.P. Kriegel and J. Sander, 1999. OPTICS: Ordering points to identify the clustering structure. Proceedings of ACM SIGMOD International Conference on Management of Data, May 31-Jun. 03, Philadelphia, PA., ACM Press, pp: 49-60.

Chehreghani, M.H., H. Abolhassani and M.H. Chehreghani, 2008. Improving density-based methods for hierarchical clustering of web pages. Data Knowl. Eng., 67: 30-50.

Ding, W. and J. Yuan, 2008. A robust fuzzy clustering method based on local density for automatic spike sorting. Exp. Syst. Appli. 10.1016/j.eswa.2008.01.023

Duan, L., L. Xu and F. Guo, 2007. A local-density based spatial clustering algorithm with noise. Inform. Syst., 32: 978-986.

Ertöz, L., M. Steinbach and V. Kumar, 2003. Finding clusters of different sizes, shapes and densities in noisy, high dimensional data. Proceedings of 3rd SIAM International Conference on Data Mining, May 1-3, San Francisco, pp: 47-58.

Ester, M., H.P. Kriegel, J. Sander and X. Xu, 1998. Clustering for mining in large spatial databases. KI-J., 1: 18-24.

Han, J. and M. Kamber, 2006. Data Mining: Concepts and Techniques. 2nd Edn., Morgan Kaufmann Publisher, San Fransisco, USA., ISBN: 1-55860-901-6.

Hinneburg, A. and D.A. Keim, 1998. An efficient approach to clustering in large multimedia databases with noise. Proceedings of 4rth International Conference on Knowledge Discovery and Data Mining, Aug. 27-31, New York, pp: 58-65.

Jain, A.K. and R.C. Dubes, 1998. Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs, New Jersey, ISBN: 013022278X.

Jim, Z.C. and L.Y. Liaw, 2008. Improvement of the k-means clustering filtering algorithm. Pattern Recogn., 41: 3677-3681.

Kaufman, L. and P.J. Rousseeuw, 2005. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, ISBN: 9780471735786.

Li, H. and L. Xu, 2001. Feature space theory-a mathematical foundation for data mining. Knowl. Based Sys, 14: 253-257.

Li, H., L. Xu, J. Wang and Z. Mo, 2003. Feature space theory in data mining: transformations between extensions and intensions in knowledge representation. Expert Sys, 20: 60-71.

Ma, S., T.J. Wang and S.W. Tang, 2003. A New Fast Clustering Algorithm Based on Reference and Density. In: Lectures Notes in Computer Science, Dong, G. (Ed.). Springer-Verlag, Berlin Heidelberg, pp: 214-225.

Ma, W.M., C. Eden and W.S. Tommy, 2004. A new shifting grid clustering algorithm. Pattern Recogn., 37: 503-514.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Jan. 17-20, Berkeley, CA., pp: 281-297.

Pilevar, A.H. and M. Sukumar, 2005. GCHL: A grid-clustering algorithm for high-dimensional very large spatial data bases. Pattern Recogn. Lett., 26: 999-1010.

Rezaie, A.H., S.H.H. Sadeghi, M.H. Moradi and M. Ahamadi, 2007. A density-based fuzzy clustering technique for non-destructive detection of defects in materials. NDT E Int., 40: 337-346.

Sheikholeslami, G., S. Chatterjee and A. Zhang, 1988. WaveCluster: A multi-resolution clustering approach for very large spatial databases. Proceedings of 24th International Conference on Very Large Data Bases, Aug. 29-Sept. 1, New York, pp: 428-439.

Sheikholeslami, G., S. Chatterjee and A. Zhang, 2000. Wavecluster: A wavelet-based clustering approach for spatial data in very large databases. The VLDB J., 8: 289-304.

Vinod, H., 1969. Integer programming and the theory of grouping. J. Am. Stat. Assoc., 64: 506-519.

Wang, W., J. Yang and R. Muntz, 1997. STING: A statistical information grid approach to spatial data mining. Proceedings of the International Conference on Very Large Data Bases, Aug. 25-29, Athens, Greece, pp: 86-195.

Zhang, T., R. Ramakrishnan and M. Livny, 1996. BIRCH: An efficient data clustering method for very large data bases. Proceedings of the ACM SIGMOD International Conference on Management of Data, Jun. 4-6, Cananda, pp: 103-114.

Zhou, A.Y. and S. Zhou, 2000. Approaches for scaling DBSCAN algorithm to large spatial database. J. Comput. Sci. Technol., 15: 509-526.