Field-theory based spatial clustering method

DENG Min[1], LIU Qiliang[1], LI Guangqiang[1], CHENG Tao[2]

1. *Department of Surveying and Geo-informatics*, *Central South University*, *Hunan Changsha* 410083, *China*;
2. *Department of Civil*, *Environmental and Geomatic Engineering*, *University College London*, *Gower St*, *WC1E 6BT*, *London*, *the UK*

Abstract: Spatial clustering is an important tool for spatial data mining and spatial analysis. It can be used to discover the spatial association rules and spatial outliers in spatial datasets. Currently most spatial clustering algorithms cannot obtain satisfied clustering results in the case that the spatial entities distribute in different densities, and therefore more input parameters are required. To overcome these limitations, a novel data field for spatial clustering, called aggregation field, is first of all developed in this paper. Then a novel concept of aggregation force is utilized to measure the degree of aggregation among the entities. Further, a field-theory based spatial clustering algorithm (FTSC in abbreviation) is proposed. This algorithm does not involve the setting of input parameters, and a series of iterative strategies are implemented to obtain different clusters according to various spatial distributions. Indeed, the FTSC algorithm can adapt to the change of local densities among spatial entities. Finally, two experiments are designed to illustrate the advantages of the FTSC algorithm. The practical experiment indicates that FTSC algorithm can effectively discover local aggregation patterns. The comparative experiment is made to further demonstrate the FTSC algorithm superior than classic DBSCAN algorithm. The results of the two experiments show that the FTSC algorithm is very robust and suitable to discover the clusters with different shapes.

Key words: spatial clustering, aggregation force, field theory, spatial data mining

CLC number: P208 **Document code:** A

Citation format: Deng M, Liu Q L, Li G Q and Cheng T. 2010. Field-theory based spatial clustering method. *Journal of Remote Sensing*. **14**(4): 694—709

1 INTRODUCTION

Spatial clustering is an important means for spatial data mining and spatial analysis, and it can be used to discover the potential rules and outliers hidden in the spatial data. Currently, spatial clustering technology has widespread applications in various fields, such as geography, geology, cartography, remote sensing, biology, economics (Blackman & Popoli, 1999; Bar-shalom & Blair, 2000; Hofmann-wellenhof *et al*., 1994; Mao & Li, 2004).

Through the analysis of literature, current spatial clustering algorithms can be roughly classified into five categories: (1) partitioning algorithms, such as k-Means (Macqueen, 1967), k-Mediods (Ng & Han, 1994) and FCM (Dave & Bhaswan, 1992); (2) hierarchical algorithms, like BIRCH (Zhang *et al*., 1994), CURE (Guha *et al*., 1998) CHAMELEON (Karypis *et al*., 1999), ROCK (Guha *et al*., 1999) and gravity-based algorithms (Wright, 1977; Gan, 2006); (3) density-based algorithms, including DBSCAN (Ester *et al*., 1996), VDBSCAN (Liu *et al*., 2007), OPTICS (Ankerst *et al*., 1999), ADBSC (Li, 2009) and DDBSC (Li, 2008); (4) graph-based algorithms, as ZEMST (Zahn, 1971), SFMST (Paivinen, 2005), AUTOCLUST (Estivill-Castro & Lee, 2000); and (5) mixed spatial clustering algorithms, such as STING (Wang *et al*., 1997), Wave Cluster (Sheikholeslami *et al*., 1998), CLIQUE (Agrawal *et al*., 1998), GDCIC (Song & Ying, 2006) and NN-Density (Pei *et al*., 2006).

Partitioning algorithms begin with an initial partition into *k* clusters and then optimize the criterion function via an iterative control strategy. The iterative process ends until the value of criterion function makes convergence to a given threshold. The partitioning algorithms seriously rely on the input parameters (i.e. the settings of the cluster number) and the initial centers. In addition, the algorithms are very sensitive to noise and cannot discover clusters with different shapes. Hierarchical algorithms usually create a hierarchical decomposition for a given spatial database. A dendrogram which splits the database iteratively into small subsets is usually used to represent the hierarchy. The dendrogram can be formed through agglomerative approach or divisive approach. The former begins with each point which is viewed as an individual cluster, and then the points or clusters are successively merged until a predefined condition holds or all the points are merged into one cluster. The latter starts with all the points into one cluster, then the cluster is hierarchically

Received: 2009-06-26; **Accepted**: 2009-09-07

Foundation: National "863" High Technology Research and Development Program of China (No. 2009AA12Z206), Key Laboratory of Geo-Informatics of State Bureau of Surveying and Mapping (No. 200805) and Scientific Research Foundation of Jiangsu Key Laboratory of Resource and Environmental Information Engineering at China University of Ming and Technology (No.20080101).

First author biography: DENG Min (1974—), male, professor. He receives the doctoral degree from the Wuhan University in 2003. His major research areas include spatio-temporal data mining, reasoning and analysis. He has published over 90 journal papers. E-mail: dengmin208@tom.com

split into smaller ones until a predefined condition holds or each point in one cluster. However, the input parameters have a significantly influence on the clustering results. Particularly, these parameters are hard to determine. Additionally, the hierarchical algorithms cannot work well in the case that there is a distinct difference among the local density of the spatial datasets. Density-based algorithms try to identify the clusters due to the idea that the density of the points within each cluster is considerably higher than that outside the cluster. The algorithms can discover clusters of arbitrary shapes, and can be used to filter outliers. However, the input parameters of the algorithms are usually fixed, which directly lead to unsatisfied clustering results when the spatial entities distribute in different densities. Moreover, there is not an effective method to set the parameter and threshold without sufficient priori information. Graph-based algorithms firstly construct a graph. Each point is represented as a vertex and edges connect pairs of points. Then, a series of sub graphs are generated by removing uninteresting edges on the basis of a certain criterion function, and each of them may be regarded as a cluster. These uninteresting edges are usually the significantly long or short edges, but actually, these edges are hard to find in the case of the uneven distribution in spatial database. Mixed spatial clustering algorithms usually work with the combination of different clustering algorithms. Likewise, the mixed algorithms cannot overcome the limitations discussed above. In addition, the clustering quality would be reduced, such as the STING algorithm.

For existing spatial clustering algorithms, two aspects of limitations can be summarized. On the one hand, the spatial distribution of the spatial entities is not fully considered, so that it is almost impossible for existing clustering algorithms to obtain satisfied clustering results in the case that the spatial entities distribute in different densities. On the other hand, more input parameters are required, but the establishment basis for each parameter is unclear. In order to overcome these limitations, the data field theory is firstly employed to describe the issue of spatial clustering, where a novel data field for spatial clustering is developed by means of Voronoi diagram and Delaunay triangulation. Then, a new concept of clustering measurement, called aggregation force, is developed. Further, a field-theory based spatial clustering algorithm (FTSC in abbreviation) is proposed. Compared with exiting spatial clustering algorithms, FTSC algorithm has two aspects of advantages. Firstly, the algorithm can adapt to the change of local density among spatial entities and can discover clusters with different shapes effectively and steadily. Secondly, the algorithm does not involve the setting of input parameters.

2 PRINCIPLE AND DESCRIPTION OF THE SPATIAL CLUSTERING ALGORITHM BASED ON FIELD THEORY

The goal of spatial clustering is to classify the entities of a database into a set of meaningful subclasses, in which the entities are similar to each other in geometry (e.g. shape, size) and the entities of different subclasses are of large dissimilarity. Currently, the geometrical distance, such as Euclidean distance, is frequently used to measure the similarity and dissimilarity during spatial clustering procedure (Kovács *et al*., 2006). A clear physical meaning for spatial clustering processes and results are also lack. Enlightened by the field theory in physics, the data field theory is employed to describe the mechanism of spatial clustering (to be discussed in detail in the next section) in this paper. A novel data field, called aggregation field is firstly developed for spatial clustering.

2.1 Aggregation field and aggregation force

From the physical point of view, each data can be assumed to have its energy, and the energy of data can radiate to the whole space. Therefore the space which receives the energy is called aggregation field (Wang, 2002). The aggregation field here is a kind of data field; which is further defined as an active vector field. Each entity can be viewed as a source of the aggregation field, also called aggregation field source. It is assumed that each entity in the aggregation field is influenced by the source with an aggregation force. Thus, the mechanism of spatial clustering can be described as follows:

(1) Each source generates an aggregation field, and each entity in that aggregation field is attracted by the source with an aggregation force.

(2) Each cluster is formed from a source whose attractive ability is powerful. Then, each entity attracts other entities in turns until a cluster is contributed.

(3) The aggregation force acted on an entity in a cluster is powerful. The aggregation force acted on an outlier is significantly small.

The spatial distribution of the aggregation field can be represented by the vector field-strength function. Generally, short-rang field is proper for spatial clustering (Gan *et al*., 2006), that is to say, the field-strength of a data field decays rapidly in a range of distance. Thus, the exiting gravity-based methods usually employ the resistant consumption or influence factor strategy, but virtually these methods are hard to achieve. In this paper, the Voronoi diagram and Delaunay triangulation are employed to define the field-strength function of the aggregation field. In the following, some related concepts are firstly explained.

Definition 1: Given spatial entities set $\boldsymbol{P}$, $\boldsymbol{P}$= $\{p_1, p_2, p_3 \ldots, p_n\}$, $\forall p_i \in \boldsymbol{P}$, the Voronoi diagram of p_i is defined as (Chen, 2002):

$$p_i^V=\{x \mid \mathrm{d}(x, p_i) \leqslant \mathrm{d}(x, p_j), p_i, p_j \in \boldsymbol{P}, i \neq j, x \in \mathrm{R}^2\} \quad (1)$$

where, d denotes the Euclidean distance function. The Voronoi diagram of set $\boldsymbol{P}$ is formed by the Voronoi diagram of all the entities in set $\boldsymbol{P}$, which is denoted by $\boldsymbol{P}^{\boldsymbol{V}}$. That is the diagram bounded by the real line, as shown in Fig. 1. It can be expressed as:

$$\boldsymbol{P^V}= \{p_1{}^V, p_2{}^V, p_3{}^V, \ldots, p_n{}^V\} \quad (2)$$

The dual graph of $\boldsymbol{P^V}$ is the Delaunay triangulation (the dotted line in Fig. 1), denoted by $\boldsymbol{D}$ ($\boldsymbol{P}$).

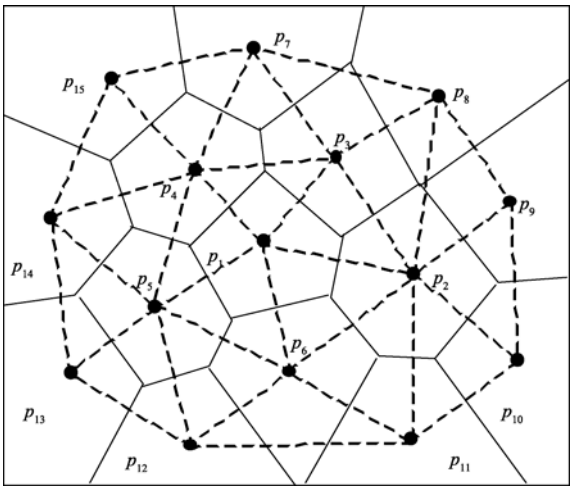

Fig. 1 Aggregation field

Definition 2: Given spatial entities set $\boldsymbol{P}$, $\boldsymbol{P}$= {p_1, p_2, p_3… p_n}, $\forall p_i$, $p_j \in \boldsymbol{P}$, if $p^V{}_i$ and $p^V{}_j$ have the same Voronoi edge, p_i and p_j are defined as the directly Voronoi neighborhood entities with each other (Gold, 1992). All the directly Voronoi neighborhood entities of an entity p_i are the Delaunay neighborhood of p_i, denoted by $\boldsymbol{ND}$ (p_i). In Fig. 1, the Delaunay neighborhood of p_1 are the entity p_2, p_3, p_4, p_5 and p_6.

Definition 3: Given an spatial entity p_i, the directly Voronoi region is defined as the region formed by $p^V{}_i$ and all the Voronoi diagram of the entities in $\boldsymbol{ND}$ (p_i), denoted as $\boldsymbol{DNV}$(p_i). In Fig. 1, the directly Voronoi regions of p_1 are $p_1{}^V$, $p_2{}^V$, $p_3{}^V$, $p_4{}^V$, $p_5{}^V$ and $p_6{}^V$, respectively.

Intuitively, the construction process of Voronoi diagram can be described as follows. Each entity in the plane can be viewed as a core, and then each core expands to all the directions in the same rate to form a certain region. When the regions reach each other, the Voronoi diagram of each entity is finally formed. In other words, the Voronoi diagram reflects the natural influence region of each entity. According to this property, the field-strength function of the aggregation field can be defined as:

$$E_p = k\frac{1}{\mathrm{d}(p,x_i)^{2\sigma}}\boldsymbol{e}_{px_i}, \sigma = \begin{cases}1, x_i \in \boldsymbol{DNV}(p)\\ +\infty, x_i \notin \boldsymbol{DNV}(p)\end{cases} \quad (3)$$

where, E_p is the field-strength of an aggregation field at a certain spatial location x_i (The aggregation field is generated by source p); k is the radiation factor of the aggregation field, and the value of k is set to 1 in this paper; d(p, x_i) is the Euclidean distance between p and x_i; σ is the attenuation factor; $\boldsymbol{e}_{px_i}$ is the unit vector from p to x_i.

In Eq. (3), one can find that the attenuation rate of the field-strength function satisfies the square inverse relation to the distance between the source and a certain location in the directly Voronoi region of that source. Out of the directly Voronoi region, the field-strength weakens rapidly, so the influence of which can be neglected. Additionally, the assumption of the aggregation field is also consistent with the essential features of the data field that are "independence, adjacency, ergodicity, additively, attenuation and isotropism". Further, the aggregation force can be defined as follows:

$$\begin{aligned}F_C(p,q) &= E_p m_q = k\frac{1}{\mathrm{d}(p,q)^{2\sigma}}m_q\boldsymbol{e}_{pq}\\ &= \frac{m_q}{\mathrm{d}(p,q)^{2\sigma}}\boldsymbol{e}_{pq}, \sigma = \begin{cases}1, q \in \boldsymbol{ND}(p)\\ +\infty, q \notin \boldsymbol{ND}(p)\end{cases}\end{aligned} \quad (4)$$

where, p is the source; m_q is the mass of the entity q, and m_q is set to 1 because the spatial entity can be regarded as the unit particle; d(p, q) is the Euclidean distance between p and q; σ is the attenuation factor; $\boldsymbol{e}_{pq}$ is the unit vector from p to q.

One can find from Eq. (4) that the aggregation field source only influences the entities in its Delaunay neighborhood; the effect on other entities can be neglected. The aggregation force here is to some degree different from the gravitation (Wright, 1977; Gan, 2006). The former gravitation concept is a scalar, but the aggregation force is a vector that the direction is considered. The difference between the two concepts is shown in detail in the next section.

2.2 The principle of FTSC algorithm

Based on the aggregation field and the aggregation force described above, the principle of FTSC algorithm can be described as that each entity attracts the other entities in its Delaunay neighborhood with the aggregation force and each cluster is constructed from a source whose attractive force is very large. Thus, there are mainly two steps for the FTSC algorithm. One is to discover the mechanism that an entity attracts other ones, and the other is to find the entity whose attractive force is maximal. In the following, the two steps will be respectively introduced.

According to the action and reacting force principle, when the source entity attracts other entities, these entities also attract the source. So the direction of the cohesive force that the source receives must point to the center of a local cluster (Li, 1999). That is to say, the entity in the opposite direction of the cohesive force can be significantly attracted by the source. The cohesive aggregation force that the source receives can be calculated as follows:

$$F_T = \sum F_C(p,q), q \in \boldsymbol{ND}(p) \quad (5)$$

Further, it can be found that if the angle between the cohesive force and the component aggregation force is smaller than 90°, the component force would have contribution to the cohesive force, so that it can be concluded that the source attracts the entity which causes the component aggregation force on the source. In Fig. 2(a), entity A is the source, the dotted directional line represents the cohesive force on A and the real directional lines representing the component aggregation forces. Obviously,

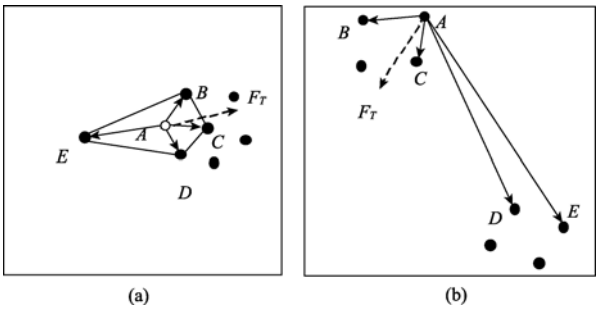

Fig. 2 Attractive operations
(a) Example of the calculation of aggregation forces; (b) Example of boundary effect

the angle between F_C (A, B) and F_T, the angle between F_C (A, C) and F_T, the angle between F_C (A, D) and F_T are all smaller than 90°, so the source A attracts the entity B, C and D. The angle between F_C (A, E) and F_T is larger than 90°, which means F_C (A, E) weakens the cohesive force, so it can be considered that the source A can not attract the entity E.

However, there is also a special condition, called boundary effect in this paper. In Fig. 2(b), when the entity A is viewed as a source, it is easy to find that the angle between each component aggregation force and the cohesive force is smaller than 90°, but obviously F_C (A, D) and F_C (A, E) are significantly smaller compared to the other component aggregation force. This also means that the source should not attract them. The reason for this case is that entity A is on the boundary of a cluster, thus the entities in other cluster make negative influence. For such special case, a constraint of the aggregation force scalar is developed. The process that each entity attracts other entities can be described as follows:

(1) Given a source p, for each entity q in the Delaunay neighborhood of p, if the angle between the cohesive force p receives and the component aggregation force between p and q is smaller than 90°, then q is viewed as the candidate entity which can be attracted by source p. All the candidate entities form the set ***CNS***(p), represent as:

$$\boldsymbol{CNS}(p)=\{q \mid \theta(F_C(p, q), F_T)<90°, q\in \boldsymbol{ND}(p)\} \quad (6)$$

where, θ (F_C (p, q), F_T) represents the angle between the component force and the cohesive force.

(2) Calculate the average aggregation force of ***CNS***(p). It includes two steps. The first is to remove the [N/2] least component aggregation forces; the second is to calculate the average value of the remaining aggregation forces, denoted by $E_p(F)$. Here symbol [] represents the rounding operation, and N is the number of the entity in ***CNS*** (p).

(3) Filter operation. If a source locates in the interior cluster, it can attract all the entities in ***CNS*** (p). If the source locates in the verge of a cluster, there may be some entities in ***CNS***(p), the aggregation forces cause by which are significantly small. This also means that the source can not attract them (the condition shown in Fig. 2(b)). Through numerous experiments, a constraint of the aggregation force scalar is developed. That is, if the aggregation force between an entity and the source is larger than 1/5 of $E_p(F)$, then it means that the source can attract the entity. Further all such entities form the set ***NS***(p), represented as:

$$\boldsymbol{NS}(p)=\{q \mid |F_C(p, q)| > E_p(F)/5, q\in \boldsymbol{CNS}(p)\} \quad (7)$$

where, $|F_C(p, q)|$ represents the aggregation force scalar.

After the above procedures, in Fig. 2(b) the interference of entity D and E can be excluded effectively. Next, how to find the entity whose attractive ability is powerful is another important part of FTSC algorithm. Intuitively, for a source, if the scalar sum of aggregation force is large, the attractive ability of the source is powerful. The scalar sum of aggregation force for a source p can be defined as follow:

$$|F_T| = \sum |F_C(p,q)|, q\in \boldsymbol{ND}(p) \quad (8)$$

In this paper, the source with powerful attractive ability is defined as the clustering core.

Definition 4: Given spatial entities set ***P***, set ***F***(***P***) is formed by the aggregation force scalar sum of each entity in ***P***. The entity, whose aggregation force scalar sum is the maximum one in ***F***(***P***), is defined as clustering Core which can be represented as:

$$Core(\boldsymbol{P})= p, SF(p) = \max(\boldsymbol{F}(\boldsymbol{P})) \quad (9)$$

Then, given spatial entities set ***P***, the clustering process based on FTSC can be described as follows:

(1) Select the clustering core.

(2) From the clustering core, each entity attracts other entities in turns until a cluster is contributed. If a clustering only has one entity, it will be labeled as outlier. The entities which have been added in a cluster or identified as outliers will be removed from set ***P***.

(3) Repeat procedures (1) and (2) until each entity is either added into a cluster or labeled as outlier.

According to the expression of aggregation force, if the scalar sum of aggregation force for a source is large, the source must be in the high density part of the spatial dataset. So a series of cluster can be automatically generated from the high-density region to the low-density one. Thus the FTSC algorithm can adapt to the change of local density among spatial entities. To avoid obtaining the clusters which are too loose, the entities are first filtered according to their aggregation force scalar sum. If the aggregation force scalar sum of an entity is significantly small, it can not be set as clustering core (the operation to be discussed in detail in section 2.3).

2.3 The FTSC algorithm

SDB is the spatial database for clustering. The main steps of implementation of the FTSC algorithm are elaborated as follows:

Step 1: Construct the Delaunay triangulation of ***SDB***; calculate the aggregation force scalar sum for each entity p_i, and form the set ***F*** (***SDB***).

Step 2: Filter the entities according to aggregation force scalar sum. If $|F_T(p_i)|$–average (***F*** (***SDB***)) is smaller than -3σ, then the entity p_i can not be set as clustering core. Where average

(***F*** (***SDB***)) represents the average value of the aggregation force scalar sum in ***F*** (***SDB***), σ is the variance of the aggregation force scalar sum in ***F*** (***SDB***).

Step 3: Select the clustering core, and then the core attracts other entities; form the initial cluster ***C***.

Step 4: For each entity in cluster ***C*** which has not attracted other entities, attract other entities and add them into the cluster ***C***. When all the entities in cluster ***C*** have attracted other entities, the cluster is formed.

Step 5: If a cluster has only one entity, the entity is identified as outlier.

Step 6: Repeat step 3 to step 5, until each entity in ***SDB*** is either added into a cluster or labeled as outlier.

3 EXPERIMENTAL RESULTS

In this section, two experiments are utilized to verify the feasibility and correctness of the FTSC algorithm. Four simulated databases are used in the first experiment. In the second experiment, two real-world datasets are employed. Moreover, the experiment results are compared with the classic DBSCAN algorithm.

3.1 Simulation experiment

In this experiment, the three simulated databases are firstly utilized to test that the FTSC algorithm can discover clusters with different shape and is robust with outliers. The databases are shown in Fig.3 (a)—(c). Fig. 3 (d)—(f) show the clustering results obtained by FTSC algorithm.

It can be found that FASC algorithm can get the same results as DBSCAN algorithm. The result in Fig. 3(d) is the same as the result obtained by DBSCAN with Eps=4 to 9, Minpts=1 to 11. The result in Fig. 3(e) is the same as the result obtained by DBSCAN with Eps=4 to 6, Minpts=1 to 17. The result in Fig. 3(f) is the same as the result obtained by DBSCAN with Eps=4, Minpts=2 to 6. From the above results, one can see that on the one hand the FTSC algorithm can discover clusters with different shapes as effectively as the DBSCAN algorithm; on the other hand the algorithm is very robust, even the outliers which are near the cluster can be identified effectively. The clusters which are near each other can be also distinguished correctly via FTSC algorithm. Although the DBSCAN algorithm can obtain the same results as FTSC algorithm, the input parameters must be strictly set. For instance, considering database 3, the optimum result can be obtained by DBSCAN only with Eps=4.

Next, in order to test that the FTSC algorithm can automatically adapt to the change of local density, database 4 is designed. Fig. 4(a) shows the distribution of the dataset 4 which contains 336 entities. 6 clusters and 13 outliers are predefined (marked by symbol ×). It can be found from Fig. 4(a) that the spatial distribution of these entities is uneven, and the shapes of the clusters differ widely. The clustering results by FTSC algorithm is shown in Fig. 4(b). In order to illustrate the advantage of our algorithm, a comparison experiment has been done with the DBSCAN algorithm. The clustering results are shown in Fig. 4(c)—(h). DBSCAN requires two input parameters, Minpts and Eps. The optimum value of Minpts is ln(n) via experiments, where n is the number of points in the dataset (Birant & Kut, 2007). In this paper, we also use this optimum value of MinPts, so the value of MinPts is set as ln(336)≈6.

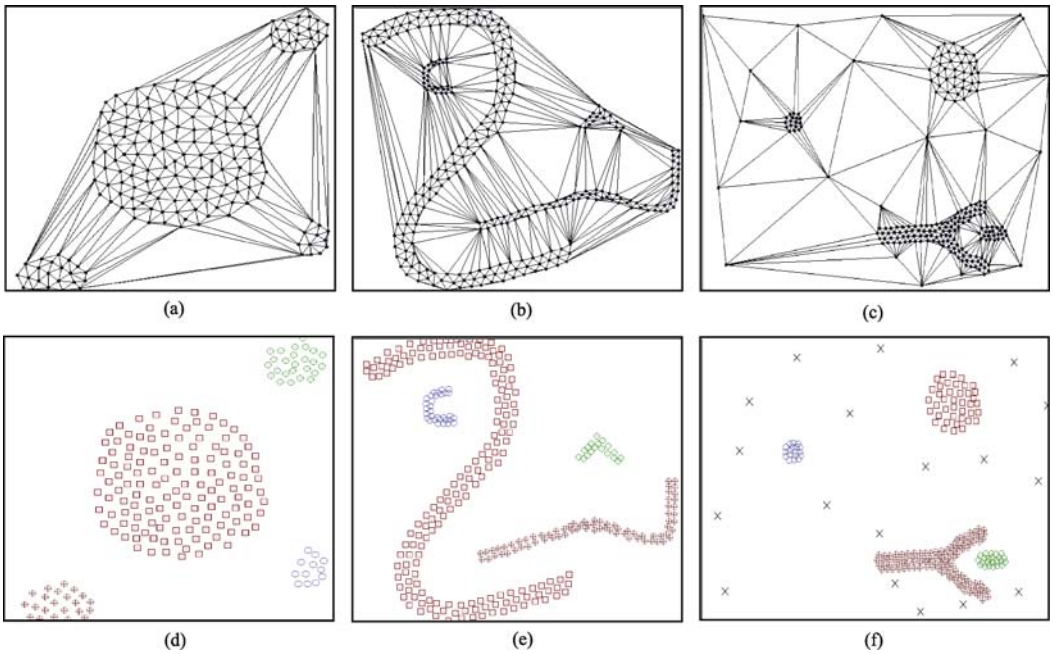

Fig. 3 Simulated databases and the clustering results obtained by FTSC algorithm (× —outlier)
(a) Databse1; (b) Database2; (c) Database3; (d) Result1; (e) Result2; (f) Result3

Comparing the clustering results obtained by DBSCAN algorithm with those by FTSC algorithm, one can find that on the one hand the DBSCAN algorithm cannot obtain satisfied clustering results in the case that the spatial entities distribute unevenly. When the value of Eps is small, the entities in the low-density regions are wrongly identified as outliers (Fig. 4(c)—(e)). With the increase of Eps, clusters are hard to be distinguished, and all the entities possibly form one cluster (Fig. 4(f) —(h)). On the other hand, all the clusters and outliers can be correctly identified by FTSC algorithm without any input parameter (Fig. 4 (b)). Through the experiment results above, it can also be found that the constraint of the aggregation force scalar does have good applicability.

3.2 Practical experiment

In order to illustrate the practicability of our FTSC algorithm, two real-world databases, the town distribution database of Yunnan province of China and the climate station database of Hunan province of China are employed in this paper. The locations of the towns and the climate stations are shown in Fig. 5(a) and (e). For comparison, the DBSCAN algorithm is also applied to these databases for spatial clustering.

At first, the FTSC algorithm is employed to discover the local aggregation pattern of the towns in Yunnan province. The clustering result by FTSC algorithm is shown in Fig. 5(b). From the clustering result, one can find that: (1) The DBSCAN algorithm

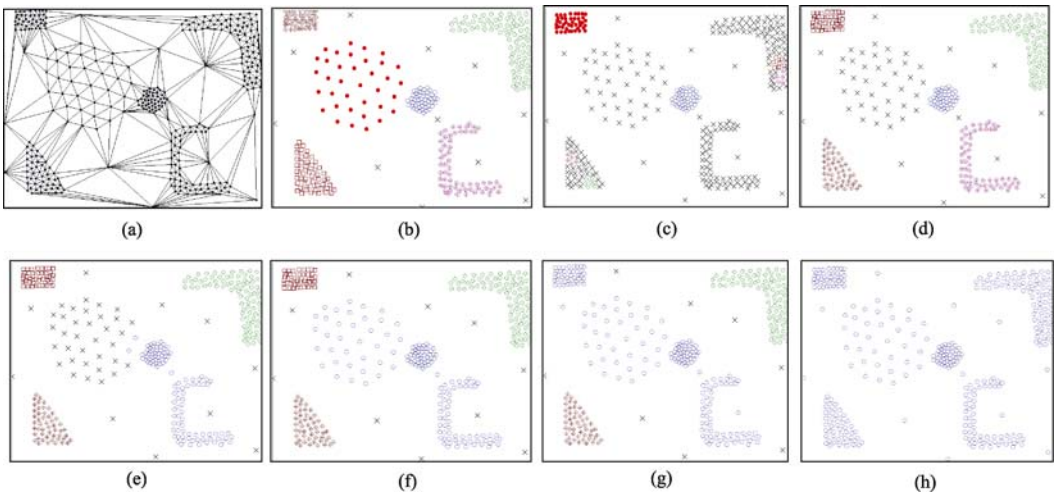

Fig. 4 Clustering results obtained by FTSC algorithm and DBSCAN algorithm (× —outlier)

(a) Database 4; (b) Result 4; (c) Eps=3, Minpts=6; (d) Eps=5, Minpts=6; (e) Eps=7, Minpts=6; (f) Eps=9, Minpts=6; (g) Eps=11—17, Minpts=6; (h) Eps=18, Minpts=6

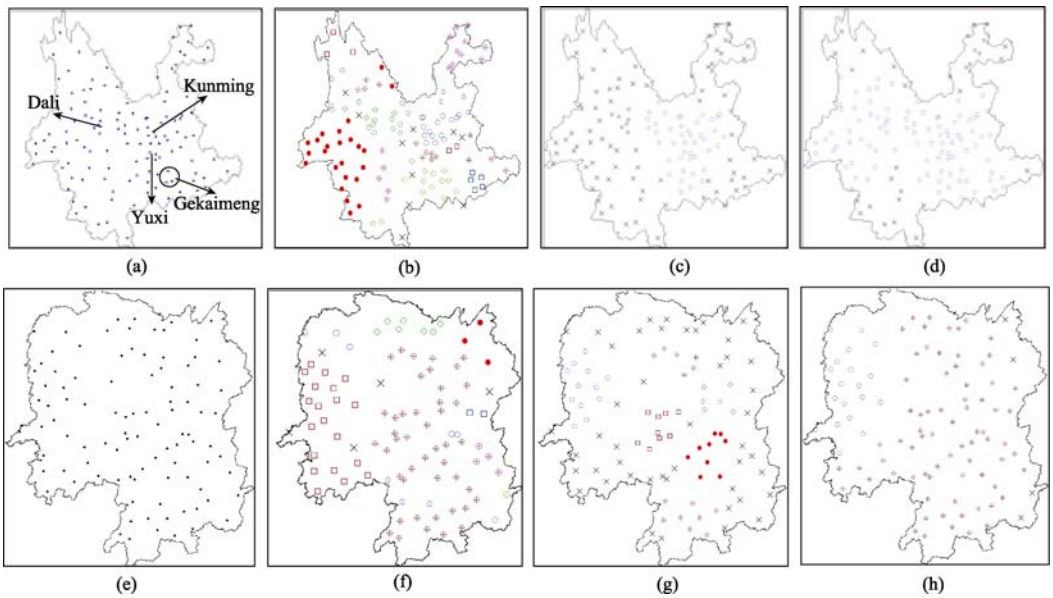

Fig. 5 Clustering results obtained by FTSC algorithm and DBSCAN algorithm (× —outlier)

(a) Towns in Yunnan province of China; (b) Result 5; (c) Eps=60km, Minpts=5; (d) Eps=80km, Minpts=5; (e) Climate stations in Hunan province of China; (f) Result 6; (g) Eps=40km, Minpts=4; (h) Eps=65km, Minpts=4

can not discover the local aggregation pattern of the towns. In contrary, several clusters the density of which are higher and the distribution of which are more uniform can be discovered through FTSC algorithm. (2) After further analyzing the clustering results obtained by FTSC algorithm, one can discover that the clusters, the densities of which are much higher mainly appear around the districts of Dali, Kunming, Yuxi and Gekaimeng (Gejiu, Kaiyuan, Mengzi). The current researches (Wu & Chen, 2007) on the development and spatial distribution of the towns in Yunnan province show that "The towns distribute dispersal on the whole but there are also several aggregation parts. The towns which developed well expand from some centers, especially the surrounding area of Dali, Kunming, Yuxi and Gekaimeng district. The level of urban development is much higher than the average condition". The clustering results by FTSC algorithm are consistent with the actual situation well.

Then, the FTSC algorithm is utilized to evaluate the layout of the climate stations in Hunan province. In order to capture the climate character of a region, the climate stations are hoped to distribute as even as possible. However, there are usually some local regions in which the climate stations are too dense or too sparse, so data from these regions may be unstable. For further spatio-temporal data analysis, these conditions should be fully considered. In this paper, the FTSC algorithm is employed to find the regions, the stations in which distribute inconsistent with the whole distribution. The clustering result generate by FTSC algorithm is performed in Fig. 5(f). From the clustering result, it can be found that (1) all the stations mainly form two large clusters, which indicates that the stations distribute evenly in the whole region. (2) However, there are also some outliers and small clusters in some local regions; one can find that the density of the stations in these regions is indeed too low or too high compared with the global condition. For further investigate, the stations in these regions may need to optimize. From the clustering results by DBSCAN shown in Fig. 5(g) and (h), the local characters of the distribution of climate stations can not be discovered.

3.3 Summary and discussion

From the above experiments, the superiority and effectively of the FTSC algorithm are fully demonstrated. Next, the relationship between the FTSC algorithm and traditional clustering algorithms will be discussed. The aggregation force is employed as the indicator to measure the similarity among spatial entities in this paper. According to the definition of the aggregation force, one can find that this indicator is consistent with the former geometrical distance measurements. However, the traditional measurements have little physical meanings; the clustering results cannot be fully explained. Here, aggregation force is a vector and it has clearly physical meaning, and the directional relationships among entities are also considered. Thus, the clustering results can be well explained. For example, when clusters are adjacent to each other, they can be clearly distinguished by the directions of the cohesive forces of the entities on the boundary of the clusters.

In addition, the multi-scale and validity measurement issues are also important parts of spatial clustering. Spatial autocorrelation and heterogeneity depend on scales, so the similarity among entities changes according to different scales. For instance, there are usually some small cities around the big cities. On large scale, all the cities may form a big cluster, but on small scale, some small cities usually construct some small clusters. In the geographical view, it is more proper to clustering from large scale to small scale, and the traditional hierarchical and graph-based algorithms may be further modified to achieve the multi-scale spatial clustering procedures. Indeed, the spatial clustering results must be well evaluated. Current clustering validity measurement approaches mainly use compactness and separation to evaluate the clustering results that if clusters are well separated and entities in each cluster are closed to each other, the clustering result is good. In this paper, the clustering results are assessed mainly according to the visualization and the prior-knowledge. In practice, a quantitative clustering validity index which can measure clustering results with arbitrary shape clusters and outliers should be further developed. Additionally, the clustering validity measurement must consider the multi-scale characteristics of the spatial entities.

4 CONCLUSIONS AND FUTURE WORK

Most existing spatial clustering methods cannot adapt to the case that the entities distribute unevenly, and more predefined parameters are required. To overcome such limitations, a field-theory based spatial clustering algorithm (i.e. FTSC algorithm) is developed in this paper. Through the simulation experiment and practical experiment, and the comparison with the classic DBSCAN algorithm, it can be concluded that: (1) the FTSC algorithm is suitable to find the clusters with arbitrary shapes, and is very robust; (2) the FTSC algorithm can automatically adapt to the change of local density; (3) the FTSC algorithm does not involve the setting of input parameters, so that it can avoid too much interference from man-made factors.

The future work will be focused on the four aspects, including: (1) to prove the setting of the constraint of the aggregation force scalar through statistics; (2) to consider the non-spatial attribute in spatial clustering; (3) to construct GRID index and develop a mixed spatial clustering algorithms so as to improve the efficiency of FTSC algorithm; (4) to develop a novel spatial clustering validity index based on the aggregation field theory.

REFERENCES

Agrawal R, Gehrke J, Gunopulos D and Raghavan P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data, Settle WA

Ankerst M, Breunig M , Kriegel H P and Sander J. 1999. OPTICS: ordering points to identify the clustering structure. Proceedings of

the 1999 ACM-SIGMOD International Conference on Management of Data. Philadelphia, PA

Bar-shalom Y and Blair W D. 2000. Multitarget-multisensor tracking: applications and advances Volume III. Artech House, Norwood, MA

Birant D and Kut A. 2007. ST-DBSCAN: an algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, **60**(1): 208—221

Blackman S and Popoli R. 1999. Design and analysis of modern tracking system. Artech House, Norwood, MA

Chen J. 2002. Dynamic Spatial Data Model Based on Voronoi. Beijing: Surveying & Mapping Press

Dave R N and Bhaswan K. 1992. Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Transactions on Neural Network*, **3**(5): 643—662

Ester M, Kriegel H P, Sander J and Xu X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd the International Conference on Knowledge Discovery and Data Mining*. Portland, OR

Estivill-Castro V and Lee I. 2000. AUTOCLUST: automatic clustering via boundary extraction for mining massive point-data sets. Proceedings of the Fifth International Conference on Geo-computation, Beijing, China

Gold C M. 1992. The meaning of "Neighbor". Theories and Methods of Spatial-Temporal Reasoning in Geographic Space, Lecture Notes in Computing Science No. 639, Berlin

Gan W Y, Li D Y and Wang J M. 2006.A hierarchical clustering method based on data fields. *Acta Electronica Sinia*, **34**(2): 258—262

Guha S, Rastogi R and Shim K. 1998. CURE: an efficient clustering algorithm for large databases. Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data. Seattle, Washington

Guha S, Rastogi R and Shim K. 1999. ROCK: a robust clustering algorithm for categorical attributes. Proceedings of the International Conference of Data Engineering. Sydney, Australia

Hofmann-wellenhof B, Lichtenegger H and Collins J. 1994. Global positioning system: theory and practice. Springer-Verlag Wien, New York

Karypis G, Han E H and Kumar V. 1999. Chameleon: hierarchical clustering using dynamic modeling. *IEEE Computer*, **32**(8): 68—75

Kovács F, Legány C and Babos, A. 2006. Cluster validity measurement techniques. Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, 88—393

Li G Q, Deng M, Cheng T and Zhu J J. 2008. A dual distance based spatial clustering method. *Acta Geodaetica et Cartographica Sinica*, **37**(4): 482—488

Li G Q, Deng M, Liu Q L and Cheng T. 2009. A spatial clustering method adaptive to local density change. *Acta Geodaetica et Cartographica Sinica*, **38**(3): 255—263

Li H M. 1999. Research on the performance of genetic algorithms and their applications in clustering analysis. Xian: Xidian University

Liu P, Zhou D and Wu N J. 2007. VDBSCAN: varied density based spatial clustering of applications with noise. Proceedings of IEEE International Conference on Service System and Service Management, Chengdu, China

Macqueen J. 1967. Some methods for classification and analysis of multivariate observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press

Mao Z Y and Li L. 2004. The Measurement of Spatial Patterns and Its Applications. Beijing : Science Press

Ng R and Han J. 1994. Efficient and effective clustering method for spatial data mining. Proceedings of the 1994 International Conference on Very Large Data Bases

Paivinen N. 2005. Clustering with a minimum spanning tree of scale-free-like structure. *Pattern Recognition Letter,* **26**(7): 921—930

Pei T, Zhu A X, Zhou C H, Li B L and Qin C Z. 2006. A new approach to the nearest-neighbor method to discover cluster features in overlaid spatial point processes. *International Journal of Geographical Information Science*, **20**(2): 153—168

Sheikholeslami G, Chatterjee S and Zhang A. 1998. Wave Cluster: a multi-resolution clustering approach for very large spatial databases. Proceedings of the 24th International Conference on Very Large Databases. New York City

Song G and Ying X. 2006. GDCIC: a grid-based density-confidence-interval clustering algorithm for multi-density dataset in large spatial databases. Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications

Wang S L.2002.Data field and cloud model based spatial data mining and knowledge discovery. Wuhan: Wuhan University

Wang W, Yang J and Muntz R. 1997. STING: a statistical information grid approach to spatial data mining. Proceedings of the 1997 International Conference on very Large Data Bases, Athens, Greece

Wright W E. 1977. Gravitational clustering. *Pattern Recognition*, **9**(3): 151—166

Wu Q Y and Chen H. 2007. Urban economic effect region spatial evolution: taking Yunnan Province as an example. *Acta Geographica Sinica*, **62**(12): 1244—1252

Zahn C T. 1971. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transaction on Computers* **C20** (1): 68—86

Zhang T, Ramakrishnan R and Livny M. 1996. BIRCH: an efficient data clustering method for very large databases. Proceedings of the International Conference Management of Data, Montreal, Canada

基于场论的空间聚类算法

邓 敏[1], 刘启亮[1], 李光强[1], 程 涛[2]

1. 中南大学 测绘与国土信息工程系, 湖南 长沙 410083;
2. 英国伦敦大学 城市、环境与地理信息工程系, 英国 伦敦

摘 要: 从空间数据场的角度出发, 提出了一种适用于空间聚类的场——凝聚场, 并给出了一种新的空间聚类度量指标(即凝聚力)。进而, 提出了一种基于场论的空间聚类算法(简称 FTSC 算法)。该算法根据凝聚力的矢量计算获取每个实体的邻近实体, 通过递归搜索的策略, 生成一系列不同的空间簇。通过模拟实验验证、经典算法比较和实际应用分析, 发现所提出的算法具有 3 个方面的优势: (1)不需要用户输入参数; (2)能够发现任意形状的空间簇; (3)能够很好适应空间数据分布不均匀的特性。

关键词: 空间聚类, 凝聚力, 场论, 空间数据挖掘

中图分类号: P208 **文献标识码:** A

引用格式: 邓 敏, 刘启亮, 李光强, 程 涛. 2010. 基于场论的空间聚类算法. 遥感学报, **14**(4): 694—709
Deng M, Liu Q L, Li G Q and Cheng T. 2010. a Field-theory based spatial clustering method. *Journal of Remote Sensing*. **14**(4): 694—709

1 引 言

空间聚类是地理信息科学与计算机科学领域共同关注的一个问题。空间聚类技术已广泛应用于地理学、制图学、地质学、遥感学、生物学、经济学等众多领域(Blackman & Popoli, 1999; Bar-shalom & Blair, 2000; Hofmann-wellenhof 等, 1994; 毛政元 & 李霖, 2004), 主要用于揭示空间数据的分布规律, 或者探测空间离群点(亦称空间异常)。

现有的空间聚类算法大致可以划分为: (1)基于划分的聚类算法, 代表算法有 k-Means(Macqueen, 1967)、k-Mediods(Ng & Han, 1994)、FCM(Dave & Bhaswan, 1992)等。基于划分的聚类算法需要首先给定聚类数目和目标函数, 然后随机选择聚类中心, 通过迭代不断降低目标函数的误差, 直到目标函数收敛至一定阈值时, 完成聚类。这类算法需要较多的先验信息来确定输入参数和收敛阈值, 并且这些参数和阈值在聚类过程中是固定的, 很难适应空间密度变化较大的情况, 聚类结果严重依赖初始聚类中心的选择, 而且不能发现任意形状的空间簇。(2)基于层次的聚类算法, 代表算法有 BIRCH(Zhang 等, 1998)、CURE(Guha 等, 1998)、ROCK(Guha 等, 1999)、CHAMELEON(Karypis 等, 1999)和基于引力的聚类算法(Wright, 1977; 淦文燕等, 2006)。基于层次的聚类算法又可以分为凝聚法和分裂法。前者从每个实体出发, 通过反复聚合, 从而得到不同层次的聚类簇; 后者对整个数据集反复进行分裂, 直至所有数据被分裂为单目标的簇, 从而得到不同层次的聚类簇。层次聚类算法采用固定的分裂或聚合度量阈值, 实质上假设了空间实体分布的均匀性。(3) 基于密度或距离的聚类算法, 代表算法有 DBSCAN (Ester 等, 1996)、VDBSCAN(Liu 等, 2007)、OPTICS (Ankerst 等, 1999)、ADBSC(李光强等, 2009)、DDBSC (李光强等, 2008)等。基于密度的聚类方法将局部密度大于给定阈值的实体聚为一类, 能够发现任意形状的簇, 具有一定的抗噪能力。但是, 基于密度的聚类方法使用的参数有时很难确定, 且这些参数在聚类过程中保持固定, 从而难以适应空间密度变化大的情况, 聚类结果易受邻域内空间离群点的影响。基于距离的聚类算法将实体间空间距离和非空间属

收稿日期: 2009-06-26; 修订日期: 2009-09-07

基金项目: 国家 863 计划项目(编号: 2009AA12Z206) ;地理空间信息工程国家测绘局重点实验室开放基金重点项目(编号: 200805)和江苏省资源环境信息工程重点实验室(中国矿业大学)开放基金项目(编号: 20080101)。

第一作者简介: 邓 敏(1974—), 男, 江西临川人, 博士, 教授, 博士生导师, 主要研究方向为时空数据挖掘、推理与分析, 发表论文 90 余篇。E-mail: dengmin208@tom.com。

性距离均小于一定阈值的实体聚为一类, 不考虑非空间属性时等同于 MinPts=1 的基于密度的聚类方法, 亦存在阈值不易确定且很难适应空间分布不均匀情况下聚类的缺陷。(4)基于图论的聚类算法, 代表算法有 ZEMST(Zahn, 1971)、SFMST(Paivinen, 2005)、AUTOCLUST(Estivill-Castro & Lee, 2000)等。基于图论的聚类算法首先在全部数据集内构建一个完全图, 每个实体视为图的一个顶点, 继而通过打断图的不一致边, 形成一系列的子图, 每个子图即视为一个簇。然而当空间分布不均匀时, 不一致边很难确定。(5)混合聚类算法, 代表算法有 STING (Wang 等, 1997)、Wave Cluster(Sheikholeslami 等, 1998)、CLIQUE(Agrawal 等, 1998)、GDCIC (Song & Ying, 2006)、NN-Density(Pei 等, 2006)等。该类算法综合采用多种类型算法进行聚类, 其中采用格网结构对空间实体进行储存, 进而结合其他聚类算法进行混合聚类是最常见的形式, 这种算法最主要的优点是减少全局遍历, 从而提高了算法的效率。然而, 混合聚类算法仍然具有上述几种聚类算法中存在的缺陷, 如最常用的格网-密度混合聚类算法只是对高密度区域较为敏感, 很难适应空间局部密度差异较大情况下的聚类。此外, 在某些情况下, 混合聚类算法的聚类质量降低, 如 STING 算法。

综上所述, 现有空间聚类算法很难适应空间数据分布差异较大的情况, 同时聚类结果多依赖于聚类参数的选择, 影响了其实用性。为此, 本文首先引入数据场概念对空间聚类问题进行描述, 继而发展一种适用于空间聚类的场, 并给出聚类度量指标, 在此基础上提出了一种基于场论的空间聚类算法。

2　基于场论的空间聚类算法原理及描述

空间聚类是将空间数据库中的空间实体划分成具有一定意义的若干簇, 使得每个簇内实体具有最大相似度, 而簇间实体差别最大。聚类“相似度”和“差别”大多是采用空间实体间的距离来度量(Kovács 等, 2006), 缺乏实际的物理意义。本文受物理学中场论思想的启发, 从数据场的角度对空间聚类问题进行解释, 提出一种适用于空间聚类的数据场, 即凝聚场。

2.1　凝聚场与凝聚力

数据通过数据辐射将其数据能量从样本空间辐射到整个母体空间, 接受数据能量并被数据辐射所覆盖的空间, 叫做数据场(王树良, 2002)。广义上讲, 凝聚场也是一种数据场, 本文将其定义为一种有源矢量场, 具体描述为: (1)空间中每个实体均视为一个具有单位质量的质点, 亦可为一个凝聚场源(简称场源); (2)每个场源在其周围一定范围内产生一个虚拟作用场, 本文称之为凝聚场; (3)位于凝聚场内的任何其他空间实体均将受到场源实体的内向引力作用, 这种引力称为凝聚力。基于凝聚场理论, 空间聚类的机理和过程可以描述为: (1)凝聚场内所有实体受到的凝聚力标量和越大, 则表明场源的“吸附”能力越强。(2)每个空间簇的形成均是从一个“吸附”能力较强的场源开始, 各实体不断向外“吸附”其他实体, 最终形成一个簇。(3)同一个簇内实体间的凝聚力作用较强, 而不同簇内的实体之间凝聚力的作用较弱。显然, 离群点受到的凝聚力较小。

凝聚场的空间分布规律可以用矢量场强函数进行描述。对于空间聚类而言, 短程场作用更有利于揭示数据分布的聚簇特性, 即凝聚场的场强在有限的影响范围内迅速衰减。淦文燕等(2006)提出了选用衰减较快标量势函数及影响因子的方法, 然而影响因子的选择非常困难, 从而增加了聚类的难度。为此, 本文借助 Voronoi 图和 Delaunay 三角网对空间进行划分, 继而构造了凝聚场的场强函数表达形式。为了便于理解和描述, 下面给出一些定义。

定义 1　二维 Voronoi 图(陈军, 2002): 给定空间实体集合 $\boldsymbol{P}$, $\boldsymbol{P}=\{p_1, p_2, p_3, \ldots, p_n\}$, 对于 $\boldsymbol{P}$ 中任一点 p_i, 其对应的 Voronoi 多边形 p_i^V 可以定义为:

$$p_i^V=\{x \mid \mathrm{d}(x, p_i) \leqslant \mathrm{d}(x, p_j), p_i, p_j \in \boldsymbol{P}, i\neq j, x\in \mathrm{R}^2\} \quad (1)$$

式中, d 表示欧氏距离函数。$\boldsymbol{P}$ 中所有点的 Voronoi 多边形构成点集 $\boldsymbol{P}$ 的 Voronoi 图(图 1 实线), 表达为:

$$\boldsymbol{P}^{\boldsymbol{V}}=\{p_1^V, p_2^V, p_3^V, \ldots, p_n^V\} \quad (2)$$

$\boldsymbol{P}^{\boldsymbol{V}}$ 的对偶图为 Delaunay 三角网, 记为 $\boldsymbol{D}(\boldsymbol{P})$, 如图 1 虚线所示。

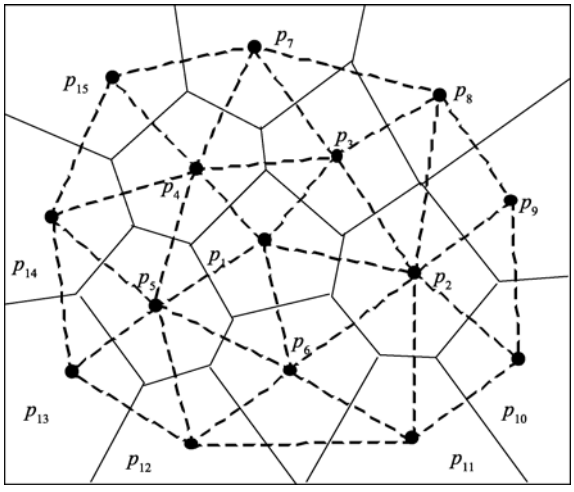

图 1　凝聚场

定义 2 直接 Voronoi 邻近实体(Gold, 1992): 给定空间实体集合 $\boldsymbol{P}$, $p_i, p_j \in \boldsymbol{P}$, 若 p_i^V 和 p_j^V 具有公共 Voronoi 边, 则 p_i 和 p_j 互为直接 Voronoi 邻近实体。实体 p_i 的所有直接邻近实体即为 p_i 的 Delaunay 邻近实体, 表示为 $\boldsymbol{ND}(p_i)$。如图 1, p_1 点的直接邻近实体为 p_2, p_3, p_4, p_5 和 p_6。

定义 3 直接 Voronoi 区域: 对于空间实体 p_i, p_i^V 与 p_i 的直接 Voronoi 邻近实体对应的 Voronoi 多边形所构成的区域称为 p_i 的直接 Voronoi 区域, 记为 $\boldsymbol{DNV}(p_i)$。如图 1, p_1 的直接 Voronoi 区域包括 $p_1^V, p_2^V, p_3^V, p_4^V, p_5^V, p_6^V$。

二维 Voronoi 图可以视为以平面内每个点作为生长核, 以相同速率向外扩张, 直到彼此相遇后停止生长, 反映了实体天然的“势力范围”。根据这种特性, 可以表达凝聚场的场强函数为:

$$E_p = k\frac{1}{\mathrm{d}(p,x_i)^{2\sigma}}\boldsymbol{e}_{px_i}, \sigma = \begin{cases} 1, x_i \in \boldsymbol{DNV}(p) \\ +\infty, x_i \notin \boldsymbol{DNV}(p) \end{cases} \quad (3)$$

式中: E_p 为场源(亦是一个实体)p 在空间上产生的凝聚场的场强; k 为凝聚场辐射因子, 本文设置为 1; x_i 为任意一个空间位置; $\mathrm{d}(p, x_i)$为实体 p 与 x_i 的欧氏距离; σ 为衰减因子; $\boldsymbol{e}_{px_i}$ 为 p 指向 x_i 的单位矢量。

从式(3)可以看出, 一个场源产生的凝聚场, 在其直接 Voronoi 区域内, 场强函数的衰减满足距离平方反比关系; 而在空间其他的区域场强函数迅速衰减, 迅速达到可以忽略的程度。同时, 对凝聚场的假设也满足数据场的基本特征, 即必须同时满足独立性、就近性、遍历性、叠加性、衰减性和各向同性等条件(王树良, 2002)。进而, 可以表达两个实体间的凝聚力为:

$$\begin{aligned} F_C(p,q) &= E_p m_q = k\frac{1}{\mathrm{d}(p,q)^{2\sigma}} m_q \boldsymbol{e}_{pq} \\ &= \frac{m_q}{\mathrm{d}(p,q)^{2\sigma}}\boldsymbol{e}_{pq}, \sigma = \begin{cases} 1, q \in \boldsymbol{ND}(p) \\ +\infty, q \notin \boldsymbol{ND}(p) \end{cases} \end{aligned} \quad (4)$$

式中, m_q 为实体 q 的质量, 考虑到可以将空间点实体均视为单位质点, 故令 m_q 为 1; $\mathrm{d}(p, q)$为实体 p 与 q 的欧氏距离; σ 表示衰减因子; $\boldsymbol{e}_{pq}$ 表示 p 指向 q 的单位矢量。

分析式(4)发现, 一个场源只对其 Delaunay 邻近实体有明显的凝聚力作用, 对其他空间实体的作用可以忽略。此外, 与 Wright(1977)、淦文燕(2006)等采用的“引力”概念不同, 本文采用的凝聚力是一种矢量, 并顾及了力的方向特性, 这亦是与传统聚类度量指标最大的区别。

2.2 基于场论的空间聚类算法原理

根据 2.1 节的假设, 基于场论的空间聚类算法包括两个核心步骤: (1)每个实体的邻近实体的获取; (2)从“吸附”能力作用较强的场源开始, 各实体不断向外“吸附”其他邻近实体, 最终生成空间簇。下面简要阐述这两个步骤的基本原理。

获取场源邻近实体实质上是确定与场源凝聚力较大的若干实体。由于凝聚力是一种矢量, 根据作用力与反作用力的原理, 场源“吸引”凝聚场域内其他实体的同时, 其他实体也会对场源产生一个反作用力, 场源所受凝聚力的合力方向一般指向局部的一个聚类中心(李海民, 1999), 即场源对该合力的反方向实体的“吸附”能力最强。凝聚场中各实体对场源的凝聚力合力表达为:

$$F_T = \sum F_C(p,q), q \in \boldsymbol{ND}(p) \quad (5)$$

若某一实体对场源的凝聚分力与合力的夹角小于 90°, 则分力对合力有增强的作用, 可以认为场源能对其有“吸附”作用, 该实体极有可能成为场源的邻近实体。如图 2(a), 虚线箭头表示合力 F_T, 显然 $F_C(A,B)$, $F_C(A,C)$, $F_C(A,D)$与 F_T 夹角小于 90°, 则 A 可以对 B, C, D 进行“吸附”; $F_C(A,E)$ 与 F_T 夹角大于 90°, 则 A 点无法对 E 进行“吸附”, 因此 B, C, D 更有可能为 A 的邻近实体。

然而, 上述判别方式没有顾及一种特殊情况, 即“边界效应”。如图 2(b), 以 A 点为场源计算聚类场内合力方向为虚线箭头所指方向, 并且 4 个分力与合力夹角都小于 90°。然而 D, E 两点明显与 A 点不邻近, 这是由于 A 点位于簇的边界上, 其他簇内的实体对其产生了干扰。因此, 下面进一步给出一个凝聚力标量的约束条件, 提出一种三步法判别邻近实体的方法, 描述为:

(1) 给定场源 p, 首先判别候选邻近实体, 即所有对场源的凝聚力与场源所受凝聚力合力的夹角小

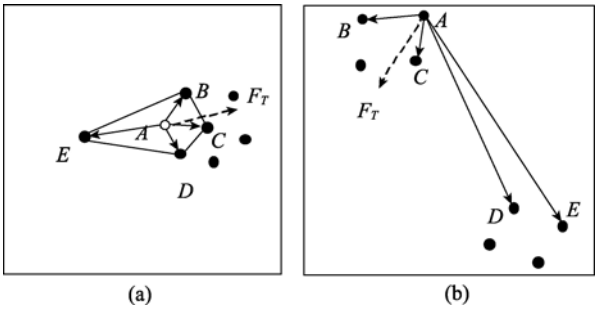

图 2 邻近实体获取

(a) 凝聚力矢量计算简例; (b) 边界效应简例

于 90°的实体均被视为候选邻近实体, 记为 $\boldsymbol{CNS}(p)$, 表达为:

$$\boldsymbol{CNS}(p)=\{q \mid \theta(F_C(p,q),F_T)<90°, q\in\boldsymbol{ND}(p)\} \quad (6)$$

式中, $\theta(F_C(p,q),F_T)$表示各分力与合力的夹角。

(2) 平均凝聚力, 即在候选邻近实体集合中获得凝聚力的平均大小。为了避免少部分极小值的干扰, 首先除去最小的[N/2]个力后, 再计算其他力的平均值, 记为 $E_P(F)$, 其中[]表示取整操作, N 为候选邻近实体数量。

(3) 邻近实体获取。由于其他簇内实体对边界点的凝聚力远小于簇内实体对其产生的凝聚力, 如图 2(b), D、E 点对 A 的凝聚力远小于 B、C 点。于是, 本文在大量实验的基础上给出了一个标量约束条件: 若凝聚力标量大小超过平均凝聚力的 1/5, 则将其视为场源的邻近实体, 记为 $\boldsymbol{NS}(p)$, 表达为:

$$\boldsymbol{NS}(p)=\{q \mid \mid F_C(p,q)\mid >E_P(F)/5, q\in\boldsymbol{CNS}(p)\} \quad (7)$$

式中, $\mid F_C(p,q)\mid$ 表示分力的标量大小。

经过以上计算和识别过程, 如图 2(a)中 A 的邻近实体为 B, C, D, 很好地与 E 点进行了分离; 而(b)中 A 点的邻近实体为 B, C, 排除了 D, E 点的干扰, 达到了获取场源邻近实体的目的。进而, 另一个重要步骤在于从“吸附”能力较强场源开始, 不断向外“吸附”邻近实体, 生成空间簇。直观上, 一个场源对其 Delaunay 邻近实体的凝聚力标量和越大, 则可以认为其“吸附”能力越强。其中, 凝聚力标量和计算表达式为:

$$\left|F_T\right|=\sum\left|F_C(p,q)\right|, q\in\boldsymbol{ND}(p) \quad (8)$$

式中, $\mid F_C(p,q)\mid$ 为凝聚力的标量大小。本文将“吸附”能力较强的场源称为聚类核, 并进行如下定义:

定义 **4**　聚类核: 给定空间实体集合 $\boldsymbol{P}$, $\boldsymbol{P}$ 中每个实体凝聚力标量和构成集合 $\boldsymbol{F}(\boldsymbol{P})$, 凝聚力标量和最大的实体即为聚类核, 记为 $Core(\boldsymbol{P})$, 表示为:

$$Core(\boldsymbol{P})=p, SF(p)=\max(\boldsymbol{F}(\boldsymbol{P})) \quad (9)$$

进而, 基于场论的空间聚类算法的基本过程可以描述为:

①在每个实体产生的凝聚场中分别获取其邻近实体。

②选取聚类核, 各实体不断向外“吸附”其邻近实体, 直到生成一个簇; 只有一个实体的簇, 标记为异常点。

③如果仍有实体未加入簇中或未被标记为异常点, 则重复步骤②。

空间实体一旦加入某个簇中, 即从 $\boldsymbol{P}$ 中退出, 以保证每个簇生成时是从当前实体中选凝聚力标量和最大的点作为聚类核。根据凝聚力的表达形式可知, 一个场源的凝聚力标量和越大, 则其与邻近实体的距离越短, 故空间密度越大。于是, 空间密度大的实体最先生成簇。最终在整个空间中, 由高密度到低密度依次生成一系列的簇, 故可以自动适应空间局部密度的变化。为了避免生成过于松散的空间簇, 可以对聚类核进行限制, 即: 凝聚力标量和极小的实体(与平均值之差的绝对值大于三倍标准差), 不将其作为聚类核。

2.3　算法描述

根据上述基本原理, 给定一个空间数据库 $\boldsymbol{SDB}$, 基于场论的空间聚类算法(即 FTSC 算法)可以描述如下:

① 在整个数据集中生成 Delaunay 三角网, 获得每一实体 p_i 的 Delaunay 邻近域实体集合; 同时计算每个实体的凝聚力的标量和, 获得凝聚力标量和集合;

② 针对 $\boldsymbol{SDB}$ 中任一点 p_i 获得其邻近实体集合;

③ 选取凝聚力标量和最大的实体 p_x, 将其作为聚类核, 将 p_x 与其邻近实体聚为一类, 存入集合 $\boldsymbol{C}$ 中并进行标记, 标明 p_x 已进行“吸附”操作;

④ 针对 $\boldsymbol{C}$ 中未进行“吸附”操作的任一实体 p_j, 将其邻近实体不断加入 C 中, 直到 $\boldsymbol{C}$ 中所有实体均已进行“吸附”操作为止, 一个簇生成结束;

⑤ 统计 $\boldsymbol{C}$ 中实体的数目 $Cnum$。如果 $Cnum<2$, 则将 C 中实体标记为异常点;

⑥ 如果 $\boldsymbol{SDB}$ 中仍有实体未被加入簇中或被标记为异常点, 则搜索下一个聚类核, 重复③—⑤步骤; 否则, 返回聚类结果。

3　实验验证与分析

设计两个实验来验证本文所提出的 FTSC 算法的正确性。实验 1 采用模拟数据, 共包含 4 组数据, 其中模拟了 Ester 等(1996)使用的 3 组经典数据库, 并另外设计了 1 组数据; 实验 2 采用 FTSC 算法针对两组实际地理空间数据进行实际应用分析。2 个实验的结果分别与经典的 DBSCAN 算法进行比较。

3.1　模拟算例分析与比较

采用 Ester 等(1996)使用的 3 组数据库来验证 FTSC 算法可以发现任意形状的簇, 并且具有抗噪性。图 3(a)—(c)给出 3 个数据库的数据分布情况及各自生成的 Delaunay 三角网; 图 3(d)—(f)给出了 FTSC 算法的聚类结果。

与 DBSCAN 算法(Ester 等, 1996)聚类结果比较,

图 3(d)的聚类结果与 DBSCAN 算法在 Eps=4—9, Minpts=1—11 时的聚类结果一致, 图 3(e)的聚类结果与 DBSCAN 算法在 Eps=4—6, Minpts=1—17 时的聚类结果一致。图 3(f)所示数据库 3 的聚类结果与 DBSCAN 算法在 Eps=4, Minpts=2—6 时的聚类结果一致。此外, 通过实验发现: (1) FTSC 算法能够探测任意形状的簇; (2) FTSC 算法对于靠近簇的异常点很敏感, 而且可以区分非常接近的簇; 而 DBSCAN 算法则需要非常严格的参数设定才能达到相同的效果。例如, 对于数据库 3, DBSCAN 算法的参数限制非常严格, 尤其是 Eps 的设定, 稍微增大则无法区分相邻的簇或离簇较近的异常点。

为了验证 FTSC 算法能够自动适应空间分布不均匀情况下的聚类, 本文设计了另一组数据(即数据库 4)进行测试。如图 4(a), 模拟数据共包含了 336 个数据点, 预设 6 个空间簇和 11 个异常点(图中用符号“×”标记)。这组数据密度变化较大, 形状各异, 同时包含了不同密度区域邻接的情况。图 4(b)为 FTSC 算法的聚类结果。为了便于比较, 图 4(c)—(h)给出了 DBSCAN 算法取不同参数时的聚类结果, 其中 MinPts 采用了 Birant 和 Kut(2007)给出的最优设置, 即 MinPts=ln(336)≈6。

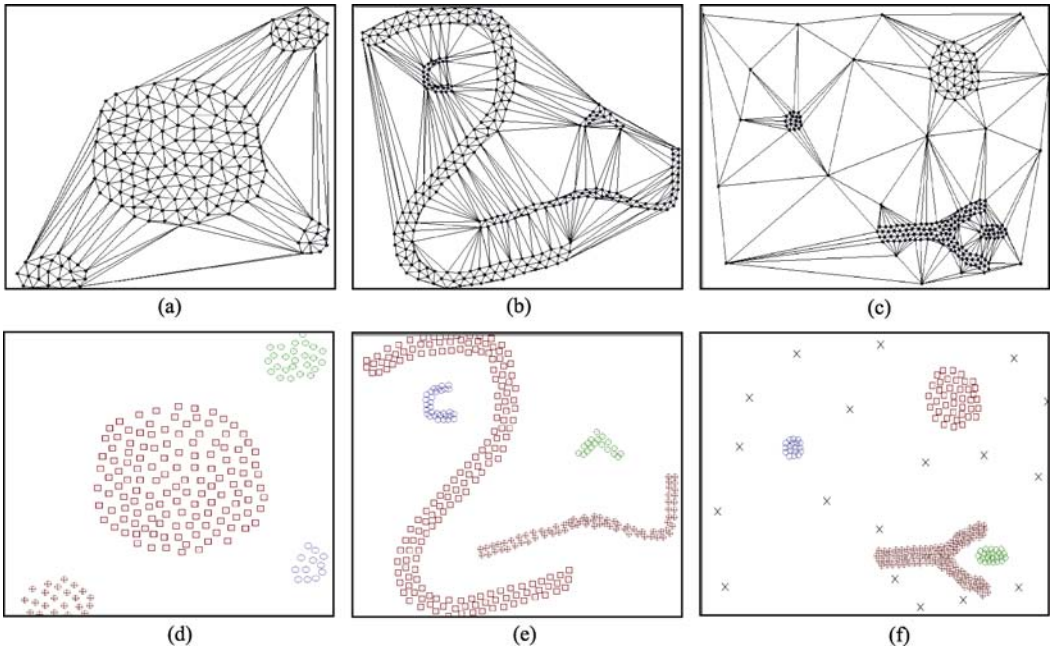

图 3 模拟数据库及 FTSC 算法聚类结果（×为异常点）

(a)数据库 1; (b) 数据库 2; (c) 数据库 3; (d) 结果 1; (e) 结果 2; (f) 结果 3

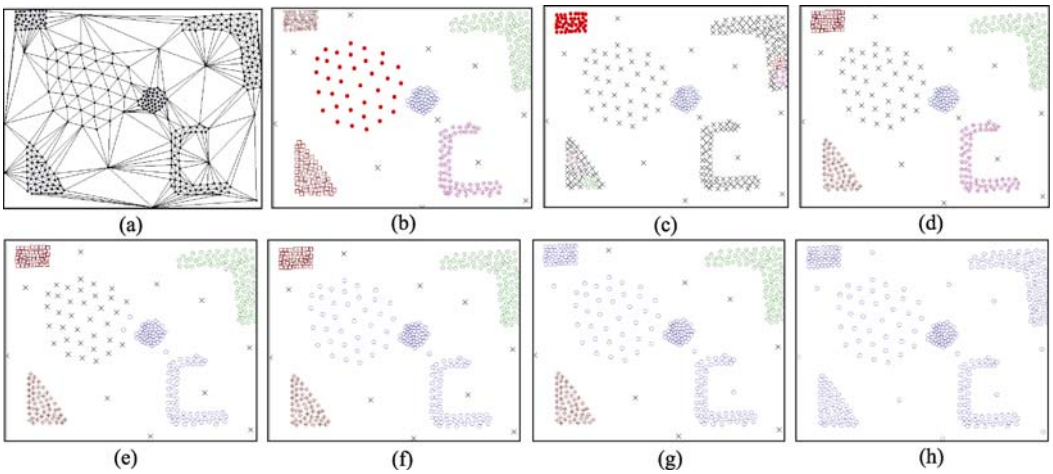

图 4 FTSC 算法与 DBSCAN 算法聚类结果比较（×为异常点）

(a) 数据库 4; (b) 结果 4; (c)Eps=3, Minpts=6; (d) Eps=5, Minpts=6; (e) Eps=7, Minpts=6; (f)Eps=9, Minpts=6; (g) Eps=11—17, Minpts=6; (h) Eps=18, Minpts=6

分析图 4 中 DBSCAN 的聚类结果发现: DBSCAN算法对于空间分布不均匀或不同密度的簇邻接情况下的聚类效果很差, 当 Eps 设定过小时, 低密度区域的点易被判为异常点, 如图 4(c)—(e); 随着Eps的增大, 不同密度的空间簇很难进行区分, 直到所有实体聚为一类, 如图 4(f)—(h), 并且最多只能正确区分 3 个簇, 如图 4(f)。而 FTSC 算法对各个簇以及异常点能较好地区分。通过上述实验表明, 本文选取的凝聚力标量约束条件具有较好的普适性。

3.2　实际算例分析与比较

实验使用 FTSC 算法分别进行城市空间分布的集聚模式挖掘以及气象站点选址优化的实际应用研究。实验数据分别为云南省 126 个县级城市空间数据和湖南省 96 个气象站点空间数据, 如图 5(a)和图 5(e)所示。针对云南省县级城市空间数据, FTSC 算法的聚类结果如图 5(b), 相应 DBSCAN 算法的聚类结果如图 5(c)和图 5(d)。针对湖南省气象站数据, FTSC算法的聚类结果如图5(f), 相应DBSCAN算法的聚类结果如图 5(g)和图 5(h)。限于篇幅, 本文只给出了 DBSCAN 算法具有代表性的聚类结果, 其中MinPts=ln(n), n 为空间实体数量。

分析云南省县级城市集聚模式的挖掘结果发现: (1)DBSCAN 算法难以适应实际空间数据分布不均匀的特性, 不能很好地发现城市的局部集聚模式, 而 FTSC 算法则可以发现不同密度的空间簇, 且各个簇内实体的分布较为均匀; (2)进一步分析 FTSC 算法聚类获得的各个簇, 可以发现实体较多且密度相对较大的空间簇主要集中在大理、昆明、玉溪、个开蒙(个旧、开远、蒙自)地区周边, 这些簇中的实体在空间上表现为局部聚集的趋势, 反映出城市的发展状况, 可以认为这些区域的城市化水平相对较高, 城市较为密集。当前云南省城市发展状况及空间分布已有的研究成果表明(吴启焰等, 2007): “云南省城市分布趋向大分散、小集中的格局, 发展较好的小城市主要从几个中心向外扩延, 特别是昆明市、个开蒙地区、玉溪市、大理市周边地区, 城市发展水平远高于全省城镇发展水平”。由此可见, 本文空间聚类分析的结果与实际情况非常吻合。

气象站在布局设置时一般要求其空间分布要尽可能均匀, 然而实际上在某些局部经常会出现一些过分稀疏或密集的区域, 因此对这些区域进行优化设置在气象研究中具有重要意义。分析 FTSC 算法空间聚类结果可以发现: (1)所有站点整体上主要形成两个较大的空间簇, 说明气象站点总体分布比较均匀, 符合气象站点空间布局的基本要求; (2)在局部发现了数个空间离群点和小簇, 这些区域的气象站点分布相对过于稀疏或密集, 不利于气象因子的恢复, 因而应考虑调整气象站布局(如增设气象站点)进行局部优化。此外, 采用这些气象站点采集的数

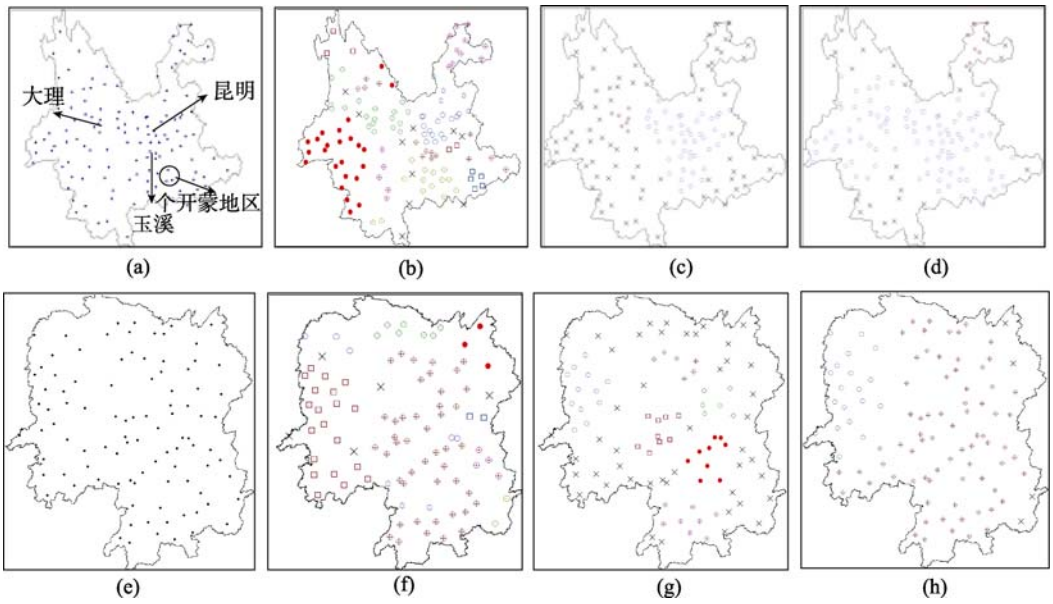

图 5　FTSC 算法与 DBSCAN 算法聚类结果比较 (× —异常点)

(a) 云南省县级城市空间分布; (b) 结果 5; (c) Eps=60km, Minpts=5; (d) Eps=80km, Minpts=5; (e) 湖南省气象站点空间分布; (f) 结果 6; (g) Eps=40km, Minpts=4; (h) Eps=65km, Minpts=4

据进行气象因子恢复时可能会有较大的误差，在进行深层次的时空数据分析时需要顾及。而 DBSCAN 算法的聚类结果难以反映气象站点分布的局部信息，对于气象站点空间分布优化的指导价值很有限。

3.3 实验结果与讨论

通过模拟算例与实际算例分析以及与 DBSCAN 算法的比较，充分验证了 FTMS 算法在进行空间模式挖掘方面的有效性和优越性。同时可以进一步分析本文提出的 FTMS 算法与传统的空间聚类算法的联系与区别。FTMS 算法采用凝聚力来度量空间聚类中的相似性问题，即空间实体通过相互间力的吸引作用而聚集成簇，从凝聚力的定义中可以发现，这种相似性度量准则也是与距离密切相关的，这一点与传统的空间聚类相似性度量方法的作用是一致的。也就是说，传统聚类中簇内实体的相似性在空间聚类中体现的是空间相关性，即空间聚类是对空间数据依据其空间相关性进行划分，同一个簇内的实体其空间相关性要尽可能大。但是，凝聚力将实体之间距离的关系转化为矢量力的关系，具有明确的物理意义，顾及了实体间的方向关系，易于对空间聚类的结果进行解释。尤其是，两个空间簇邻接时的区分问题采用凝聚力的矢量关系进行度量，这个难点问题就迎刃而解了，这主要是因为不同簇中实体受到凝聚力合力的方向具有较强的可区分性。

此外，空间聚类的多尺度与有效性评价问题虽然在本文中没有具体涉及，但也是空间聚类的两个重要研究内容。空间相关性是依赖尺度变化的，故在不同的尺度上空间聚类的相似性程度判断将有所区别，如在大城市周围可能分布若干卫星城市，在大尺度上所有的城市构成一个空间簇；而随着尺度减小，一些卫星城市可能又会构成一些更为细致的空间簇。因此，空间聚类的多尺度问题应该采取一种“自上而下”的策略进行考虑，从大尺度到小尺度进行层次聚类。对于空间聚类有效性的评价，现有的聚类有效性评价指标多是从簇内实体的紧实度和簇之间的分离程度进行度量，即簇内实体越紧密，簇之间分离度越大，则聚类效果越好(Kovács 等，2006)。而本文在很大程度上仅是从视觉感知以及根据先验知识对聚类结果的有效性进行评价的。在实际应用中，对聚类结果进行有效性评价通常还需要进一步考虑空间数据的多尺度特征，因为在不同尺度上进行空间聚类得到的结果可能代表着不同的空间格局或模式。

4 结论与展望

针对已有空间聚类算法大多需要输入参数，人为干预多且不适应空间数据不均匀分布的局限，本文首先从空间数据场的角度对空间聚类问题进行解释，发展了一种适用于空间聚类的凝聚场。在此基础上，提出了一种基于场论的空间聚类算法——FTSC。通过模拟实验和实际数据验证，以及与经典的 DBSCAN 算法进行比较发现：(1)FTSC 算法可以发现任意形状的空间簇，可以有效发现空间异常点；(2)FTSC 算法具有自适应性，可以处理空间分布不均匀、空间簇邻接等复杂情况下的聚类；(3)FTSC 算法不需要用户输入参数，避免了过多人为因素的干扰，实用性强。

本文的进一步研究工作主要包括：(1)利用统计学方法确定凝聚力标量约束条件并加以证明，有利于本文所提聚类方法的进一步完善。本文对凝聚力标量约束条件的设置是建立在大量实验的基础上，并没有给出严密的理论证明；(2)研究顾及非空间属性的聚类方法，拓展空间聚类的应用范围；(3)构建 GRID 索引，发展一种混合聚类算法，提高算法的运行效率，以适用于海量空间数据库；(4)研究基于场论的空间聚类有效性评价方法。

REFERENCES

Agrawal R, Gehrke J, Gunopulos D and Raghavan P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data, Settle WA

Ankerst M, Breunig M , Kriegel H P and Sander J. 1999. OPTICS: ordering points to identify the clustering structure. Proceedings of the 1999 ACM-SIGMOD International Conference on Management of Data. Philadelphia, PA

Bar-shalom Y and Blair W D. 2000. Multitarget-multisensor tracking: applications and advances Volume III. Artech House, Norwood, MA

Birant D and Kut A. 2007. ST-DBSCAN: an algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, **60**(1): 208－221

Blackman S and Popoli R. 1999. Design and analysis of modern tracking system. Artech House, Norwood, MA

Chen J. 2002. Dynamic Spatial Data Model Based on Voronoi. Beijing : Surveying & Mapping Press

Dave R N and Bhaswan K. 1992. Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Transactions on Neural Network*, **3**(5): 643－662

Ester M, Kriegel H P, Sander J and Xu X. 1996. A density-based

algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd the International Conference on Knowledge Discovery and Data Mining*. Portland, OR

Estivill-Castro V and Lee I. 2000. AUTOCLUST: automatic clustering via boundary extraction for mining massive point-data sets. Proceedings of the Fifth International Conference on Geo-computation, Beijing, China

Gold C M. 1992. The meaning of "Neighbor". Theories and Methods of Spatial-Temporal Reasoning in Geographic Space, Lecture Notes in Computing Science No. 639, Berlin

Gan W Y, Li D Y and Wang J M. 2006.A hierarchical clustering method based on data fields. *Acta Electronica Sinia*, **34**(2): 258—262

Guha S, Rastogi R and Shim K. 1998. CURE: an efficient clustering algorithm for large databases. Proceedings of 1998 ACM-SIGMOD International Conference on Management of Data. Seattle, Washington

Guha S, Rastogi R and Shim K. 1999. ROCK: a robust clustering algorithm for categorical attributes. Proceedings of the International Conference of Data Engineering. Sydney, Australia

Hofmann-wellenhof B, Lichtenegger H and Collins J. 1994. Global positioning system: theory and practice. Springer-Verlag Wien, New York

Karypis G, Han E H and Kumar V. 1999. Chameleon: hierarchical clustering using dynamic modeling. *IEEE Computer*, **32**(8): 68—75

Kovács F, Legány C and Babos, A. 2006. Cluster validity measurement techniques. Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, 88—393

Li G Q, Deng M, Cheng T and Zhu J J. 2008. A dual distance based spatial clustering method. *Acta Geodaetica et Cartographica Sinica*, **37**(4): 482—488

Li G Q, Deng M, Liu Q L and Cheng T. 2009. A spatial clustering method adaptive to local density change. *Acta Geodaetica et Cartographica Sinica*, **38**(3): 255—263

Li H M. 1999. Research on the performance of genetic algorithms and their applications in clustering analysis. Xian: Xidian University

Liu P, Zhou D and Wu N J. 2007. VDBSCAN: varied density based spatial clustering of applications with noise. Proceedings of IEEE International Conference on Service System and Service Management, Chengdu, China

Macqueen J. 1967. Some methods for classification and analysis of multivariate observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press

Mao Z Y and Li L. 2004. The Measurement of Spatial Patterns and Its Applications. Beijing : Science Press

Ng R and Han J. 1994. Efficient and effective clustering method for spatial data mining. Proceedings of the 1994 International Conference on Very Large Data Bases

Paivinen N. 2005. Clustering with a minimum spanning tree of scale-free-like structure. *Pattern Recognition Letter,* **26**(7): 921—930

Pei T, Zhu A X, Zhou C H, Li B L and Qin C Z. 2006. A new approach to the nearest-neighbor method to discover cluster features in overlaid spatial point processes. *International Journal of Geographical Information Science*, **20**(2): 153—168

Sheikholeslami G, Chatterjee S and Zhang A. 1998. Wave Cluster: a multi-resolution clustering approach for very large spatial databases. Proceedings of the 24th International Conference on Very Large Databases. New York City

Song G and Ying X. 2006. GDCIC: a grid-based density-confidence-interval clustering algorithm for multi-density dataset in large spatial databases. Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications

Wang S L.2002.Data field and cloud model based spatial data mining and knowledge discovery. Wuhan: Wuhan University

Wang W, Yang J and Muntz R. 1997. STING: a statistical information grid approach to spatial data mining. Proceedings of the 1997 International Conference on Very Large Data Bases, Athens, Greece

Wright W E. 1977. Gravitational clustering. *Pattern Recognition*, **9**(3): 151—166

Wu Q Y and Chen H. 2007. Urban economic effect region spatial evolution: taking Yunnan Province as an example. *Acta Geographica Sinica*, **62**(12): 1244—1252

Zahn C T. 1971. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transaction on Computers* **C20** (1): 68—86

Zhang T, Ramakrishnan R and Livny M. 1996. BIRCH: an efficient data clustering method for very large databases. Proceedings of the International Conference Management of Data, Montreal, Canada

附中文参考文献

陈军. 2002. Voronoi 动态空间数据模型. 北京: 测绘出版社

淦文燕, 李德毅, 王建民. 2006. 一种基于数据场的层次聚类算法. 电子学报, **34**(2): 258—262

李光强, 邓敏, 程涛, 朱建军. 2008. 一种基于双重距离的空间聚类算法. 测绘学报, **37**(4): 482—488

李光强, 邓敏, 刘启亮, 程涛. 2009. 一种适应局部密度变化的空间聚类方法. 测绘学报, **38**(3): 255—263

李海民. 1999. 遗传算法性能及其在聚类分析中的应用研究. 西安: 西安电子科技大学博士学位论文

毛政元, 李霖. 2004. 空间模式的测度及其应用. 北京: 科学出版社

王树良. 2002. 基于数据场与云模型的空间数据挖掘和知识发现. 武汉: 武汉大学博士学位论文

吴启焰, 陈浩. 2007. 云南城市经济影响区空间组织演变规律. 地理学报, **62**(12): 1244—1252