J. Software Engineering & Applications, **2010, 3: 141-149**
doi:10.4236/jsea.2010.32018 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

A Novel Spatial Clustering Algorithm Based on Delaunay Triangulation

Xiankun Yang[1,2], Weihong Cui[1]

[1]Institute of Remote Sensing Application, Chinese Academy of Sciences, Beijing, China; [2]Graduate University of Chinese Academy of Sciences, Beijing, China.
Email: xiankungis@163.com.

Received August 17th, 2009; revised September 16th, 2009; accepted November 13th, 2009.

ABSTRACT

Exploratory data analysis is increasingly more necessary as larger spatial data is managed in electro-magnetic media. Spatial clustering is one of the very important spatial data mining techniques which is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. So far, a lot of spatial clustering algorithms have been proposed in many applications such as pattern recognition, data analysis, and image processing and so forth. However most of the well-known clustering algorithms have some drawbacks which will be presented later when applied in large spatial databases. To overcome these limitations, in this paper we propose a robust spatial clustering algorithm named NSCABDT (*Novel Spatial Clustering Algorithm Based on Delaunay Triangulation*)*. Delaunay diagram is used for determining neighborhoods based on the neighborhood notion, spatial association rules and collocations being defined. NSCABDT demonstrates several important advantages over the previous works. Firstly, it even discovers arbitrary shape of cluster distribution. Secondly, in order to execute NSCABDT, we do not need to know any priori nature of distribution. Third, like DBSCAN, Experiments show that NSCABDT does not require so much CPU processing time. Finally it handles efficiently outliers.*

Keywords: *Spatial Data Mining, Delaunay Triangulation, Spatial Clustering*

1. Introduction

Data mining is a process to extract implicit, nontrivial, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases [1,2]. The explosive growth in data and databases used in business managements, government administration, and scientific data analysis has created a need for tools that can automatically transform the processed data into useful information and knowledge [3]. Spatial data mining as a subfield of data mining refers to the extraction from spatial databases of implicit knowledge, spatial relations or significant features or patterns that are not explicitly stored in spatial databases [4]. It is concerned with the discovery of spatial relationships and intrinsic relationships between spatial and non-spatial data. With the large amount of spatial data obtained from satellite images and geographic information systems (GIS), it is an inevitable task for humans to explore spatial data in detail. Spatial datasets and patterns are abundant in many application domains related to the Environmental Protection Agency, the National Institute of standards and Technology, and the Department of Transportation. Challenges in spatial data mining arise from the following issues [3,5]. Firstly, classical data mining is designed to process numbers and categories. In contrast, spatial data is more complex and includes extended objects such as points, lines and polygons. Secondly, classical data mining works with explicit inputs, whereas, spatial predicates and attributes are often implicit. Finally, classical data mining treats each input independently of other inputs, while spatial patterns often exhibit continuity and high autocorrelation among nearby features.

Clustering is the process of grouping a set of objects into classes or clusters so that objects within a cluster have similarity in comparison to one another, but are dissimilar to objects in other clusters. So far, many clustering algorithms have been proposed. They differ in their capabilities, applicability and computational requirements. Based on a general definition, they can be categorized into five broad categories, i.e., hierarchical, partitional, density-based, grid-based and model-based [4]. 1) Partitional clustering methods [6], for example, CLARANS. It classifies data into some groups, which together satisfy the following requirements: firstly, each group must contain at least one object; secondly, each object must belong to exactly on group. It is noticed that the second requirement can be relaxed in some fuzzy partitioning techniques. 2) Hierarchical clustering methods [7,8], such as DIANA [9] and BIRCH [8]. A hierarchical method creates a hierarchical decomposition of a given set of data objects. Hierarchical methods can be classified as agglomerative (bottom-up) or divisive (top-down), based on how the hierarchical decomposition is formed. 3) Density-based clustering methods. Their general idea is to continue growing a given cluster as long as the density (the number of objects or data points) in the "neighborhood" exceeds a threshold. Such a method is able to filter out noises (outliers) and discover clusters of arbitrary shape. Examples of density-based clustering methods include DBSCAN [10], OPTICS [11], GDBSCAN [12] and DBRS [13]. 4) Grid-based clustering methods, such as STING [14] and WaveCluster [15]. Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. 5) Model-based clustering methods. For example, COBWEB. It is clearly that no particular clustering method has been shown to be superior to all its competitors in all aspects. Typically, the problem is that clusters identified with one method cannot be detected by other methods [16]. This is because that many clustering methods need user-specified arguments or prior knowledge to produce their best results. Such information needs are supplied as density threshold values, merge/ split conditions, number of parts, prior probabilities, assumptions about the distribution of continuous attributes within classes, and/or kernel size for intensity testing, for example, grid size for raster-like clustering [17] and radius for vector-like clustering [18]. This parameter-tuning is expensive and inefficient for huge data sets because it demands several trial and error steps.

Clustering based on Delaunay triangulation is not a new and has been described in some papers [16, 19, 20, 21]. Kang *et al* [14] proposed a clustering algorithm that utilizes a Delaunay triangulation; however, there is a need in the algorithm to provide a global argument as a threshold to discriminate perimeter values or edges lengths. As a result, the algorithm is not able to detect local variations. The first non-parametric clustering algorithm based on the Delaunay diagram, called AMOEBA, has been proposed in Estivill-Castro and Lee [16]. It overcomes some of the problems of the static approaches that required a distance threshold to be specified, but still fails to find relatively sparse clusters in certain situations. An upgraded version of AMOEBA, called AUTOCLUST, has been proposed by the same authors in Estivill-Castro and Lee [21].

But these methods also have some drawbacks. For example, if two clusters are mixed or connected by bridges, this methods described above cannot detect all the two clusters as shown in Figure 1. In this paper we propose a robust spatial clustering algorithm named NSCABDT (Novel Spatial Clustering Algorithm Based on Delaunay Triangulation). NSCABDT uses the Delaunay triangulation as analysis source, because Delaunay triangulation is a structure that is linear in the size of the data set and implicitly contains vast amounts of proximity information. That is to say, we can use the graph information of Delaunay triangulation and metric information to obtain remarkable robust clustering. In this study, we first construct a graph, and record the information of the graph as presented in Section 3. In the graph, vertices represent data points and edges connect pairs of points to model spatial proximity or interactions and all clustering operations are performed on the graph information.

The remainder of the paper is organized as follows: In Section 2, we will give an introduction to data preprocessing for NSCABDT. And Section 3 presents the NSCABDT algorithm. Section 4 reports the experimental evaluation. Finally, Section 5 concludes the paper.

2. Definitions and Notions

2.1 Spatial Clustering methods

Geographic data often show properties of spatial dependency and spatial heterogeneity [22]. Spatial dependency is a tendency of observations located close to one another in the geographical space to show a higher degree of similarity or dissimilarity (depending on the phenomenon under study). Closeness can be defined very generally—through distance, direction and/or topology. Spatial heterogeneity or inconsistency of the process with respect to its location is often visible, while many geographic

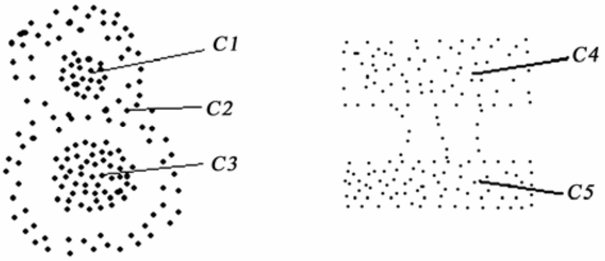

Figure 1. Three very dense clusters (*C*1, *C*2, *C*3), but most of clustering methods cannot detect the cluster *C*2. Because of the bridges between cluster *C*4 and cluster *C*5, the two clusters often are incorrectly thought to be one cluster

processes have a local character. Spatial dependency and heterogeneity can reflect the nature of the geographic process. Central to spatial data mining is clustering, which seeks to identify subsets of the data having similar characteristics. Two-Dimensional clustering is the non-trivial process of grouping geographically closer points into the cluster. Therefore, a model of spatial proximity for a discrete point-data set $P = \{p_1, \cdots, p_n\}$ must provide robust answers to which are the neighbors of a point p_i and how far the neighbors relative to the context of the entire data set P. A cluster is a group of objects, which are homogeneous among themselves. Clustering has been identified as one of the fundamental problems in the area of knowledge discovery and data mining, and it is of particular importance for spatial data sets. A distinct characteristic of spatial clustering for data mining applications is the huge size of the data files involved [23]. As Tobler's famous proposition [24] states: "Everything is related to everything else, but near things are more related than distant things." Thus proximity is pretty critical to spatial analysis and in spatial settings; clustering criteria almost invariably makes use of some notions of proximity, usually based on the Euclidean metric, as it captures the essence of spatial autocorrelation and spatial association [14].

$$d(X_j, Z_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \quad (1)$$

$d(X_j, Z_j)$is the Euclidean metric. And $(x_{i1}, x_{i2}, \cdots, x_{ip})$ and $(x_{j1}, x_{j2}, \cdots, x_{jp})$ are two $p(p \geq 2)$ dimensions data objects.

We assume that $S = \{p_0, p_1, p_2, \cdots p_{n-1}\}$ is a set of n data items in the $m-$ dimensional real space $\Re^m$. A cluster is a collection of S that is similar to one another within the same cluster and is dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group. We assume that C_k is a cluster of S and $C = \{C_1, C_2, \cdots, C_k\}$ is the collection of $C_i (1 \leq i \leq k)$, then:

$$S = \bigcup_{j=i}^{p} C_j \quad (2)$$

$$C_j \neq \emptyset \text{（} j = 1,2,...,k \text{）} \quad (3)$$

$$C_i \bigcap C_j = \emptyset \qquad \text{（} i, j = 1,2,..., j; i \neq j \text{）} \quad (4)$$

If Z_j is the clustering center of $C_j (1 \leq j \leq k)$, then:

$$J = \sum_{j=1}^{k} \sum_{s_j \in C_j} d(p_j, Z_j) \quad (5)$$

where J should be minimal.

2.2 Delaunay Triangulations

In mathematics, and computational geometry, a Delaunay triangulation for a set S of points in the plane is a triangulation $D(S)$ such that no point in S is inside the circumcircle of any triangle in $D(S)$. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles. The triangulation was invented by Boris Delaunay [25]. Delaunay triangulations have been widely used in a variety of applications in geographical information systems (GIS). Using Delaunay triangulations, it is simpler to tackle the problems associated with spatial topology automated contouring, two-and-half dimensional (2.5-D) visualization, surface characterization and reconstructions, and site visibility analyses on terrain surfaces.

Given the set of data points $S = \{p_0, p_1, \cdots, p_{n-1}\}$ in the plane, the Voronoi region of $p_i \in S$ is the locus of points (not necessarily data points) which have p_i as a nearest neighbor; that is, $\{x \in \Re^2 \mid \forall j \neq i, d(x, p_i) \leq d(x, p_j)\}$. Taken together, the n Voronoi regions of S form the Voronoi diagram of $\boldsymbol{S}$ (also called the Dirichlet tessellation or the proximity map). The regions are (possibly unbounded) convex polygons, and their interiors are disjoint [23]. Based on Delaunay's definition [25], the circumcircle of a triangle formed by three points from the original point set is empty if it does not contain vertices other than the three that define it (other points are permitted only on the very perimeter, not inside).

The Delaunay triangulation $D(S)$ of S is a planar graph embedding defined as follows: the nodes of $D(S)$ consist of the data points of S, and two nodes p_i, p_j are joined by an edge if the boundaries of the corresponding Voronoi regions share a line segment.

Delaunay triangulations capture in a very compact form the proximity relationships among the points of S. They have many useful properties, the most relevant to our application being the following:

1) If p_j is the nearest neighbor of p_j from among the data points of S, then $< p_i, p_j >$ is an edge in $D(S)$. That is to say, the 1-nearest-neighbor digraph is a subgraph of the Delaunay triangulation.

2) The number of edges in $D(S)$ is at most 3n -6.

3) The average number of neighbors of a site s_i in $D(S)$ is less than six.

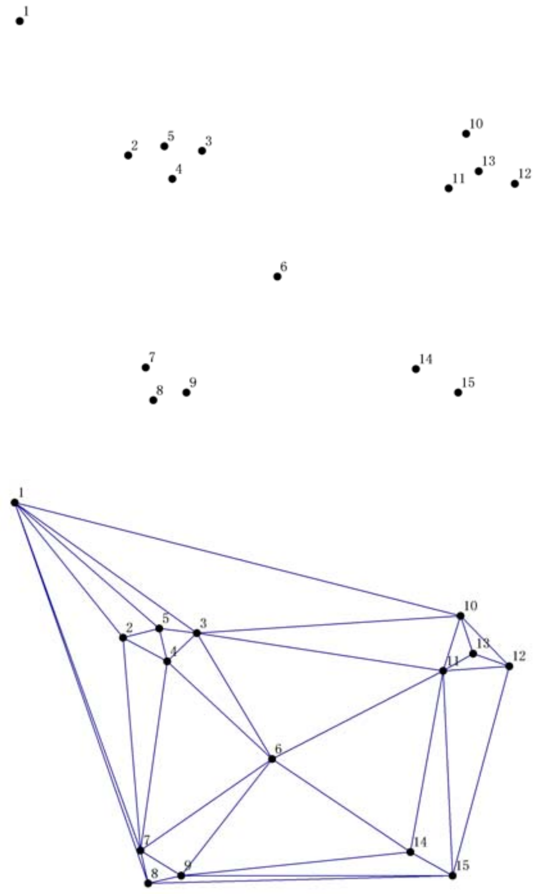

Figure 2. A data set (n=15) and its Delaunay triangulation

4) The Delaunay triangulation is the most well proportioned over all triangulations of *S*, in that the size of the minimum angle over all its triangles is the maximum possible.

5) If p_i and p_j form a triangle in *D*(*S*), then the interior of this triangle contains no other point of S.

6) The triangulation *D*(*S*) can be robustly computed in $O(n\log n)$ time.

7) The minimum spanning tree is a subgraph of the Delaunay triangulation, and, in fact, a single-linkage clustering (or dendrogram) can be found in $O(n\log n)$ time from *D*(*S*).

Figure 2 shows a set of 15 data points and its corresponding Delaunay triangulation. More information regarding Delaunay triangulations and Voronoi diagrams can be found in other literature. From Figure 2, we can conclude that, in a proximity graph like Delaunay triangulation (Delaunay diagram); the points are connected by edges, if and only if they seem to be close by some proximity measure [26]. By applying to this rule, if two points are connected by a small enough Delaunay edge, the two points belong to the same cluster.

3. Initialization Using the Delaunay Triangulations

3.1 Data Preprocessing

Given a set of data points $S=\{p_0,p_1,\cdots,p_{n-1}\}$ in the plane (as shown in Figure 2), $n=15$ .The triangulations were computed by Bowyer-Watson algorithm in $O(n\log n)$ time. In the creation process of Delaunay triangulation, we recorded node, edge and surface information of Delaunay triangulation for clustering later. This nodes, edges and surfaces information was stored in Oracle database. Oracle database includes numerous data structures to improve the speed of SQL queries. Taking advantage of the low cost of disk storage, Oracle includes many new indexing algorithms that dramatically increase the speed with which Oracle queries are serviced. And, Oracle database includes so many statistical functions which include descriptive statistics, hypothesis testing, and correlations analysis, for distribution fit and so forth. The statistical functions in the database can be used in a variety of ways, for example, we can call Oracle's DBMS_STAT_FUNCS functions to obtain basic cont, mean, max, min and standard deviation information of Delaunay triangulation edges. For Figure 2, we got three tables as follows:

In the Table 1, the first column is the index of the points in S, the second column is X coordinate and the third column is Y coordinate respectively. The degree denotes the number of Delaunay edges which incident to a point. The "ClassType" column represents the category number after clustering process, and after clustering process if it is -1, we think the point is an outlier or noise.

In the Table 2, the second column is the index of the edge's starting point, and the third column is the index of the edge's end point. The length of edges is represented by the fourth column. In our algorithm, every edge is needed to be computed only once.

The chart illustrates the table structure and relationships of the three tables. The Delaunay triangulation node table contains all the spatial objects (points); the

Table 1. Delaunay triangulation nodes table

Index	X	Y	Degree	ClassType
1	3853964.924	-803305.9261	6	-1
2	3853985.696	-803331.6837	4	-1
3	3853998.714	-803330.2989	6	-1
4	3853994.005	-803335.8381	5	-1
5	3853992.066	-803329.7449	4	-1
6	3854013.393	-803354.3946	6	-1
7	3853989.297	-803371.0124	6	-1
8	3853990.128	-803377.6595	4	-1
9	3853996.221	-803375.7208	5	-1
10	3854049.121	-803327.2523	5	-1
11	3854045.52	-803337.4999	7	-1
12	3854058.538	-803336.669	4	-1
13	3854051.337	-803334.4533	3	-1
14	3854039.981	-803371.8433	4	-1
15	3854047.459	-803375.7208	4	-1

Table 2. Delaunay triangulation edge table

Index	Start	End	Length
1	1	8	76.03228669
2	8	7	6.69884313
3	7	1	69.50014083
4	1	7	69.50014083
5	7	2	39.49321264
6	2	1	33.08972562
7	1	2	33.08972562
8	2	5	6.65851675
9	5	1	36.11126413
10	1	5	36.11126413
……	……	……	……

Table 3. Delaunay triangulation surface table

Index	Node1	Node2	Node3	Edge1	Edge2	Edge3
1	1	8	7	1	2	3
2	1	7	2	4	5	6
……	……	……	……	……	……	……

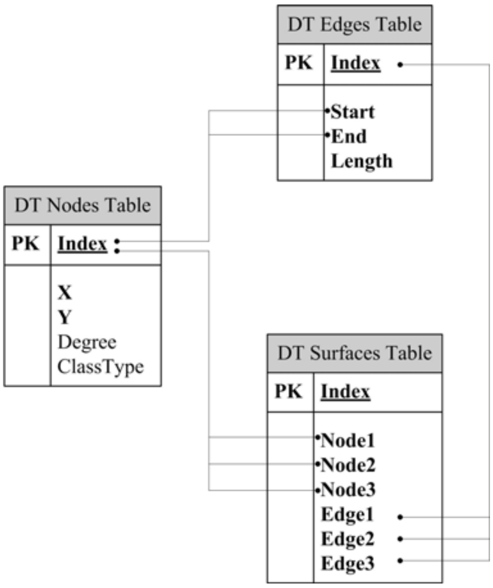

Figure 3. Table structure and relations in the database

Delaunay triangulation edge table includes all the Delaunay edges and the relationships with the Delaunay triangulation node table. And the Delaunay triangulation surface table records all the Delaunay triangulation surfaces and the relationships with Delaunay triangulation nodes table as well as Delaunay edge table.

3.2 Some Definitions and Notions in NSCABDT

Given a set of data points $S = \{p_0, p_1, \cdots, p_{n-1}\}$ in the plane (as shown in Figure 2), after data preprocessing, we got the nodes, edges and surfaces information of the Delaunay triangulation. Given a set of edges $E = \{e_0, e_1, \cdots, e_{n-1}\}$, for each edge $e_k (0 \le k \le n-1)$ in E is a record of Delaunay triangulation edge table. For each edge $e_k < p_i, p_j >$, $(0 \le k \le n-1)$ p_i is its starting point, and p_j is its end point. Both p_i and p_j belong to S. And $N(p_i)$ denotes a set of edges which incident to p_i.

Definition 1 (Local_Mean): We denote by mean (p_i) the mean length of edges in $N(p_i)$.

$$Local\_Mean(p_i) = \sum_{j=1}^{d(p_i)} Len(e_j) \Big/ d(p_i) \quad (6)$$

where $d(p_i)$ denotes the degree of p_i in graph theory; and $Len(e_j)$ denotes the length of the Delaunay edge e_j.

Definition 2 (Global_Mean): We denote by mean S the mean length of edges in E.

$$Global\_Mean(S) = \sum_{j=1}^{sum(E)} Len(e_j) \Big/ sum(E) \quad (7)$$

where $sum(E)$ is the number of edges in E.

Definition 3 (Global_Sta_Dev): We denote by global standard deviation of the lengths of all edges. That is,

$$Global\_Sta\_Dev(S) = \sqrt{\sum_{i=0}^{n} (Global\_Mean(S) - Len(e_i))^2 \Big/ n} \quad (8)$$

Definition 4 (Relative_Mean): We let $\mathrm{Re}lative\_Mean(p_i)$ denote the ratio of $Local\_Mean(p_i)$ and $Global\_Mean(S)$. That is,

$$\mathrm{Re}lative\_Mean(p_i) = Local\_Mean(p_i) / Global\_Mean(S) \quad (9)$$

Definition 5 (Positive Edge): If the length of a Delaunay edge is less than the given criterion function $F(p_i)$, the edge is a positive edge. Positive edges and points incident to them form a new proximity graph and the newly created graph is subgraph of the Delaunay graph (Delaunay Triangulation).

Definition 6 (Positive path): A path in current proximity graph where every edge in the path is a positive edge; and all the points connected by actives paths belong to a cluster.

Finally, this edge analysis is captured in a criterion function $F(p_i)$. The cut-off value for edges incident in p_i is defined as follows:

$$\begin{aligned} F(p_i) &= Global\_Mean(S) + Global\_Sta\_Dev(S) \\ &\times \mathrm{Re}lative\_Mean(p_i)^{-1} \\ &= Global\_Mean(S) + Global\_Sta\_Dev(S) \\ &\times \frac{Global\_Mean(S)}{Local\_Mean(p_i)} \end{aligned} \quad (10)$$

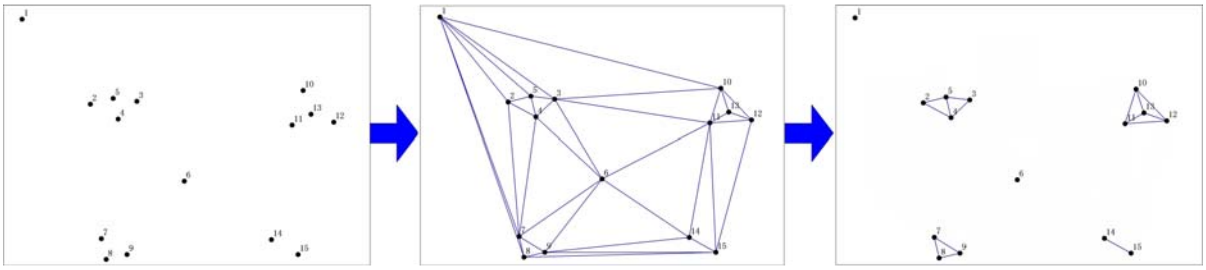

Figure 4. NSCABDT procedure

Definition 7 (Effective Region): For a point p in S, we call

$$p_\delta = \{x \mid |p - x| < \delta, p \in S, \forall x \in R^2\} \qquad (11)$$

The effective region of point p with respect to radius δ. As the union of the effective region of each point, we define the effective region of a random point set.
For a point set S, we call

$$\bar{S} = \bigcup_{p \in S} p_\delta \qquad (12)$$

The effective region of point set S with respect to radius δ.

Definition 8 (γ – *boundary*): We define the boundary set as

$$V_\gamma = \overline{V_\gamma} \cap V, \quad \overline{V_\gamma} = \overline{V} \setminus \{\overline{V} \Theta \gamma Dm(\delta)\} \qquad (13)$$

where $\gamma > 1$ is a constant and $Dm = \{x \mid |x| \le \delta\}$ is the set of all points in the circle with the radius δ. We call V_γ the γ – *boundary* of the point set *S*.

Definition 9 (γ – *curve*): The principal boundary of a random point set is the principal manifold of the point in the γ – *boundary* of a point set.

We also call this principal manifold extracted from point set S the γ – *curve* of S.

If the edges which incident to p_i are greater than or equal to $F(p_i)$, the edges are eliminated and the edges that are less than the criterion function survive. For each definition above, it is not necessary to iteratively calculate the results by programming; because we can get the results by using oracle statistical functions. For example, for Definition 1, a SQL statement can be created as follows: *SELECT avg(length) FROM edgetable WHERE start* = p_i, where “edgetable” is the table which restores the edges information of Delaunay triangulation.

We now present the algorithm of NSCABDT:

Initialize the points of a data points set S as being assigned to no cluster; Initialize an empty data set C;

1) Create Delaunay triangulation and record the information of Delaunay triangulation in Oracle database.
2) For each node p_i in Delaunay triangulation, extract edges $N(p_i)$ incident to node p_i via SQL queries and calculate $Local\_Mean(p_i)$ as well as $F(p_i)$.
3) For each edge e in $N(p_i)$, if $Len(e) \ge F(p_i)$, the edge will be deleted.
4) After 3, if $d(p_i) = 0$, the node p_i will be deleted, otherwise, the node p_i is added to C.
5) Using the same method, iteratively calculate all the nodes which connect with p_i.
6) Extract the boundary of the cluster C and eliminate the bridges.
7) If all the points have been not processed, end the process. Otherwise, initialize a new empty data set C, go to next un-processed node.

Phase 1 of NSCABDT is the construction of Delaunay triangulation. Then, recursively, all points in a connected component are reported as a cluster. Thus every edge is tested for the criterion function only once. After eliminating no-interesting edges and noises, only positive edges are remaining. According to the positive path, we can iteratively find all the points connected by positive paths and add the points to a cluster.

Obviously, it can be seen from the Figure 4 that it consists of two phases; the first phase is building Delaunay Triangulations from spatial objects. And on the second phase, we eliminate all edges in the way which we introduced above. And then, we got that point 1, point 6, point 14 and point 15 are outliers.

In order to eliminate the bridges between two different clusters, a detection of cluster boundary is executed; the algorithm is according to [5]. The boundary of a point set is extracted by the principal curve analysis. The principal curve analysis is a generalization of principal axis analysis, which is a standard method for data analysis in pattern recognition. For a cluster, if we can get two different boundaries, we think there are two smaller clusters in the point set, and bridges exist between the two smaller clusters. If an edge is not in the boundaries, the edge should be deleted.

The algorithm for eliminating the bridges between two different clusters is as follows:

1) For the collection of all edges E, get the median length via SQL queries. And set it as δ.
2) Get the effective region of random point set V.
3) Get $\gamma - boundary$ of random point set V.
4) Get $\gamma - curve$ of random point set V.

Although, the construction of Delaunay triangulation using all points in S is a time-consuming process for a large number of points even if we use an optimal algorithm, we can use the information stored in the database instead of the construction of Delaunay triangulation again and we also can get the median length via SQL queries. Obviously, it is more efficient.

4. Experimental Results

We evaluate NSCABDT according the three major requirements for clustering algorithms on large spatial databases as stated above. We compare NSCABDT with the clustering algorithm DBSCAN in terms of effectivity and efficiency. The evaluation is based on an implementation of NSCABDT in .NET 2005. All the experiments were run on Windows Server 2003.

4.1 Discovery of Clusters with Arbitrary Shape

Clusters in spatial databases may be of arbitrary shape, e.g. spherical, drawn-out, linear, elongated etc. Furthermore, the databases may contain noise [27]. We used visualization to evaluate the quality of the clusterings obtained by the NSCABDT. In order to create readable visualizations without using color, in these experiments we used small databases. Due to space limitation, we only present the results from one typical database which was generated as follows:

1) Draw three polygons of different shape (one of them with a hole) for three clusters.
2) Generate 500, 200 and 200 uniformly distributed points in each polygon respectively.
3) Insert 100 noise points into the database, which is depicted in Figure 5.

For NSCABDT, we set 10% noise for the sample database. NSCABDT discovers all clusters and detects the noise points from the sample database. The clustering result of NSCABDT on this database is shown in Figure 7. Different clusters are depicted using different symbols and noise is represented by crosses. This result shows that NSCABDT assigns nearly all points to the correct clusters.

4.1 Efficiency

It has been proved that DBSCAN has better performancethan partitioning and hierarchical algorithms for spatial data mining, so we only compare our algorithm with DBSCAN [28]. In the following, we compare NSCABDT with DBSCAN with respect to efficiency on synthetic databases. The run time and correct rate for NSCABDT, DBSCAN on these test databases are listed in Table 4.

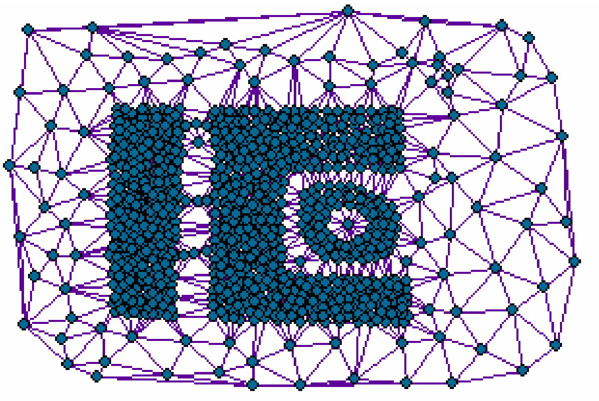

Figure 5. A data set and its Delaunay triangulation (n=1000)

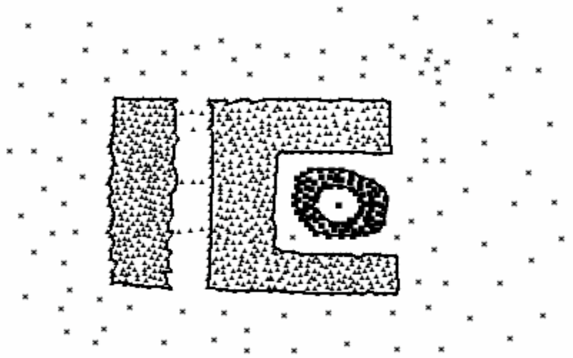

Figure 6. Extract the boundary of the cluster and eliminate the bridges

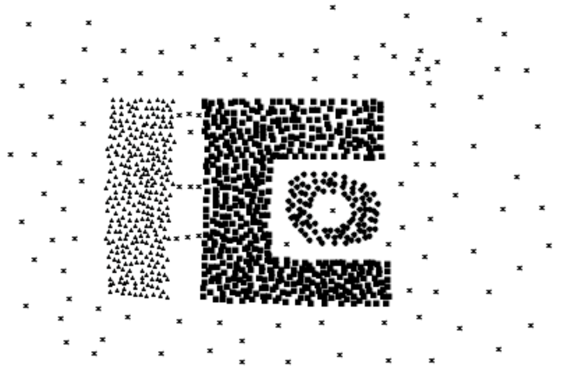

Figure 7. Clustering by NSCABDT. Finally we got 3 clusters

We generated some large synthetic test databases with 5000, 6000, 7000, 8000, 9000 and 10000 points to test the efficiency and scalability of DBSCAN and NSCABDT. We can conclude that NSCABDT is significantly slower than DBSCAN (see Figure 8), but the correct rate of NSCABDT is higher than DBSCAN (see Figure 9).

Because our approach does not require any assumptions or declarations concerning the distribution of the

Table 4. Run time in seconds

Number of Points	5000		6000		7000		8000		9000		10000	
	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time
DBSCAN	97.8%	36.7	97.2%	52.8	97.4%	72.6	97.9%	93.7	97.5%	121.5	97.8%	154.6
NSCABDT	99.7%	48.2	99.8%	71.4	99.8%	93.8	99.7%	117.9	99.7%	151.3	99.7%	189.4

data, the parameters of DBSCAN is difficult to be fixed. If the parameters are not inappropriate, the correct rate will be very low. DBSCAN must continually ask for assistance from the user. The reliance of DBSCAN on user input can be eliminated using our approach.

5. Conclusions

The application of clustering algorithms to large spatial databases raises the following requirements [27]: 1) minimal number of input parameters, 2) discovery of clusters with arbitrary shape and 3) efficiency on large databases. The well-known clustering algorithms offer no solution to the combination of these requirements.

In this paper, we introduce the new clustering algorithm NSCABDT. Our notion of a cluster is based on the distance of the points of a cluster to their neighbors. The neighboring region formed in our algorithm reflects the neighbor's distribution. Experimental results demonstrated that our clustering algorithm can provide significant improvement of accuracy of the cluster detecting, especially for objects with arbitrary and linear distribution.

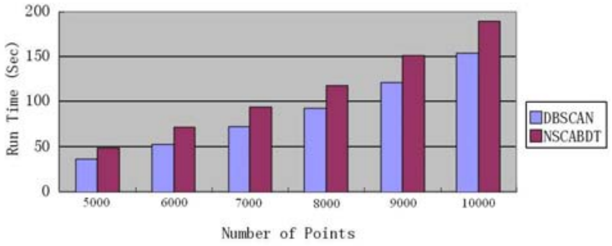

Figure 8. Efficiency: SCABDT VS DBSCAN

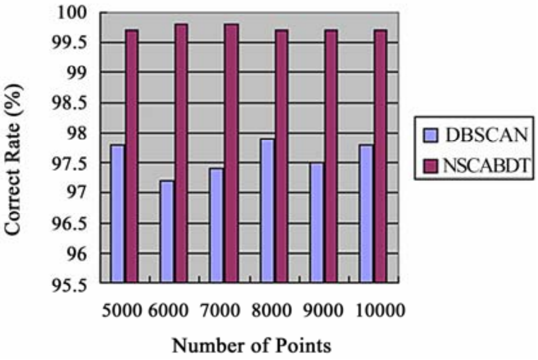

Figure 9. Correct rate: SCABDT VS DBSCAN

REFERENCES

[1] G. Piatetsky-Shapiro and W. J. Frawley. "Knowledge discovery in databases," AAAI/MIN Press, 1999.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. "The KDD process for extracting useful knowledge from volumes of data," Communications of ACM, Vol. 39, 1996.

[3] S. Shekhar, C. T. Lu, P. Zhang, and R. Liu, "Data mining for selective visualization of large spatial datasets," Processing of 14th IEEE international conference on tools with artificial intelligence (ICTAI'02), 2002.

[4] J. Han and M. Kamber, "Data mining: Concepts and Techniques," Academic Press, 2001.

[5] I. Atsushi and T. Ken, "Graph-based clustering of random point set," Structural, Syntactic and Statistical Pattern Recognition, Springer Berlin, pp. 948–956, 2004.

[6] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 5, pp. 1003–1016, 2002.

[7] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," Proceedings of the 1998 ACM SIGMOD international conference on Management of data, ACM Press, pp. 73–84, 1998.

[8] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," Proceedings of the 1996 ACM SIGMOD international conference on Management of data, ACM Press, pp. 103–114, 1996.

[9] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An introduction to cluster analysis," John Wiley & Sons, 1990.

[10] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "Density-based algorithm for discovering clusters in large spatial databases with noise," Proceedings of the 1996 Knowledge Discovery and Data Mining (KDD'96) international conference, AAAI Press, pp. 226–231, 1996.

[11] M. Ankerst, M. M. Breunig, H. P. Kriegel, *et al*., "OPTICS: Ordering points to identify the clustering structure," Proceedings of the International Conference on Management of Data (SIGMOD), ACM Press, pp. 49–60, 1999.

[12] J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The algorithm GDBSCAN and its applications," Data Mining and Knowledge Discovery, Vol. 2**,** No**.** 2, pp.169–194, 1998.

[13] X. Wang and H. J. Hamilton, "DBRS: A density-based spatial clustering method with random sampling," Proceedings of the 7th PAKDD, Springer, pp. 563–575, 2003.

[14] R. P. Haining, “Spatial data analysis in the social and environmental sciences,” Cambridge University Press, 1990.

[15] J. Han and M. Kamber, “Data Mining: Concepts and techniques,” Second Edition, Morgan Kaufmann, 2006.

[16] V. Estivill-Castro and I. Lee, “AMOEBA: Hierarchical clustering based on spatial proximity using delaunay diagram,” Proceedings of the 9th international symposium on spatial data handling, pp. 7a. 26–7a. 41, 2000.

[17] E. Schikuta and M. Erhart, “The BANG-clustering system: Grid-based data analysis,” Proceedings of the 2nd international symposium IDA-97, Advances in intelligent data analysis, Springer-Verlag, pp. 513–524, 1997.

[18] S. Openshaw, “A mark 1 geographical analysis machine for the automated analysis of point data sets,” International Journal of GIS, Vol. 1, No. 4, pp. 335–358, 1987.

[19] In-Soo Kang, Tae-wan Kim, and Ki-Joune Li, “A spatial data mining method by delaunay triangulation,” Proceeding of 5th ACM Workshop on Geographic Information Systems, Las Vegas, Nevada, pp. 35–39, 1997.

[20] C. Eldershaw and M. Hegland, “Cluster analysis using triangulation,” Computational Techniques and Applications (CTAC97), World Scientific, Singapore, pp. 201–208, 1997.

[21] V. Estivill-Castro and I. Lee, “AUTOCLUST: Automatic clustering via boundary extraction for mining massive point-data sets,” Proceedings of the 5th international conference on geocomputation, 2000.

[22] H. J. Miller, “Geographic data mining and knowledge discovery,” Handbook of geographic information science. Malden, MA: Blackwell, pp. 149–159, 2009.

[23] V. Estivill-Castro and M. E. Houle., “Robust Distance-based clustering with applications to spatial data mining,” Algorithmica, Vol. 30, No. 2, pp. 216–242, 2001.

[24] W. R. Tobler, “A computer movie simulating urban growth in the Detroit region,” Economic Geography, Vol. 46, No. 2, pp. 234–240, 1970.

[25] B. Delaunay, “Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk,” pp. 793–800, 1934.

[26] G. Liotta, “Low degree algorithm for computing and checking gabriel graphs”, Report No. CS–96–28, Department of Computer Science in Brown University, Providence, 1996.

[27] X. Xu, M. Ester, H. Kriegel, and J. Sander, “A distribution-based clustering algorithm for mining in large spatial databases,” Proceedings of the 14th International Conference on Data Engineering (ICDE’98), pp. 324–331, 1998.

[28] D. Y. Ma and A. D. Zhang, “An adaptive density-based clustering algorithm for spatial database with noise,” ICDM, Proceedings Fourth IEEE International Conference, pp. 467–470, 2004.