2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.2.08

An Effective Nonparametric Clustering Algorithm Based on Statistical Features of Neighborhood

Xi Peng[+], Zhang Yi and Dongchen Wei

Machine Intelligence Laboratory, College of Computer Science, Sichuan University

Chengdu, 610065, China

Abstract-Many clustering algorithms require to input user-specific parameter or acquire some prior knowledge. It is not practical in some real-world applications. In this paper, an algorithm without predetermined parameters or prior knowledge, called DECLUST, is proposed to cluster spatial numeric data set. DECLUST calculates the statistical features of neighborhood for each point, and creates the candidate clusters based on the statistical features with criteria functions. The final clustering result is obtained by bottom-up agglomerative method. The time complexity of the DECLUST is O(NlogN)(N is size of the data). Three well known algorithms (DBSCAN,k-mean and EM) together with four datasets are used to demonstrate the effectiveness of DECLUST. It shows that the DECLUST outperforms these three algorithms on five measure metrics. In addition, he anti-noise ability of DECLUST is analyzed. The precision is about 93.71% when the noise point rate is around at 60%.

Keywords-component; Nonparametric Clustering; Statistical Features; Delaunay Triangulation

1. Introduction

Many clustering algorithms require some user-specific parameters or prior knowledge to acquire a good result, which is unpractical because of:

- additional costs: multi-time trials are required to determine the value of parameter, which increases the time costs and space costs.
- poor robustness: the predetermined parameter only work well under specific conditions, that is, determination of parameters need to be performed for each dataset. For instance, DBSCAN[1] requires two predetermined parameters EPS and min Pts which are related with the distribution of dataset.
- the requirement of expert knowledge: to determine the value of parameter, it is necessary to understand therole of parameter, which puts a burden on users.
- limitation to the real-world application: some algorithms require the dataset to follow specific hypothesis, the dataset to be Gaussian distribution for instance. In most cases, however, it is hard to know whether this prior knowledge is satisfied or not.

Thus, the nonparametric clustering algorithm has attracted more and more attention of researchers. It mainly includes statistics-based method and graph-based method. Statistics based method is based on the hypothesis that the whole data set or its clusters follow given distribution models. The final result can be achieved by fitting the distribution of data set and pre-assumed model. Some works certified its effectiveness [2], [3], [4], [5]. However, it is difficult to get a desired result when the distribution of dataset is unknown or doesn't follow given hypothesis. Alternatively, graph-based method[6], [7] mainly includes *k*-Nearest Neighbor (kNN)method which implements classification or clustering task based on the feature of neighborhood. There are two key points in the kNN research area. One is determining the value of k, the other is finding a criteria function. Most kNN methods set the value of *k* by using cross-validation method to

[+] Corresponding author.
E-mail address: pangsaai@gmail.com

minimize the loss rate on the whole dataset, which requires additional costs and is likely to achieve poor result in high noise dataset.

This paper presents a nonparametric clustering algorithm which determines the neighborhood for each point via Delaunay triangulation method. The statistical features of neighborhood are used as basic metrics to create criteria function, and bottom-up agglomerative method is adopted to get the final result. The main contributions of this paper lie in:

- DECLUST can cluster numeric data without any pre determined parameter or prior knowledge.
- it is effective to process the complex datasets, such as, multi-resolution(multi-density) datasets, there is one or more bridges between two clusters, the data sets has different shape clusters, the higher density cluster ad joins the lower density cluster, etc.
- DECLUST has good anti-noise ability, which is essential to process the real-world dataset that includes a large amount of noise points, e.g. spatial data analysis.

2. Related Works

As one method of computational geometry, Delaunay triangulation can get the neighborhood for each object. In literature, some works introduced Delaunay triangulation into clustering research. RDBC [8] successfully obtained the number of the clusters, but the time complexity is high. SMTIN[9] extracted different shape clusters through categorizing the Delaunay edges into different groups. AMOEBA[10] got the clusters number and reduced the number of user specifies parameters. AUTOCLUST [11],to the best of our knowledge, is only one non parametrical gorithm (based on Delaunay triangulation) which used the average edge length as basic benchmark to divide all adjoining edges into three groups for each point. TRICLUST[12]also employed average edge length as benchmark, and found the boundary points of cluster and the intra-cluster points by employing k-mean. In spite of those achievements above-mentioned, there are still some problems needs to further study. Such as, almost all algorithms still require pre-determined parameters except AUTOCLUST; few methods can handle the complex datasets, etc.

Therefore, this paper proposes DECLUST algorithm which uses Euclidean distance as basic metric and adopts the median length of nearest edges as local feature and the mean length as global feature. DECLUST can work well without user-specific parameters and prior knowledge. Extensive experiments are conducted to manifest the effectiveness of DECLUST.

3. The Statistical Features Of Neighborhood

3.1.The definition of statistical features

Assuming there is a dataset $D = \{P_1, P_2, \dots, P_N\}$, where N is the number of D, and $N \geq 3$.

Definition 1: Neighborhood

The neighborhood $Ne(P)$is the sub-graph of the Delaunay diagram $Tri(D)$which is built by Delaunay triangulation on whole dataset. We called two points to be neighbor mutually only when they are linked by the edge of Delaunay diagram. For each point$P$, $\mathrm{Ne(P)} = \{\mathrm{TriP}, \mathrm{TriE}\}$, where $TriP$isthe ad joining points of P, and $TriE$is the correspondingadjoining edges.

Definition 2: Mean length of neighborhood

For each point$P$, $mean(P)$is the mean length of all edges inside its neighborhood$Ne(P)$, its mathematic formis:

$$\mathrm{mean(P)} = \sum_{L_i \in Ne(P)} |L_i|/k,$$

where$L_i$is the edge between point P and its $i$thadjoiningpoint, $|L_i|$ is the length of L_i, and $k$is the number ofpoints(edges) in$Ne(P)$.

Definition 3: Median length of neighborhood

For each point$P$, $median(P)$ is the median length of all edges inside its neighborhood $Ne(P)$, that is:

$$\text{Median(P)} = \begin{cases} (\text{Trie}_{\frac{k}{2}-1} + \text{Trie}_{\frac{k}{2}+1})/2\,, & k \text{ is even} \\ \text{Trie}_{\frac{k-1}{2}+1}, & k \text{ is odd} \end{cases},$$

where L_i, $|L_i|$, and k are defined as above.

Definition 4: Local mean absolute deviation

For each point P, $LocalAvgD(P)$ is defined:

$$\text{LocalAvgD(P)} = \sum_{L_i \in Ne(P)} |L_i - mean(P)|/k$$

Definition 5: Global mean absolute deviation

For Delaunay diagram $Tri(D)$, $GlobalAvgD$ is defined:

$$GlobalAvgD = \sum_{P_i \in Tri(D)} LocalAvgD(P_i)/N$$

Definition 6: Local mean relative deviation

For each point P, $PMD(P)$ is defined:

$$\text{PMD(P)} = \frac{LocalAvgD(P)}{GlobalAvgD}$$

Definition 7: Local median absolute deviation

For each point P, $LocalMD(P)$ is defined:

$$LocalMD(\text{P}) = \sum_{L_i \in Ne(P)} |L_i - meadian(P)|/k$$

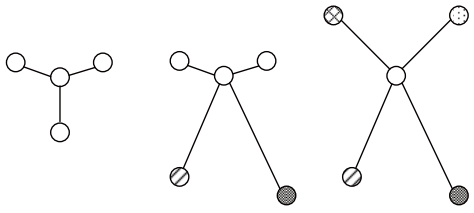
Definition 8: Global median absolute deviation

For Delaunay diagram$Tri(D)$, $GlobalMD$ is defined:

$$\text{GlobalMD} = \sum_{P_i \in Tri(D)} GlobalMD(P_i)\,/N$$

3.2.Criteria function

For each pointP ∈ Tri(D), there are three potentialneighborhoods as illustrated in Fig.1. Assuming $P$is amember of the clusterC_i. Here, the noise can also beregarded as a cluster. Fig.1(a) represents that P is an intra-cluster point, the points of $Ne(P)$are the members of C_i.Fig.1(b) represents that P is a boundary point of clusterC_i, its adjoining points include the points of $C_i$and thepoint outside C_i (noise point or the boundary point of otherclusters). Fig.1(c) is the neighborhood of the noise point Pwhose adjoining points include noise points and the cluster boundary points. From the three types shown in Fig.1,



(a) (b) (c)

Fig.1.all possible neighborhood graph, same points are regarded as into same cluster

We can notice the differences among those neighborhood graphs so that we are able to study the variation of values of the statistic features defined in Section III, and we can get following criteria function to construct candidate clusters.

Criteria Function 1:

For each point P ∈ Tri(D), where $\text{Trie}_i = (P, Q)$ is the edge between point Pand point Q. If following inequalityholds for Ne(P)or Ne(Q).

$$\text{Trie}_i \leq median(P) + GlobalMD,$$

Then, Trie_i is the short edge of Ne(P)or Ne(Q). Here, to use global median absolute deviation as corrected coefficient can avoid deleting the edge between intra-cluster points.

Criteria Function 2:

For $\forall \mathrm{Tri}_i \in Tri(D)$, $\mathrm{Trie}_i = (P, Q)$, if following inequality holds

$$Trie_i \leq \frac{\sum_{P \in Tri_D} LocalAvgD(P)}{2k} + \frac{GlobalAvgD}{PMD(P)},$$

then, $Trie_i$ is a global short edge, where k is the edges number inside $Tri(D)$. Here, to use local mean relativedeviation as corrected coefficient can avoid deleting the edge inside lower density cluster.

Criteria Function 3:

For candidate clusters C_i and C_j, the dissimilarity between them is defined as follows:

$$\mathrm{sim}(\mathrm{C}_i, \mathrm{C}_j) = (\mathrm{cmean}(\mathrm{C}_i) - \mathrm{cmean}(\mathrm{C}_j))^2 + \left(cAD(\mathrm{C}_i) - cAD(\mathrm{C}_j)\right),$$

where $\mathrm{cmean}(\mathrm{C}_i)$ means the average edge length in C_i, $cAD(\mathrm{C}_i)$ is the mean absolute deviation.

4. Algorithm Description and complexity analysis

There are three parts in DECLUST. The first part is calculating the neighborhood for each point, and finding the short edges by utilizing local statistical feature and global statistical feature.

step1: get the k nearest neighbor for each point with Delaunay triangulation, the value of k can be determined by Delaunay triangulation dynamically, and we can get corresponding adjacency graph $\mathrm{Tri(D)} = \{\mathrm{DP}, \mathrm{DE}\}$, where DP is the point set, DE is the collection of Delaunay edges. It includes following steps:

step2: calculate the statistical feature defined in Section III-A.

step3: for each point, find its local short edges based on Criteria Function 1, and all local short edges are included into $LocalshortEdge$.

step4: find global edge set Global short Edge based on Criteria Function 2.

step5: get the collection of short edge G', where $\mathrm{G}' = \mathrm{LocalshortEdge} \cup \mathrm{GlobalshortEdge}$.

step6: for point $\mathrm{P} \in \mathrm{G}'$, if $|\mathrm{TriE(P)} \cap \mathrm{G}'| \leq 1$, then P is a noise point.

The second part is to create candidate clusters:

step1: find the point P which is not labeled and has most adjoining edges, then create candidate cluster C_i, and label CP and CE with C_i, where $\mathrm{C_i} = \{\mathrm{CP}, \mathrm{CE}\}$, $\mathrm{CP} = \{\mathrm{TriP(P)} \cup \mathrm{P}\}$, $\mathrm{CE} = \{\mathrm{TriE(P)}\}$, $TriE(P)$ is the adjoining points of P, $TriE(P)$ is the adjoin edgesof P.

step2: for point $\forall \mathrm{S} \in \mathrm{TriP(P)}$, $\forall \mathrm{R} \in \mathrm{TriP(Q)}$, ifedge $\mathrm{e} = (\mathrm{S}, \mathrm{R})$ is not labeled, and the length of e follows $|\mathrm{e}| \leq \mathrm{maxlen}(\mathrm{C}_i) + \mathrm{clustAvgD}(\mathrm{C}_i)$, then add e and R into candidate cluster C_i, otherwise, delete edge e from G', where $\mathrm{maxlen}(\mathrm{C}_i)$ represents the max edge length of C_i clusteAvgD(C_i) is the mean absolute deviation of the edgesin C_i.

step3: go to *step1* until all edges of G' are labeled.

step4: assume the candidate clusters is $\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_k\}$, where k is the number of candidateclusters. We called C_i is a trivial component if the edgenumber in C_i less than threshold TCwith the default value3. Here, the TC is referred from AUTOCLUST[13], and its value is related to noise rate.

The third part, which is to merge candidate clusters for forming final result, includes following definition and steps.

step1: calculate the dissimilarity between each pair candidate clusters as defined in Criteria Function 3.

step2: for the minimums $sim(C_i, C_j)$, if following twoconditions are satisfied:

Firstly,

$$\max\left(\mathrm{cmean}(C_i), \mathrm{cmean}(C_j)\right) - \frac{cAD(C_i) + cAD(C_j)}{2} \leq \frac{cmean(C_i) + cmean(C_j)}{2}$$

Secondly, there is a edge $\mathrm{e} = (\mathrm{P}, \mathrm{Q})$ in original Delaunay diagram $\mathrm{Tri(D)}$, and $|e| \leq \mathrm{cmean}(\mathrm{C}_k) + \mathrm{cAD}(\mathrm{C}_k)$, where $(\mathrm{P} \in \mathrm{C}_i) \cup (\mathrm{Q} \in \mathrm{C}_j)$ or $(\mathrm{P} \in \mathrm{C}_j) \cup (\mathrm{Q} \in \mathrm{C}_i)$, and $\mathrm{C}_k = C_i \cup C_j$.

Then, create cluster $\mathrm{C}_k = \mathrm{C}_i \cup \mathrm{C}_j$, re-calculate the $cmean(C_k)$, $\mathrm{cAD}(\mathrm{C}_k)$ and the dissimilarity between C_k and other candidate clusters; otherwise, set $\mathrm{sim}(\mathrm{C}_i, \mathrm{C}_j)$ asglobal maximum.

step3: go to *step2* until above two conditions are not held to any candidate clusters.

step4: label all unclassified points as noise.

Assuming there are N point in dataset, the time cost of first part, which is mainly for finding the neighborhood with Delaunay triangulation and calculating the statistical features, is $O(NlogN + Nk)$, where k is the average number of adjoining edges for each point. From Euler's formula, we can arrived $k \leq 6$. The costly time for second part is traverse operation with $O(NlogN)$. The time complexity of thirdpart is $O(mlogm$, where m is the number of candidate clusters, and $m \ll N$. So the total time complexity ofDECLUST is $O(NlogN)$.

5. Experiments

We implement DECLUST on Mat lab 2010a and VC++ 2008. Four datasets (DS1, DS2, DS3, DS4) and three algorithms(DBSCAN, k-mean, EM) are used to test the effectiveness of DECLUST. Some materials, such as the code of DECLUST, the data sets, can be downloaded from our website[1].

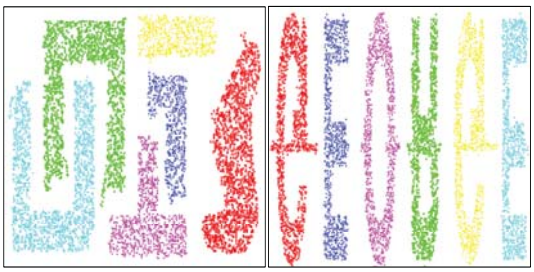
DS1, which is a synthetic dataset, includes four different shape clusters with different density, where the density of circle cluster is the highest, and the density of rectangular box is the lowest. Additionally, the noise points of DS1are generated by random, where we can increase or decrease the size of any clusters or noise. DS2 includes 8000 points distributed over 6 clusters and noise. Each cluster appears to be a character, and some noises form multi-bridge to link all clusters. DS3 and DS4 are Chameleon datasets[13],each dataset includes 8000 points. DS3 is composed of six clusters, and some noise points form a sine-curve-like bridge to links all clusters. Like DS3, DS4 includes nine different shape clusters, and there is a bridge between two pie shape clusters and a multi-bridge between two rectangular areas.

To verify the performance of DECLUST, the experiment includes two schemas. First is the performance comparison among DECLUST, DBSCAN, EM and k-mean on four datasets. Second is to compare the cluster quality of DECLUST under different noise rate.

5.1.Performance comparison

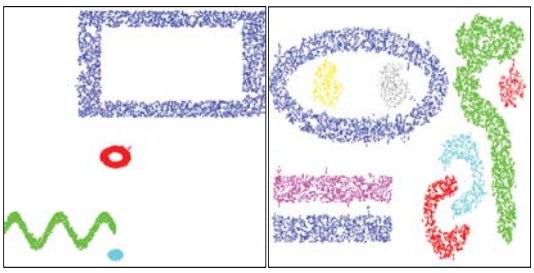
In this experiment, DS1 includes 6000 points, and 10%is noise. Five key abilities are considered to evaluate the performance of four cluster algorithms. Firstly, the ability of working on the dataset with complex shape clusters. Secondly, the ability of processing multi-density dataset(refer to the result to DS1). Thirdly, the number of recognized clusters. Fourthly, whether the algorithms can work on dataset which includes bridge links two or more clusters(refer to the result to DS2, DS3, DS4). Finally, the ability of finding the clusters when high density cluster is adjacent with low density cluster(refer to the result to DS1).

DBSCAN requires to input two parameters eps and min Pts. For fair comparison, we changed the value of eps from 0.01 to 4.0, min Pts from 2 to 6, and we got the best result when eps = 0.01, min Pts = 4 for DS1 andDS2; eps = 0.012, min Pts = 4 for DS3; eps = 0.015,minPts = 5 for DS4. For algorithm EM and k-mean, we set the parameter iteration as 10^4, and the number of seedas the size of dataset. In addition, EM produces two result son same dataset, one is when we input the cluster number like k-mean, and the other is running in nonparametric mode like DECLUST.



(a)DS1 (b) DS2

[1]http://www.machineilab.org/users/pengxi/



(c)DS3 (d) DS4

Fig.2.The cluster result of DECLUST on four data sets

We just show the result of DECLUST (see Fig. 2) since the limited space. From the experimental results, we find that DECLUST works well on all test datasets, and it is predominant on five important abilities as above-mentioned. DBSCAN, which is the sub-optimal algorithm, is successful to produce the results on DS1 and DS3. However, it performs poor when two clusters get closer to each other. As for EM, it is interesting that EM performs better when no cluster number is input, and it find three clusters from DS1 without parameters. Finally, k-mean algorithm obtains similar results on test datasets like EM with expected cluster number.

Table I is the overview of the cluster number generated by testing algorithms on four datasets, except the results of k-mean and EM with prior knowledge which require to input the cluster number.

TABLE I. THE CLUSTER NUMBER GENERATED BY TESTING ALGORITHMS

Data set	*cluster#*	DECLUST	DBSCAN	EM
DS1	4	4	5	6
DS2	6	6	46	7
DS3	6	6	52	23
DS4	9	9	24	34

5.2.Cluster quality

The precision is widely used to measure the cluster quality of algorithm, its definition is:

$$\text{precision} = \frac{TP}{TP + FP},$$

where *TP*(True Positive) means an observed positive result is a true positive, by contrast, an observed positive result is false, then we called it is *FP*(False Positive).

Following two experiments are designed to evaluate the cluster quality of DECLUST with metric Precision. The first result is the precision of DECLUST on above four data sets(see Fig. 3); the second result is the precision of DECLUST on DS1 with different noise rate(see Fig. 4).

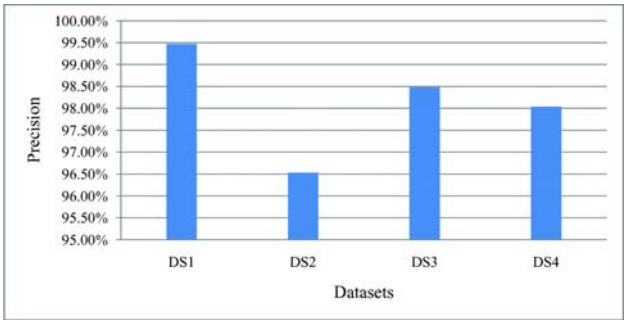


Fig.3.the precision of DECLUST on four datasets

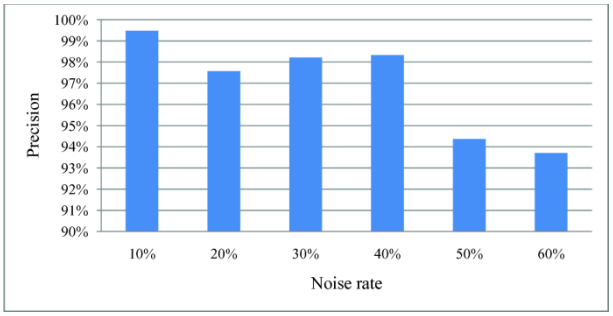
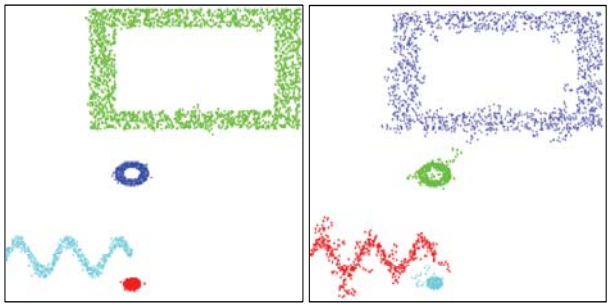


Fig.4.the anti-noise ability of DECLUST

As shown in Fig. 3, the precision scores of DECLUST on all datasets are higher than 96%. DECLUST performs best on DS1, we assume the reason to be the bridge in DS2, DS3 andDS4 leading error classifying noise points into clusters which cause increasing of FP and decreasing of precision. Moreover, another potential reason for DECLUST to perform worst on DS2(96:53%) is the special shape of clusters G making more noise points of bridge are clustered into this clusters in correctly. Finally, we can see the precision of algorithm decreased as the number of clusters increased(see the result on DS4).

Fig. 4 is the precision of DECLUST on DS1 with different noise rate, where the noise rate increased from 10% to 60%with interval 10%. It is not strange that the precision decreased with the increasing of noise rate. However, in the worst case, DECLUST still can successfully find most points.

For getting a more directed view on the anti-noise ability of DECLUST, and limited by the space, we show the cluster results of DECLUST on DS1 with 30% and 60% noise rate respectively(see Fig. 5).



(a) 30% noise point(b) 60% noise point

Fig. 5.The result of DECLUST on DS1 with 30% and 60% noise rate respectively

6. Conclusion

In this paper, we propose a effective nonparametric clustering algorithm DECLUST which implements clustering based on the statistic features of neighborhood. Related definitions and criteria functions are defined, and the reasons for adopted feature benchmarks (mean length and median length) are analyzed. A series of experiment is conducted to evaluate the effectiveness of DECLUST. The performance comparison results demonstrate that DECLUST outperforms peer algorithms on five important metrics. In addition, the cluster quality experiment verifies its remarkable anti-noise ability.

There are some potential future research directions to extend and improve this work, especially, extending DECLUST to cluster high dimension dataset. Delaunay triangulation, which is utilized to determine the neighborhood in our method, on high dimension dataset still is a challenging problem. Therefore, it is feasible to process the high dimension dataset by combining the definitions and criteria functions in this paper with other kNN method.

7. Acknowledgment

This work was supported by the Chinese 863 High-Tech Program under grant 2008AA01Z119.

8. References

[1] Ester M., Kriegel H.P., Sander J., et al. *A density-based algorithm ford is covering clusters in large spatial databases with noise*. The Second International Conference on Knowledge Discovery and Data Mining. Evangelos Simoud is, AAAI Press, 1996, Pages: 226-331.

[2] Peter Mller and Fernando A. Quintana. *Nonparametric bayesian data analysis*. Statistical Science, 2004, 19(1):95-110.

[3] Katherine A. Heller and Zoubin Ghahramani. *A nonparametric Bayesian approach to modeling overlapping clusters*. Journal of Machine Learning Research, 2007, 2:187-194.

[4] M. Law, A. Topchy, A. Jan. *Clustering with Soft and Group Constraints*. The Joint IAPR Int. Workshop on Structural, Syntactic, and Statistical Pattern Recognition, 2004, pp. 662-670.

[5] Witten L. H. and Frank E. Data Mining: Practical Machine Learning Tools and Techniques, 2rd. Elsevier Science and Technology, 2005.

[6] Wang X., Qiu W. and Zamar R. H. *CLUES: A non-parametric clustering method based on local shrinking*. Computational Statisticsand Data Analysis, 2007, 52(1): 286-298.

[7] Franti P., Virmajoki O. and Hautamaki V. *Fast Agglomerative Clustering Using a k-Nearest Neighbor Graph*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(11): 1875-1881.

[8] Vladimir Estivill-Castro and M. E. Houle. *Robust clustering of large Geo-referenced data sets*. The 3rd Pacific-Asia Conference on knowledge Discovery and Data Mining, London, UK: Springer-Verlag Press,1999, Pages: 327-337.

[9] Kang, T. Kim, and K. Li. *A spatial data mining method by Delaunay Triangulation*. The 5th International Workshop on Advances in Geo-graphic Information Systems, Las Vegas, Nevada, ACM Press, 1997,Pages: 35-39.

[10] Vladimir Estivil-Castro and I. Lee. *AMOEBA: Hierarchical clustering based on spatial proximity using Delaunay diagram*. The 9th Int.Spatial Data Handling, Beijing, China, ACM Press, 2000, Pages: 10-26.

[11] Vladimir Estivill-Castro and Ickjai Lee. *AUTOCLUST-Automatic clustering via boundary extraction for mining massive point-data sets*. The5th Int. Conf. Geo-Computation, University of Greenwich, Kent, UK,2000, Pages: 26-41.

[12] Liu D. Q., Nosovskiy G. V. and Sourina O. *Effective clusteringand boundary detection algorithm based on Delaunay triangulation*. Pattern Recognition Letters, 2008, 29(9): 1261-1273.

[13] Karypis G., Han E.H. S. and Kumar V. *Chameleon: Hierarchical Clustering Using Dynamic Modeling*. IEEE Computer, 1999, 32(8):68-75.