Pattern Recognition 44 (2011) 1721–1737

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Molecular dynamics-like data clustering approach

Li Junlin*, Fu Hongguang

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

ARTICLE INFO

Article history:
Received 7 July 2010
Received in revised form
7 December 2010
Accepted 14 January 2011
Available online 1 February 2011

Keywords:
Molecular dynamics
Dynamics clustering
Data mining
Data clustering

ABSTRACT

Based on the molecular kinetic theory, a molecular dynamics-like data clustering approach is proposed in this paper. Clusters are extracted after data points fuse in the iterating space by the dynamical mechanism that is similar to the interacting mechanism between molecules through molecular forces. This approach is to find possible natural clusters without pre-specifying the number of clusters. Compared with 3 other clustering methods (trimmed *k*-means, JP algorithm and another gravitational model based method), this approach found clusters better than the other 3 methods in the experiments.

1. Introduction

Clustering is an important data processing procedure in data mining. Recognition of natural clusters in data not only helps us get an insight into the structure of the data, but also provides a base for constructing data prediction (classification) model. In the past few decades, many well-known clustering techniques were developed, such as *k*-means [1,2], hierarchical clustering methods [3], KDE-based techniques [4], fuzzy clustering methods, mixture probabilistic model [5], graph-based clustering [6] and so on. Besides, there is a kind of natural-law-driven method that makes use of natural laws to do clustering analysis, or to change conventional ways of finding optimal or suboptimal solutions. This kind of method is often based on biological rules [7,8], or laws in physics such as gravitational model [9–15]. Natural-law-driven methods are inspired from people's understanding of the nature and symbolize an interesting and potential field in data clustering.

Methods based on gravitational model regard data point as mass point with different mass. Similarity of two points is measured by "data gravitational force". For calculating data gravitational force, universal gravitation formula (1) is often used.

$$F = G\frac{Mm}{R^2} \tag{1}$$

$$F_i = \sum_{j=1}^{n} \frac{1}{\|x_i - x_j\|^2} \tag{2}$$

where M and m are point masses, R is the distance between two points, and G is a constant. In calculating data gravitational force, G is often ignored. There are different ways of defining point mass. In many cases, the mass of point A is defined to be the number of data points that are located around A within a pre-assigned neighborhood. Different definition can be found in [12,14,15], that regards a point or a cluster as an object of unit mass. So, resultant force on a point can be simply written as (2), where n is the total number of data points. After the data gravitational force between a point and each cluster is obtained, the point can be assigned to the cluster that gives it the maximum gravitational force. Orhan et al. [14] introduce a gravitational fuzzy clustering approach that provides a solution of selecting initial centers according to refined density measure inspired by gravitation law. Endo and Iwata [15] introduce another gravitational model-based clustering approach that repeats the process of moving and merging points (clusters) according to gravitational force till the number of clusters reaches the target cluster number. In addition, if data gravitational force is defined as vector, angles between these force vectors can be used for supervised data classification [16]. The molecular dynamics-like clustering proposed in this paper has some difference with the approaches of Orhan et al. [14] and Endo and Iwata [15], though it also considers interaction forces between points. The main difference is the existence of repulsive force that is absent in gravitational model. Additionally, there is no pre-assigned target cluster number used in [14,15], so that natural clusters may possibly be found at some level determined by parameters. Some noisy points may be deleted by this molecular model, which is some functionally similar to clustering methods like in [17–19] such as trimmed *k*-means.

* Corresponding author.
E-mail addresses: babyron2000@sohu.com, myronsaga1@sohu.com (L. Junlin).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.01.008

Gravitational model for data clustering has "black hole" problem [13]. It is analogous to the black hole in cosmology. A black hole is a great massive object that produces extremely strong gravitational field around it. This field is so strong that gravitational forces between other low mass objects can be ignored. As clustering is concerned, if there exist some high density clusters in dataset, these clusters will produce relatively stronger data gravitational field. Therefore, data points tend to be attracted to high density clusters. If clusters in dataset have extremely great differences in density, high density clusters will raise black hole problem that may be bad enough to cause all points of low density clusters to be assigned to high density clusters. Besides black hole problem, some gravitational model-based method is quite sensitive to input sequence.

Gravitational model comes from observation of the macroscopic world. But when we explore into the microcosmic world, we may find that the dynamics mechanism of molecules can become a reference model of data clustering. Molecular kinetic theory says that attractive force and repulsive force both exist between molecules and both of the two forces decline as the distance between molecules increases [20–22]. Dotted lines in Fig. 1 indicate attraction or repulsion, and solid line indicates resultant force. According to molecular kinetic theory, there is a balance point r_0 (where resultant force$=$0) when the distance r between molecules is in the order of about 10^{-10} m. When the distance r is less than r_0, molecules will repulse each other because repulsive force is greater than attractive force. When the distance r exceeds r_0, molecules will attract each other because attractive force is greater than repulsive force. After resultant force peaks at some place where r is greater than r_0, resultant force will begin to decline. When the distance r is in the order of about 10^{-9} m, forces between molecules will be small enough to be ignored. According to Newton's law, attraction is often described as being inversely proportional to the square of the distance. Regarding repulsion, we define it to be inversely proportional to the 4th power of the distance in our molecular dynamics-like model. Then, we can have a simplified resultant

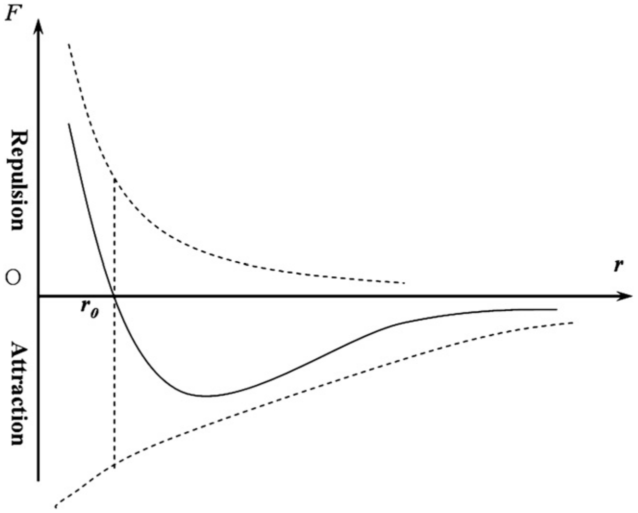

Fig. 1. Forces between molecules.

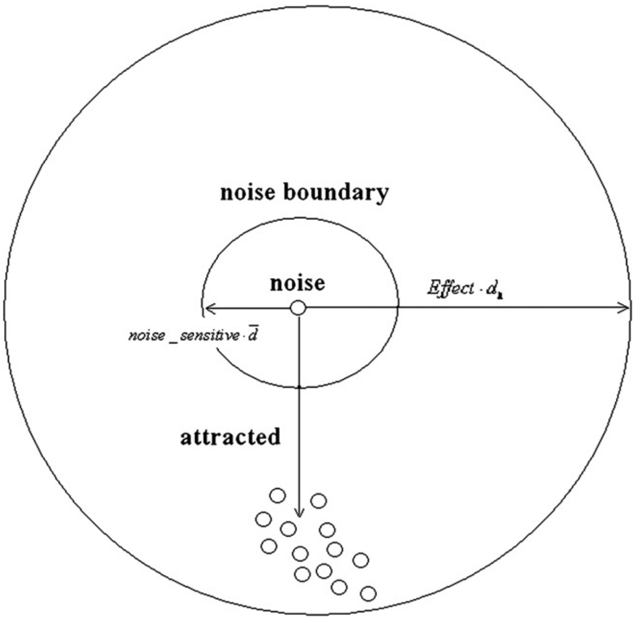

Fig. 2. Effects of parameter "noise_sensitive".

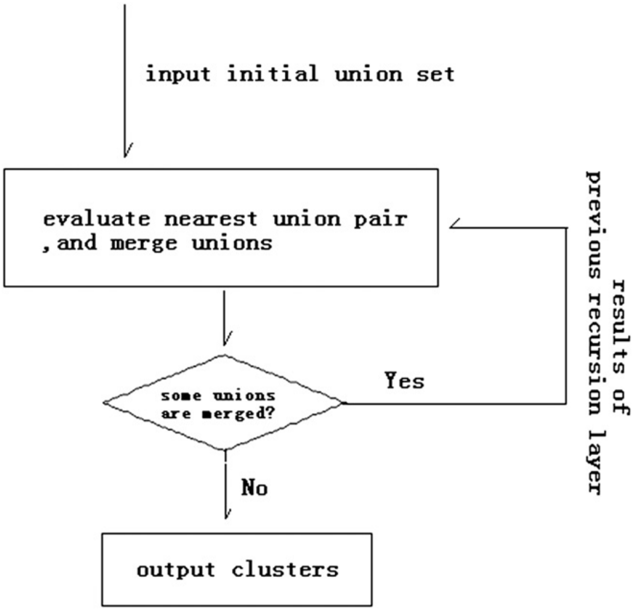
input initial union set
evaluate nearest union pair ,and merge unions
some unions are merged?
Yes
No
results of previous recursion layer
output clusters

Fig. 3. Recursive union-merging process.

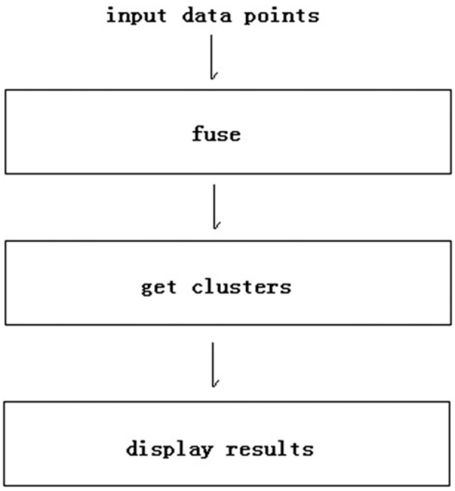

Fig. 4. The procedure of molecular dynamics-like data clustering approach.

force formula as

$$F = \frac{1}{r^2} - \frac{1}{r^4} \tag{3}$$

Eq. (3) defines the balance point to be at $r=1$, and peak at $r=\sqrt{2}$. This simplified formula conforms to the variation trend of resultant force shown in Fig. 1. In our model, resultant force formula is similar to (3) in form, but have some differences.

According to molecular kinetic theory, if two molecules are far enough away from each other, forces between them can be ignored. This is one of the intrinsic characters of molecular dynamics model. In accordance with this character, we need to introduce relevant parameter for force restriction. This intrinsic character of molecular dynamics model can avoid black hole problem that gravitational model must confront, because points of low density cluster will not be attracted by high density cluster no matter how much these clusters differ in density, as long as they are not in each other's force-effective area.

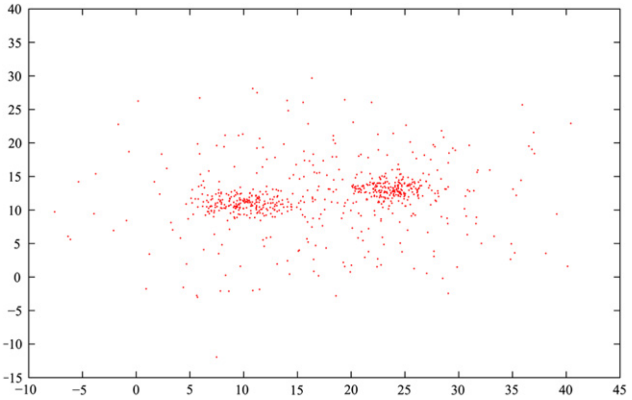

Fig. 5. Dataset having two clusters with strong Gaussian noise.

The rest of this paper is organized as follows. We introduce the molecular dynamics-like clustering approach in Section 2. Experiment results are shown in Section 3. Finally, we conclude the paper in Section 4.

2. Molecular dynamics-like clustering approach

2.1. Resultant force definition

Generally, clusters in dataset often have some differences in density besides being more or less apart from each other. Although there are also some density variations within a cluster, they are often less significant than between clusters. Obviously, it is more natural and more reasonable to divide a cluster into smaller ones, if several parts of the cluster have great density differences. For this matter, we need to consider both distance and density variation when clustering data.

One means of defining density is based on the distance to points in neighborhood [5,23]. But here we concern density variation, so we follow the idea of density definition in [4] to define the density variation in our approach as

$$\Delta d_{AB} = |d_A - d_B| \tag{4}$$

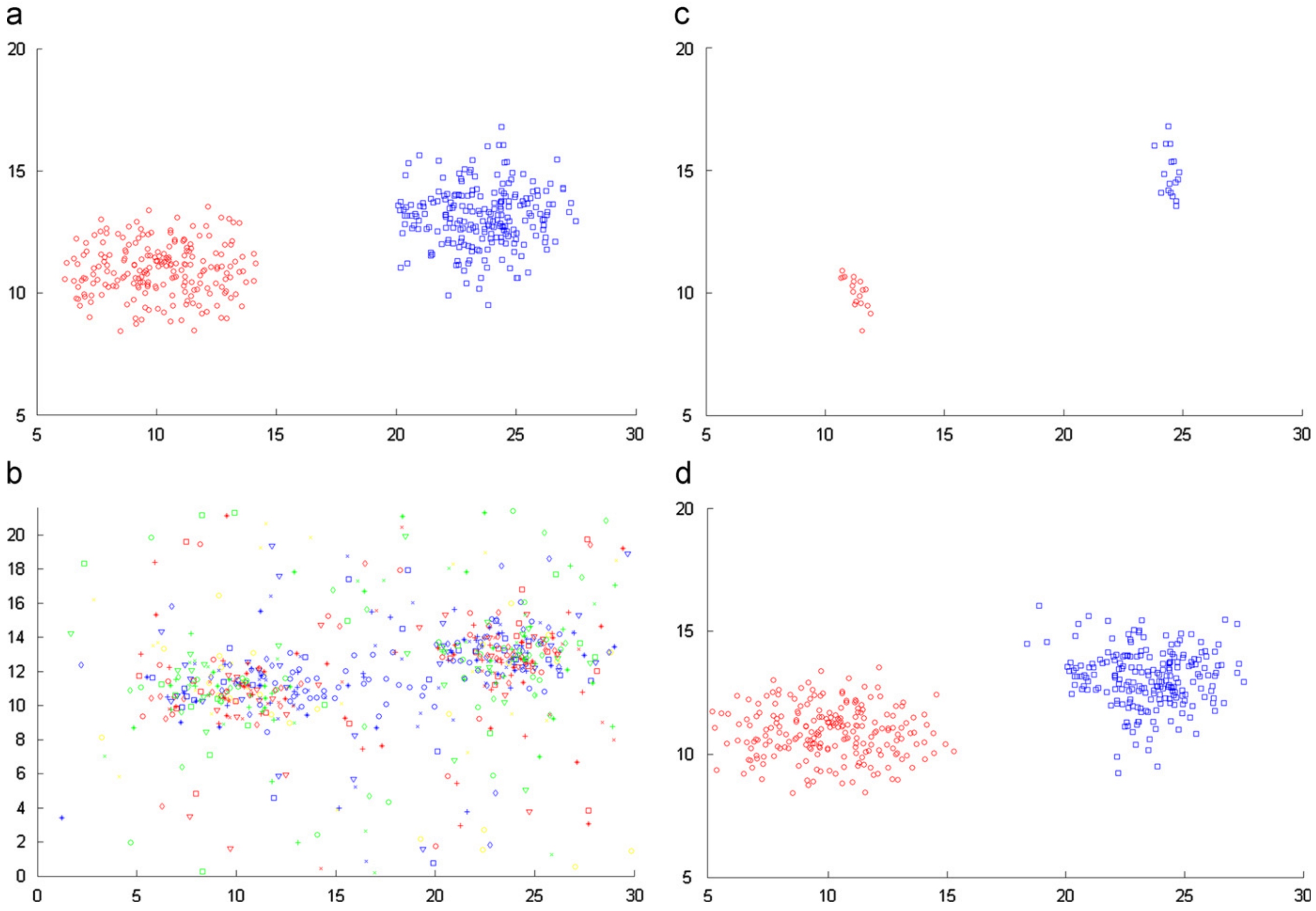

Fig. 6. Clustering results on the dataset with Gaussian noise.

where d_A (d_B) is the distance from point A (B) to its first nearest neighbor.

Data points fuse by molecular dynamics-like mechanism. For convenience, we name the space where data points fuse "iterating space", so as to distinguish from original feature space. Let S_o indicate the distance between point A and B in original space, and S_i the distance between point A and B in iterating space. Assuming that point mass is 1, the resultant force is defined as following:

$$F_{AB} = \frac{1}{(rd_A^{-1}S_o)^2} - \frac{\Delta d_{AB}}{(rd_A^{-1}S_i)^4} \quad (r > 1) \tag{5}$$

where Δd_{AB} is density variation between A and B in original space, d_A is the distance from point A to its first nearest neighbor. r is the parameter for controlling fusion degree, which is related to

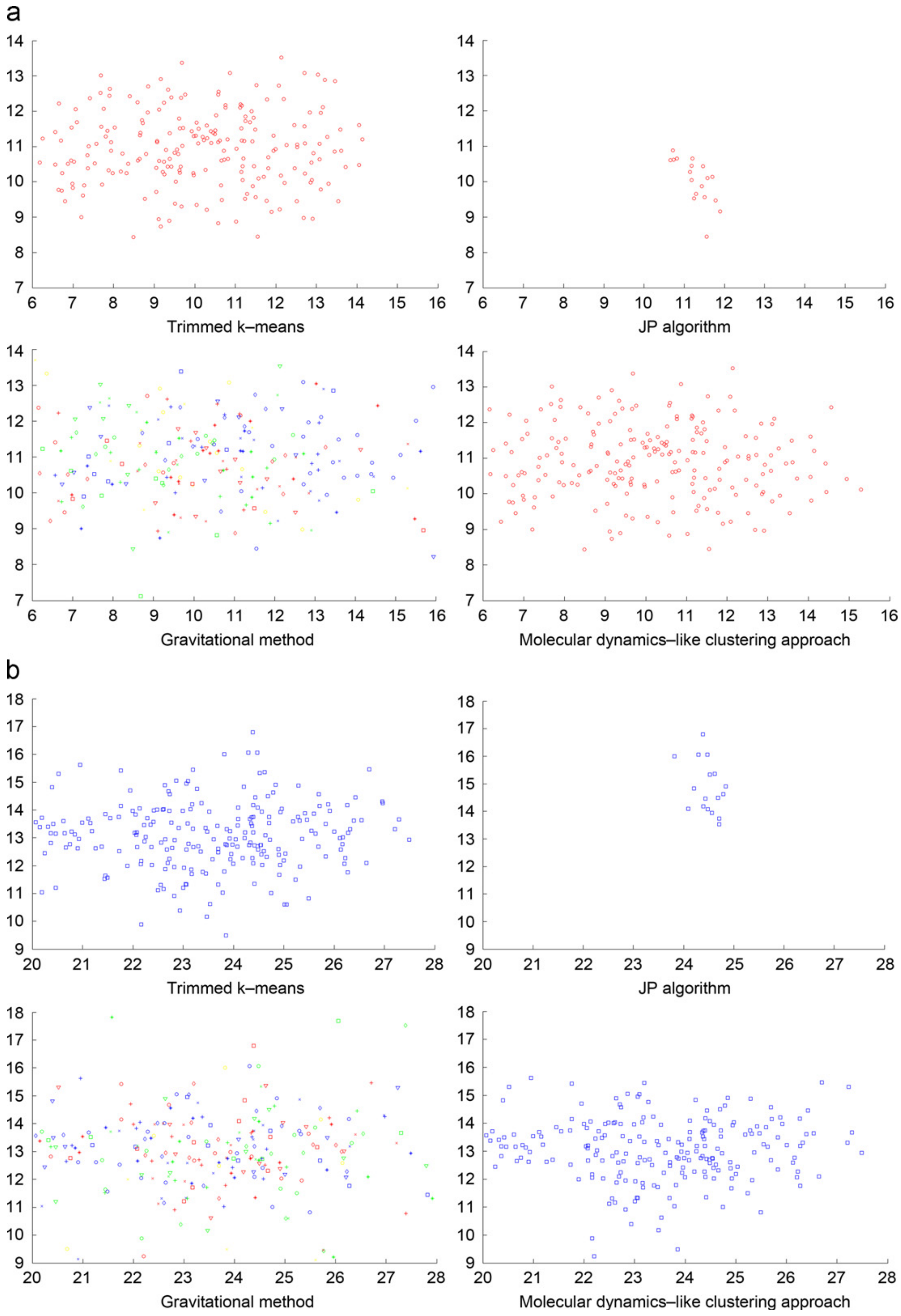

Fig. 7. Local views of the clustering results.

balance point. Fusion will become deep as r increases. When $F_{AB}>0$, A is attracted by B. When $F_{AB}<0$, A is repulsed by B.

Resultant force (5) is similar to (3) in form, but attractive force and repulsive force come from original space and iterating space, respectively. So, it is a little different from the molecular dynamics model in physics. The difference is that attractive force does not change when data points fuse in iterating space. It has two purposes: (1) put some emphasis on the distribution of the data in original space in order to avoid unreasonable strong attraction after fusion between points that may have weak attraction in original space and (2) speed up fusion to reach balance state. This is because r_0 tends to be smaller if attractive force and repulsive force both increase.

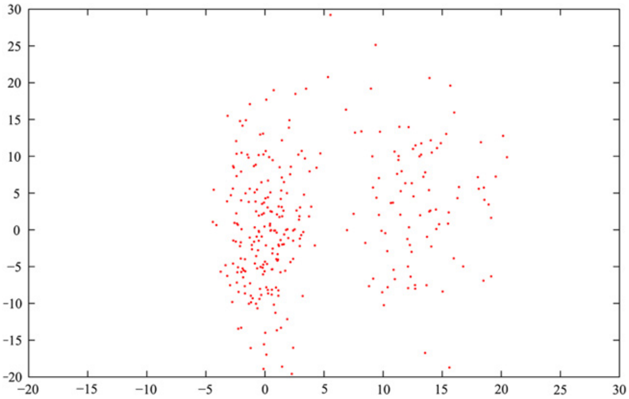

Fig. 8. Dataset with close clusters of similar density.

The procedure of this approach is described in PROCEDURE 1. Afterwards, some details about the steps are given.

```
PROCEDURE 1
S=the current dataset to be clustered
N=the total number of data points, namely $|S|$

1. Set $r$, $Effect$, $noise\_sensitive$
2. Calculate the distance from a point to its nearest neighbor $d_A$, $\forall A \in S$
3. Calculate $\overline{d}=(1/N)\sum_{A=1}^{N} d_A$
4. REPEAT
5. FOR each point ($A$) in S
6. {
7.    FOR each point ($B \neq A$) in $S$
8.    {
9.          Set $F_A$ to be 0 vector
10.         IF the distance from $B$ to $A$ in iterating space is not greater than $Effect \cdot d_A$ and $noise\_sensitive \cdot \overline{d}$,
              THEN calculate molecular force $F_{AB}$ by using Eqs. (4) and (5), $e_{\overrightarrow{AB}}=(\overrightarrow{AB}/|\overrightarrow{AB}|)$, and
              $F_A=F_A+F_{AB}\cdot e_{\overrightarrow{AB}}$.
11.      }
12.      Calculate $Ulti\_Direction_A=(F_A/|F_A|)$, according to Eq. (6).
13.      Calculate $\Delta A$ %This step does not move any point to its next position
14. }
```

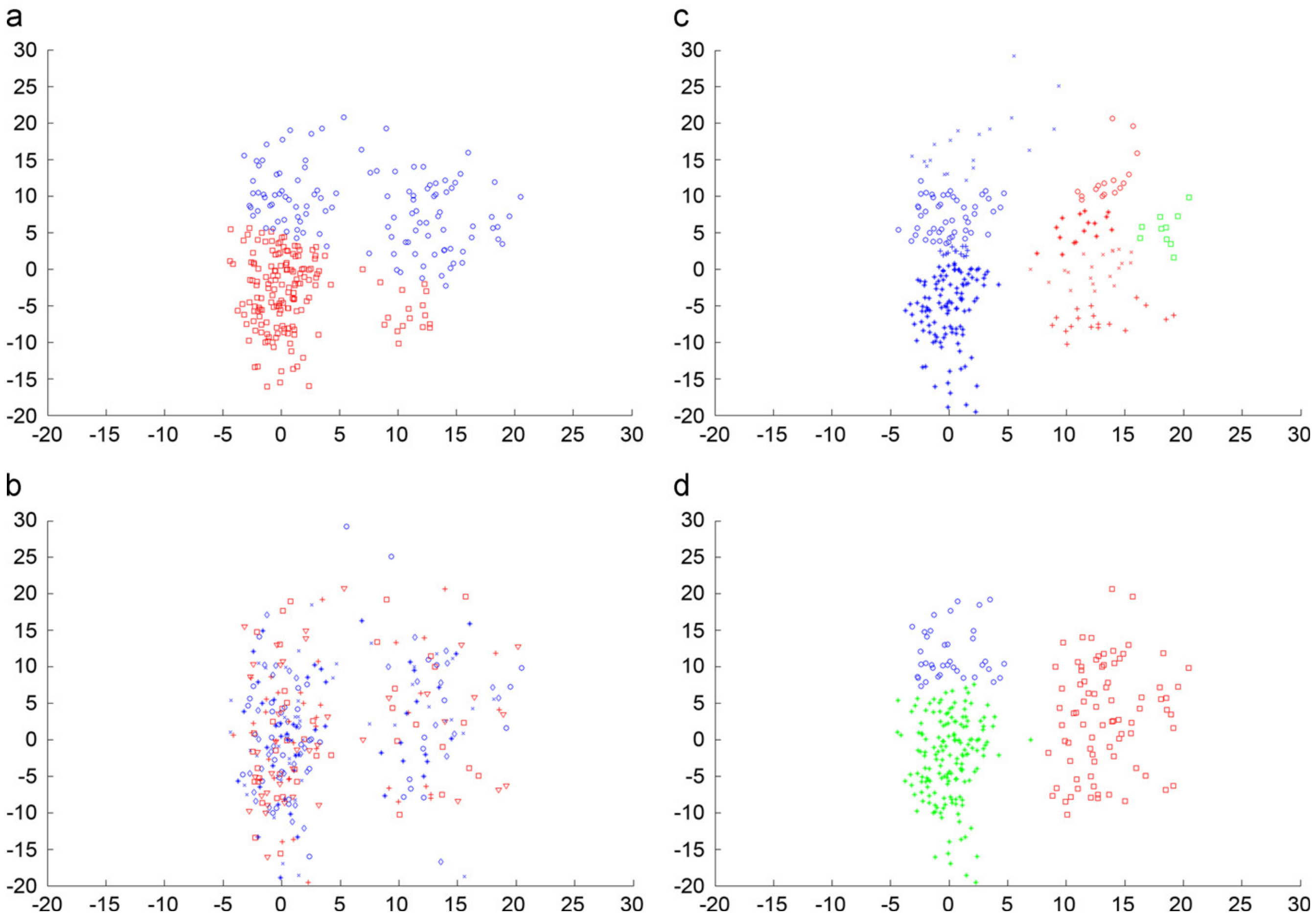

Fig. 9. Clustering results on close clusters with similar density. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

```
15. Move all points to their next positions by using Eq. (7) %This
    step moves all points
16. UNTIL some termination condition is satisfied
17. Group points into unions % Get Clusters step 1
18. REPEAT % Get Clusters step 2 (recursively merge unions into
    clusters)
19. For each nearest union pair, IF the two unions can be merged,
    THEN mark the pair
20. Merge marked unions according to their marks (IF P, Q can be
    merged and Q, R can be merged, THEN P, Q, R will be merged
    together into one union)
21. UNTIL no union pair can meet merging conditions
22. display results
23. End
```

2.2. *Move data point to its new position in iterating space*

According to molecular kinetic theory, forces between two molecules can be ignored, if they are far enough away from each other. Therefore, a parameter *Effect* for controlling the scope of molecular forces is introduced. $Effect \cdot d_A$ defines the scope of molecular forces. One character of molecular dynamics-like model is that the intrinsic effective-force restriction can deal with black hole problem. Points of low density cluster will not be attracted by high density cluster no matter how much these clusters differ in density, as long as they are not in each other's force-effective area defined by $Effect \cdot d_A$.

Although $Effect \cdot d_A$ defines the scope of molecular forces, some noise points can be included. Noise point tends to have big d_A, so that the product of *Effect* and d_A tends to be big. In this case, noise point tends to be attracted and move to data cluster as shown in Fig. 2. In order to deal with this case, another parameter *noise_sensitive* is introduced. $noise\_sensitive \cdot \overline{d}$ is related to sensitivity to noise, where $\overline{d} = (1/N)\sum_{A=1}^{N} d_A$ (N is the total number of data points, $\overline{d}$ is calculated in original space).

The initial state of iterating space is just the state of original space. Let N be the total number of points in dataset and η the step factor ($\eta \in (0,1]$). η can be set to decrease with time. After setting r, *Effect* and *noise_sensitive*with values, we calculate F_{Ai} according to (5) for point A. In calculating F_{Ai}, point i is such point that has its distance to A not greater than both $Effect \cdot d_A$ and $noise\_sensitive \cdot \overline{d}$. Then we calculate the direction $e_{\overrightarrow{Ai}}$ which is the unit vector from A to i in iterating space. The ultimate direction of A is got by (6) as following:

$$Ulti\_Direction_A = \Big(\sum_{i \in U} F_{Ai} e_{\overrightarrow{Ai}}\Big) \Big/ \Big|\Big(\sum_{i \in U} F_{Ai} e_{\overrightarrow{Ai}}\Big)\Big| \tag{6}$$

where U is A's neighborhood defined by $Effect \cdot d_A$ and $noise\_sensitive \cdot \overline{d}$ in iterating space.

Let $\Delta A = \eta * d_A * Ulti\_Direction_A$. The next position of A is got by

$$A^{next} = A + \Delta A \tag{7}$$

We can calculate the next position of each point in dataset, and move point to its new position in iterating space. Points in new positions in iterating space are the inputs for the next iteration. This process is iterated till termination condition is satisfied (e.g. the preset number of iterations is reached, or the magnitude of resultant vector of all data points changes less than a preset small value ε). In this iterating process, data points organize and fuse to balance state in which the distribution of the points in iterating space changes little.

This fusion mechanism is different from that of gravitational model. Supposing that point A is located in a high density cluster, point B, C in the same low density cluster, and distance between A and B is equal to the distance between B and C. This case may happen near the boundary between clusters. Considering gravitational model, because point A in high density cluster has greater mass than C in low density cluster, so A gives B stronger gravitational force than C gives. Then, B will be attracted by A and move in the direction of high density cluster. But it is different in this molecular dynamics-like model. Rewrite (5) as the following:

$$F_{BA} = \frac{1}{(RS_o)^2} - \frac{\Delta d_{BA}}{(RS_i)^4} \tag{8}$$

where $R = rd_B^{-1}$. When Δd_{BA} is big to a degree, F_{BA} can be weaker thanF_{BC}. A may even repulse B. In this case, B will be attracted by C and move towards low density cluster. This is more reasonable than gravitational model. For example, in the initial iteration of fusion process (the iterating process described above), iterating space is the same as original space. So, we get

$$S_{oBA} = S_{iBA}$$
$$S_{oBC} = S_{iBC}$$

Because $S_{oBA} = S_{oBC}$,so

$$\frac{1}{(RS_{oBA})^2} - \frac{1}{(RS_{iBA})^4} = \frac{1}{(RS_{oBC})^2} - \frac{1}{(RS_{iBC})^4}$$

Because A is in high density cluster and B, C both in the same low density cluster, so $\Delta d_{BA} > \Delta d_{BC}$. Then,

$$\frac{1}{(RS_{oBA})^2} - \frac{\Delta d_{BA}}{(RS_{iBA})^4} < \frac{1}{(RS_{oBC})^2} - \frac{\Delta d_{BC}}{(RS_{iBC})^4}$$

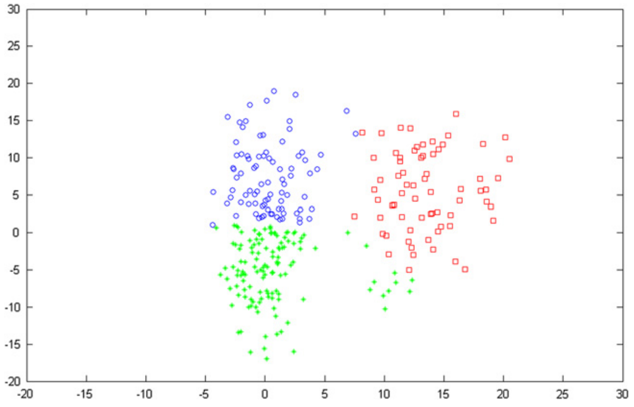

Fig. 10. Trimmed *k*-means clustering results with initializing 3 clusters.

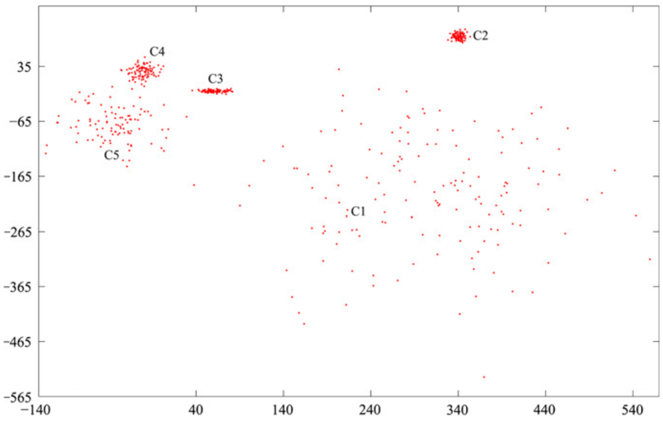

Fig. 11. Clusters differ greatly in density and size. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

This means $F_{BA} < F_{BC}$.Therefore in this molecular dynamics-like model, B is attracted by C and move in the direction of low density cluster that it belongs to. But in gravitational model, point A that has greater mass gives B dominant gravitational force and makes B move towards high density cluster, which is obviously unreasonable.

2.3. Get clusters

We get clusters through this stage (starting from step 17 of PROCEDURE 1), in which two steps are taken: (1) points are grouped into "unions" and (2) unions are merged into clusters. The following further describes the 2 steps.

Step 1 (Group points into unions): Point A and B are grouped into one union as long as the distance between them in iterating space is not greater than $\overline{d}$ that is calculated in original space. If A, B have been in a union and B, C have been in union, then A, B, C will be grouped into one union. If the distance between A and each of the other points is greater than $\overline{d}$, A will be recognized as noise point that does not belong to any cluster. If a point is recognized as noise, it can usually be deleted away from results. However, importance of noise depends on different goals (e.g. in intrusion detection, noise point often indicates an incoming attack), so that we can choose to either delete noise points or to assign them to certain clusters by some rules like *K-NN*, or to keep them in another special cluster for some other analyses.

Step 2 (merge unions into clusters): It is a recursive process to merge unions into clusters. In each layer of recursion, each union pair is evaluated to see whether they can be merged. Before going to the next layer of recursion, all pairs that meet merging conditions are merged all at one time. This recursive process terminates till no union pair can meet merging conditions. Then, clustering results are obtained and can be displayed in some manner.

We talk about the merging operation in one recursion layer in detail. The distance between two unions is defined as the distance between their centroids. The distance is calculated in iterating space. Merging can happen only between a union and its first nearest neighbor (union). Supposing that the first nearest neighbor of union P is Q in iterating space, then

1. In original space, calculate SSEs [5] of P, Q and the union produced by merging P and Q (indicated by SSE_P, SSE_Q and SSE_{PQ}).
2. If $(SSE_{PQ}/(|P|+|Q|)) \leq (SSE_P/|P|)$ or $(SSE_{PQ}/(|P|+|Q|)) \leq (SSE_Q/|Q|)$, then P and Q will be merged. If P, Q can be merged and Q, R can be merged, then P, Q, R will be merged together into one union.

The general recursive union-merging process is illustrated in Fig. 3. The general procedure of this clustering approach is summarized in Fig. 4.

According to PROCEDURE 1, considering operations like addition, distance computation and comparison, computational cost of 2, 3 is $O((2N^2-N)M)$, where M is the number of dimensions. 4–16 form iterative fusion process. In each iteration, distance between point A and B is calculated and molecular force F_{AB} is also calculated and added to resultant force F_A if related conditions are

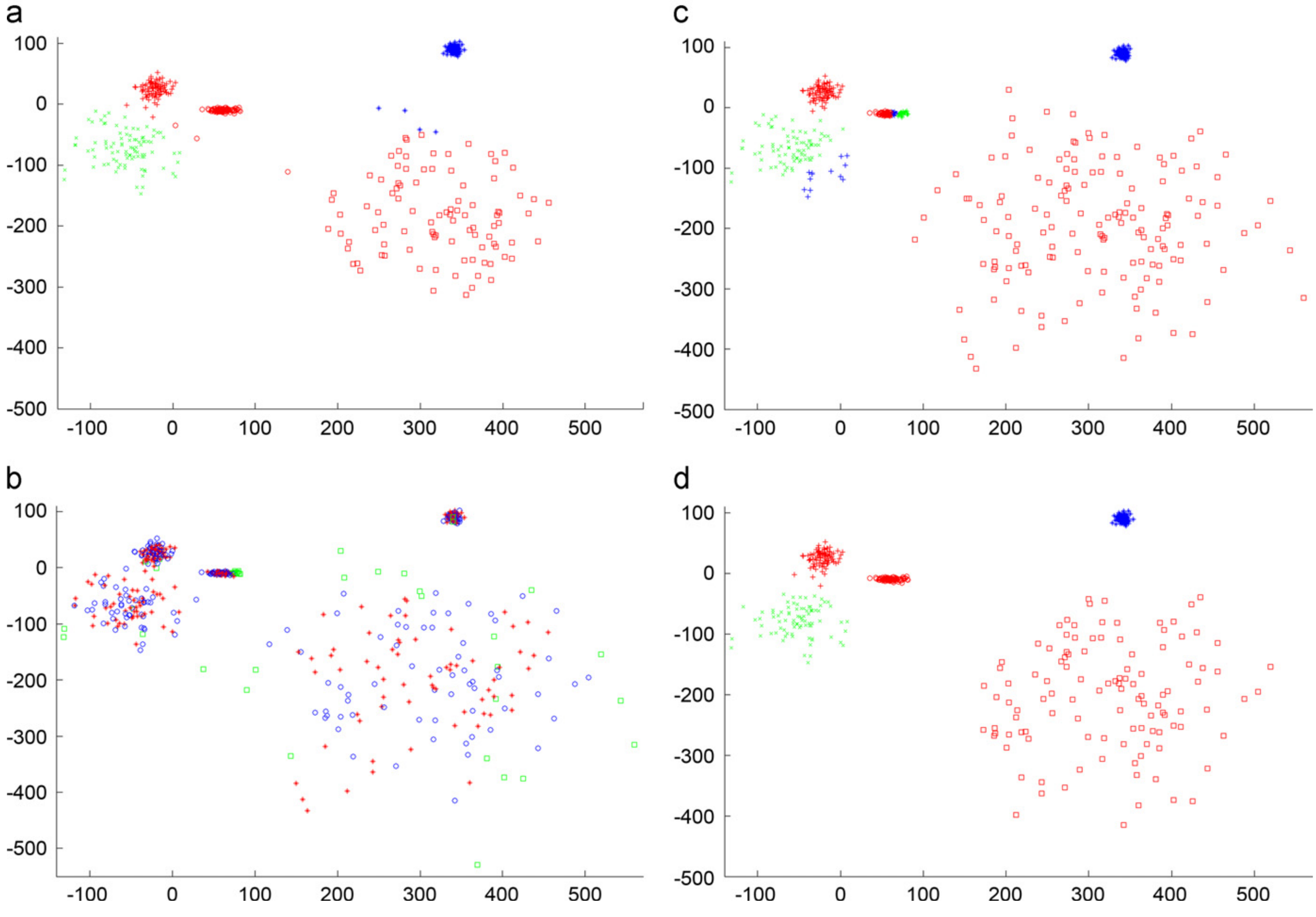

Fig. 12. Clustering results on dataset with obvious density differences.

satisfied. At the end of each iteration, all points are moved to their next positions (in 15). So, the main involved operations in one iteration are distance computation, comparison of values and addition operation. In the worst case, the main computational cost of 4–16 is $O((5N^2-2N)MT)$, where T is the number of iterations. 17 mainly involves distance computation and comparison, which is $O(2MN(N-1))$. 18–21 merge unions into clusters. This step involves computing centroids and distances computation. In the best case, this recursive process is skipped, which causes 0 computational costs. In the worst case, the main cost is

$$O\left(3MN\left(\left[\frac{N}{2}\right]-1\right)+2\sum_{i=0}^{i=[N/2]-2}\left(\left[\frac{N}{2}\right]-i\right)\left(\left[\frac{N}{2}\right]-i-1\right)\right),$$

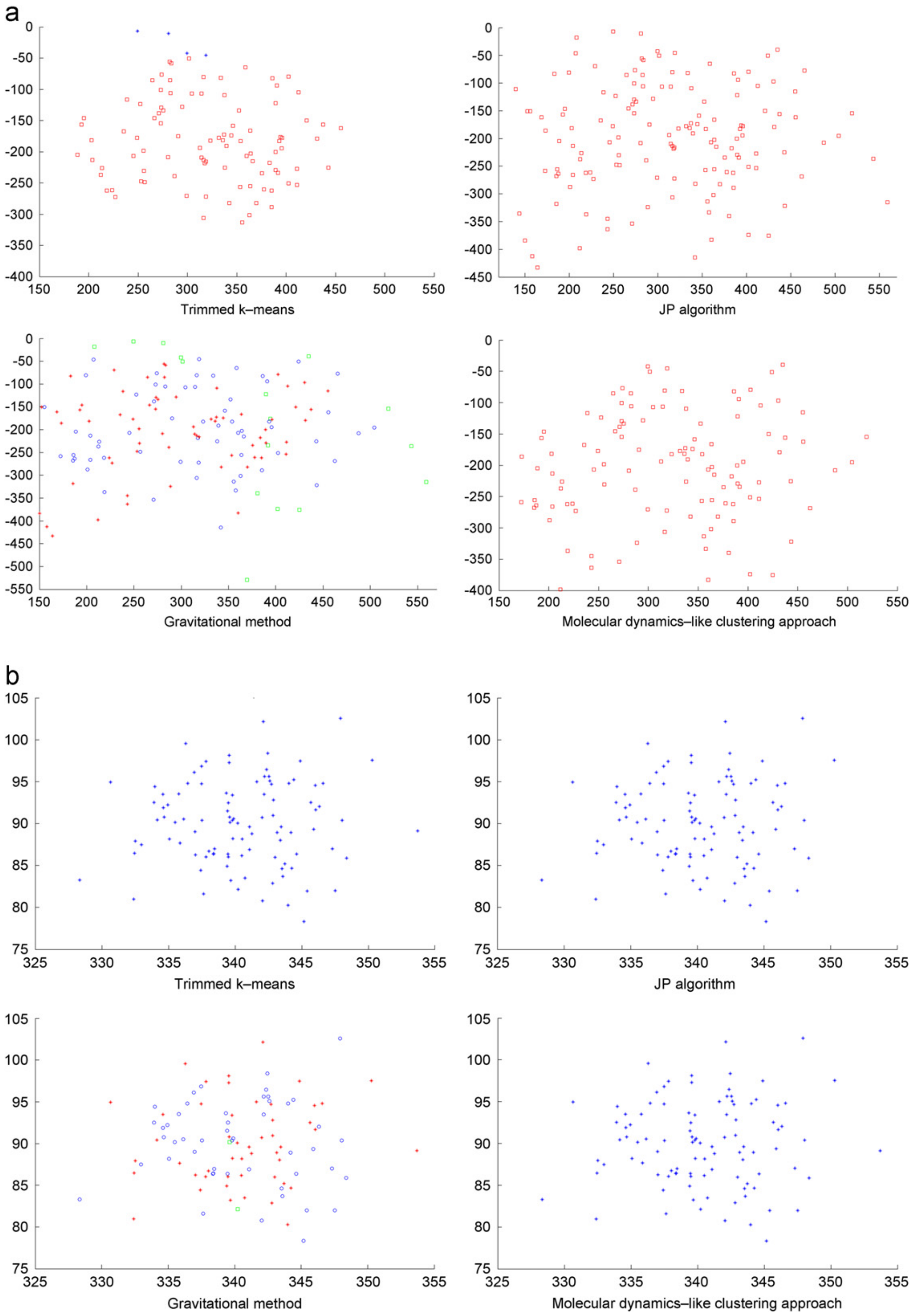

Fig. 13. Local views of the clustering results C1–C5. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

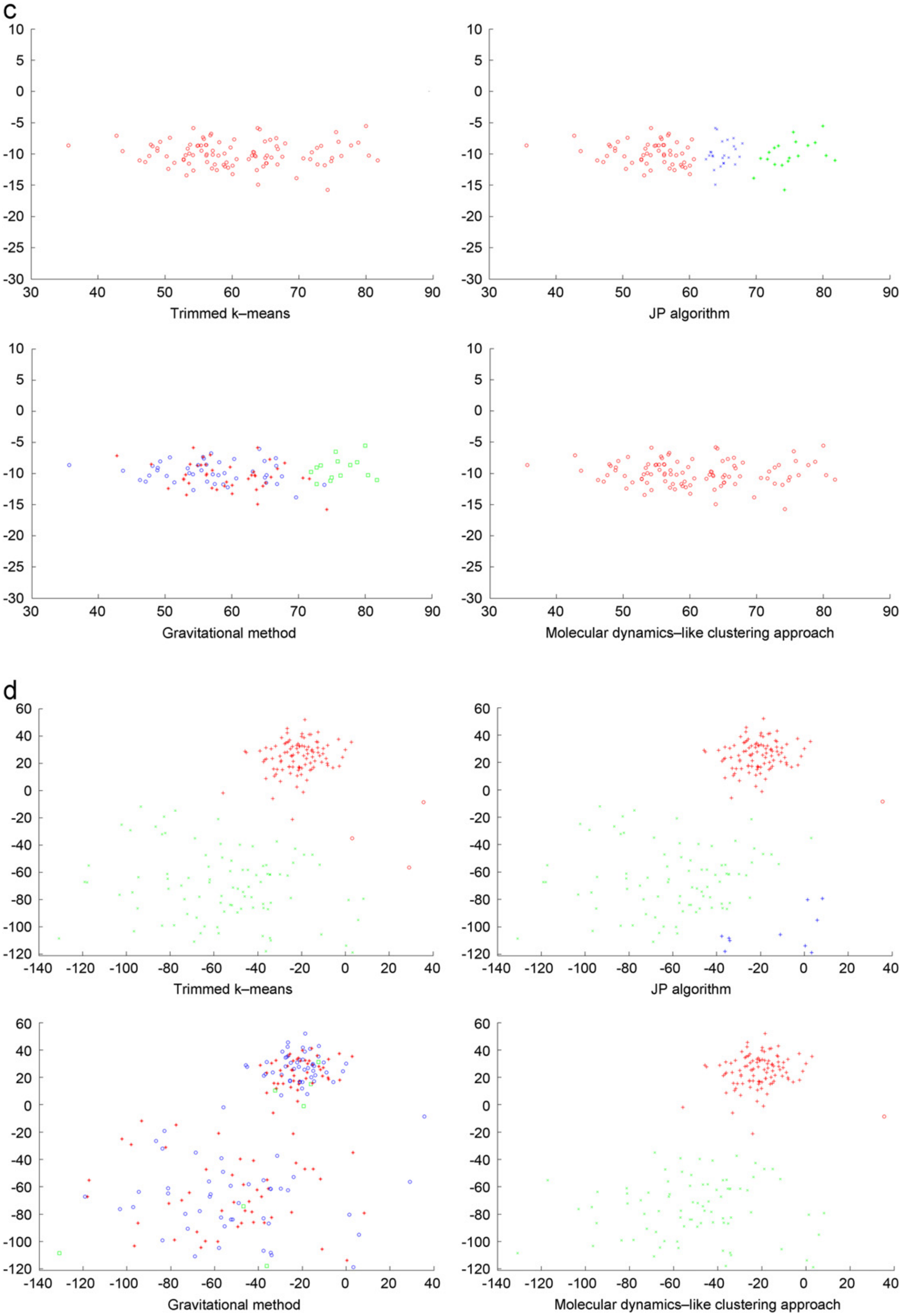

Fig. 13. (*Continued*)

where [x] indicates the biggest integer that is smaller than x. In general, the computation complexity is quadratic about the number of points in dataset and related to the number of dimensions and iterations.

There are different ways to display results. For example, clustering results can be displayed directly, or after eliminating noise, and so on.

3. Experiments

Artificial datasets and supervised real datasets were used in the experiments. The artificial datasets were generated with different distribution characteristics. The supervised datasets were Iris [24], Wine [25], and two gene expression dataset [26,27]. On artificial datasets, four clustering methods were applied: trimmed *k*-means [17,19], Jarvis–Patrick (JP) algorithm [5], a gravitational model based method [13], and the approach described in this paper. Trimmed *k*-means aims at robustifying generalized *k*-means through the use of a trimming procedure. JP algorithm based on SNN similarity concept is good at dealing with noise, and can find clusters of different densities and sizes. The method in [13] starts from one single cluster (one point), and then make use of data gravitational force to assign other points into the clusters that are successively formed later on. The same input sequence that was determined by the original positions of data points in dataset were used for all of the 4 methods. We adjusted parameters of these methods to obtain their best results in the experiments. The results of trimmed *k*-means were obtained after at least 100 runs with a given number of initial centers that were randomly chosen from the data points, and with the trimming level set to eliminate the same number of points as that of the molecular dynamics-like approach. Any cluster having $\leq 2.2\%$ of points of the dataset was deleted away as noises (rare clusters) in all these experiments. Results were displayed after erasing noises. The following discusses experiment results. The latter part of this section discusses the 3 important parameters: *r*, *Effect* and *noise_sensitive*.

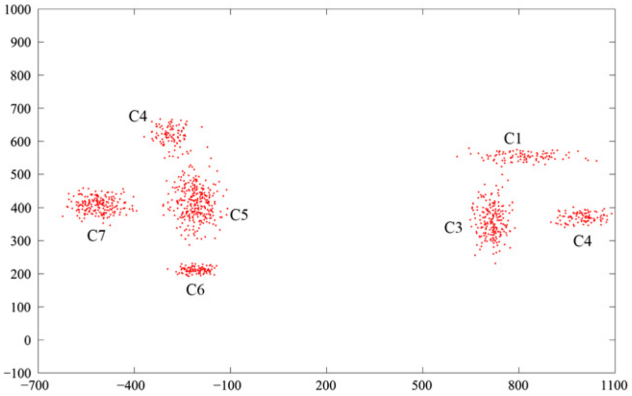

Fig. 14. Multi-clusters with some noises.

The artificial dataset in Fig. 5 has 700 points that contains two clusters with strong Gaussian background noise. We expected to extract the two clusters (C1, C2) from these noises. Fig. 6 shows the clustering results of the four methods. Points that were marked in the same color and shape were grouped into the same cluster. Gravitational model based method failed to remove noises from this dataset (Fig. 6b), because it is very sensitive to input sequence of data points. JP algorithm (length of nearest neighbor list (*K*) is 7, SNN threshold (KT) is 4), trimmed *k*-means ($k=2$, trimming level $a=0.38$), and molecular dynamics-like approach ($r=5$, effect$=15$, *noise_sensitive*$=2.9$) all displayed noise elimination ability, and found exact 2 clusters. However, JP algorithm removed so many points in dense areas that it only extracted very small pieces of the two clusters (Fig. 6c). Trimmed *k*-means and molecular dynamics-like clustering approach not only erased most noises, but also preserved shapes of the two clusters (Fig. 6a, d). The difference is that the cluster C1 extracted by trimmed *k*-means looks more near-spherical than that extracted by molecular dynamics-like approach. This can be seen clearly in Fig. 7.

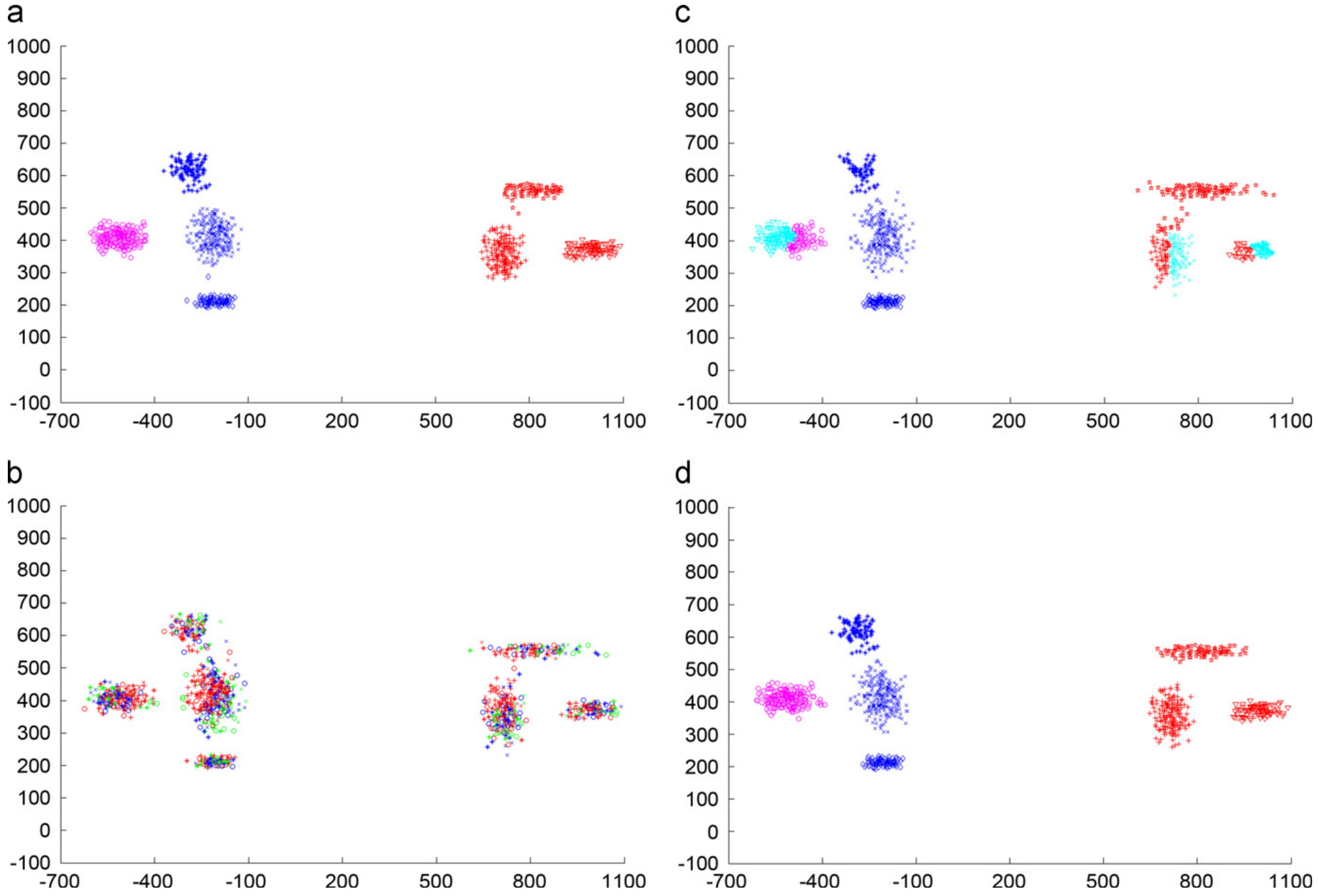

Fig. 15. Clustering results on multi-clusters. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Another dataset as shown in Fig. 8 has 300 points. It contains two clusters (the left and the right) that are close to each other and does not have great differences in density. We hoped to extract the main parts of the two clusters. Fig. 9a displays the results of trimmed *k*-means ($a=0.06$). The number of initial centers is 2. Trimmed *k*-means did not correctly find the two clusters. The two clusters that trimmed *k*-means found both contain many points that should obviously have been assigned to different clusters. For comparison, we initialized 3 clusters for trimmed *k*-means ($a=0.06$). Results were a kind of improved, but still over ten points were clustered wrongly as in Fig. 10. Jiang and Li's [13] gravitational model based method yielded the worst

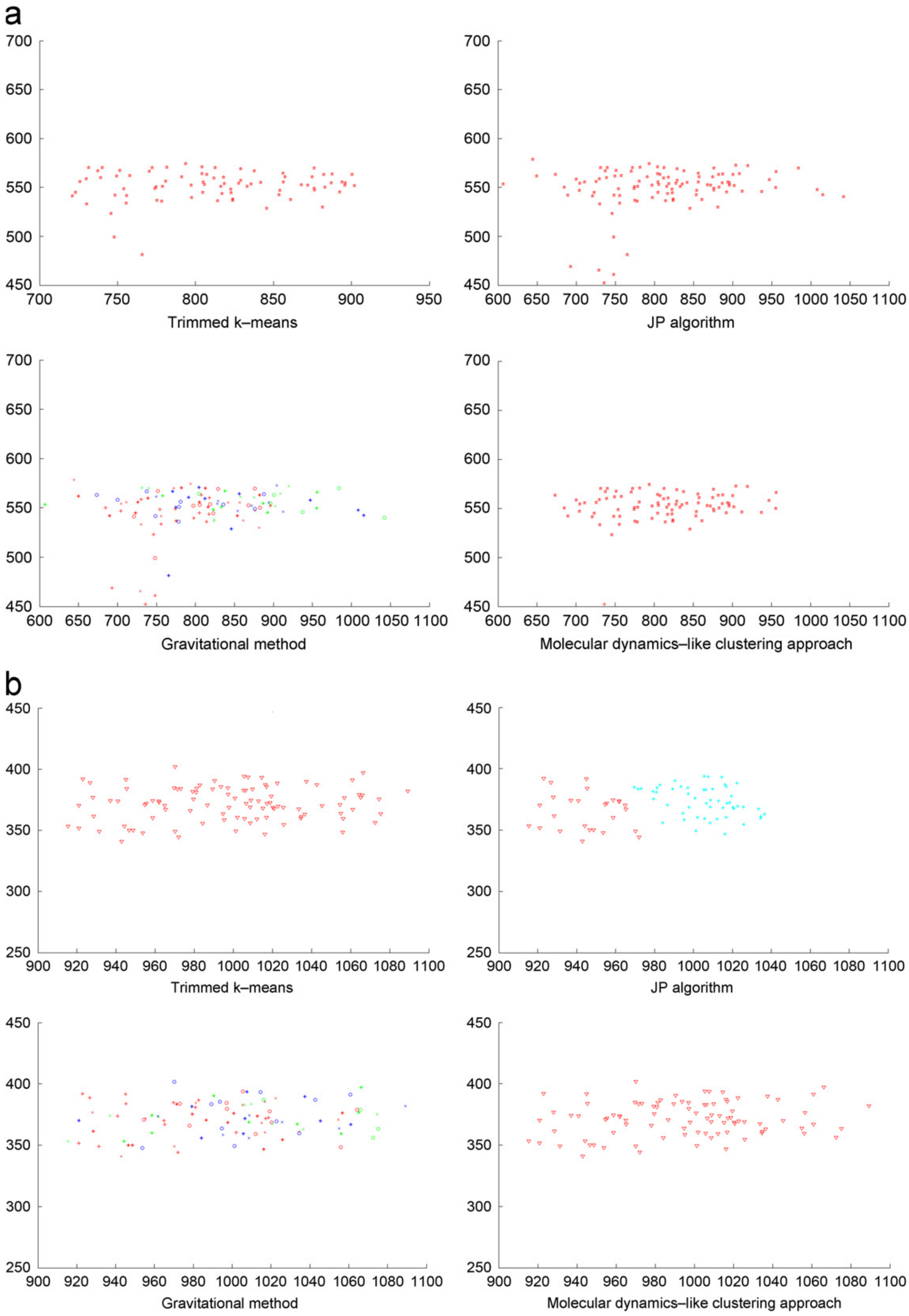

Fig. 16. Local views of the clustering results C1–C7.

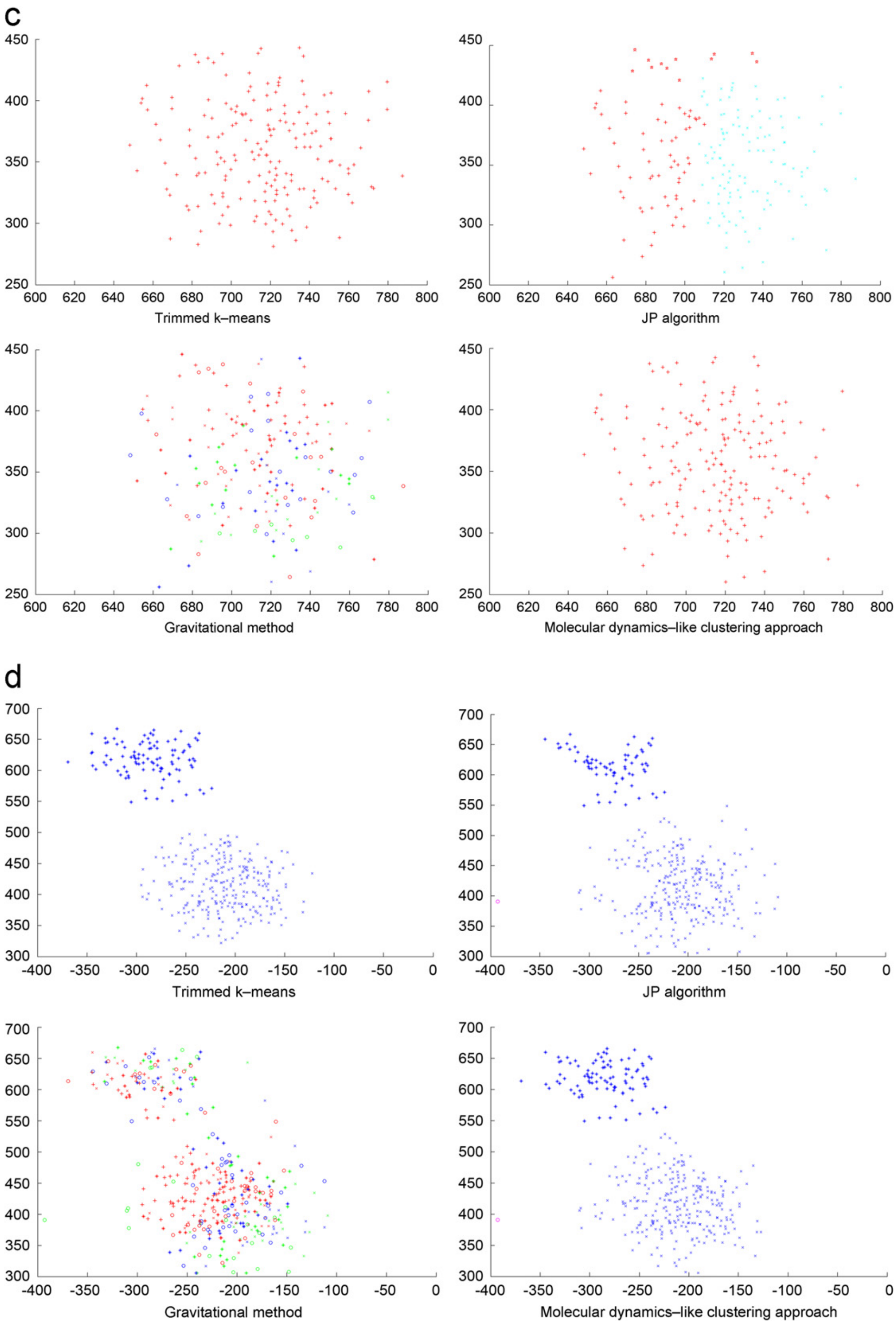

Fig. 16. (*Continued*)

results again. JP algorithm ($K=7$, $KT=3$) almost caught the general distribution of the two clusters, but it divided the two clusters into 9 smaller clusters, and the cluster marked with blue x seems not good (Fig. 9c). In comparison, molecular dynamics-like clustering approach ($r=8$, effect$=5.1$, *noise_sensitive*$=10.8$) caught the distribution characteristics of this dataset with less number of clusters (3 clusters).

Clusters may sometimes differ greatly in density and size. Dataset shown in Fig. 11 has 550 points forming 5 clusters (C1–C5). The maximum density ratio is more than 250:1. The

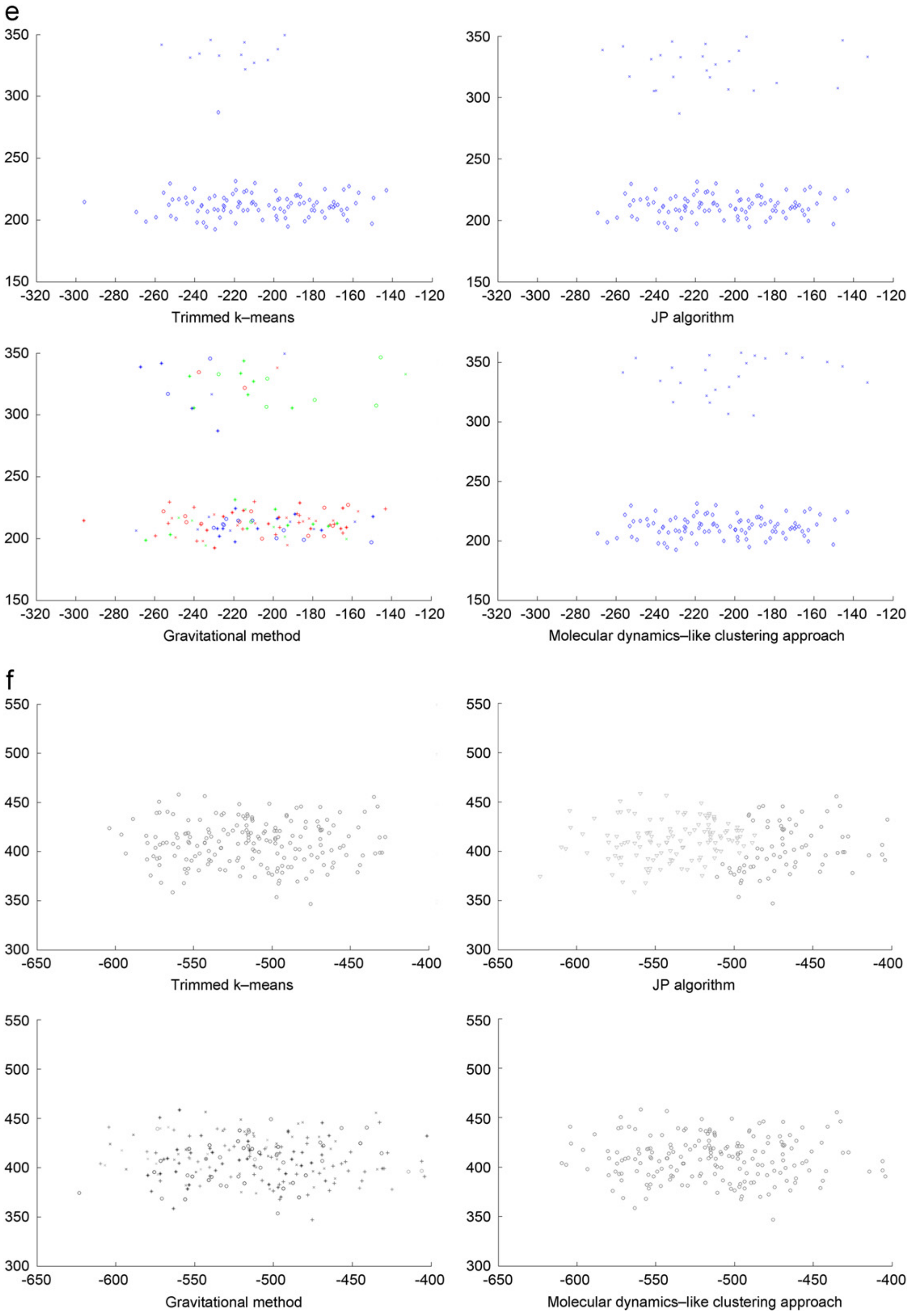

Fig. 16. (*Continued*)

number of initial centers for trimmed *k*-means is 5. The clusters (marked with blue "⁎") found by trimmed *k*-means ($a=0.091$) wrongly included several points that were located on the border of the cluster marked with red square (Fig. 12a). For the same reason of being sensitive to input sequence, the gravitational method did the worst job. Trimmed *k*-means, JP ($K=11$, $KT=5$) and molecular dynamics-like clustering approach ($r=100$, effect$=20$, *noise_sensitive*$=100$) caught the main distribution characteristics of the data. Molecular dynamics-like clustering approach found exactly the 5 clusters, and JP found totally 8 clusters. Especially, JP separated the oblate cluster into 3 clusters (marked with red circle, blue "x" and green "⁎"; see Fig. 13c). JP algorithm was more likely to separate clusters into smaller ones than molecular dynamics-like clustering approach in the experiments on datasets of Figs. 8 and 11.

Another dataset shown in Fig. 14 has 1100 points forming 7 clusters with some noises. Molecular dynamics-like clustering approach (r=15, effect=85, *noise_sensitive*=160) found 7 clusters as expected (Fig. 15d). Results of trimmed k-means (a=0.0719) were good except for some badly clustered points marked with red pentagram and blue diamond. The gravitational method still did worse than the other 3 methods. JP (K=10, KT=5) found 10 clusters. It divided 3 clusters (C2, C3, and C7) into 6 smaller ones. The cluster of red pentagram that was found by JP included some points that should have been assigned to other clusters, or been deleted as noise. In comparison, molecular dynamics-like clustering approach not only found the exact 7 clusters, but also erased some noise-like points between and on border of clusters better than other methods on this dataset (Fig. 15d). We can have a clearer view of these results in Fig. 16.

Iris and Wine are two famous real datasets from UCI Machine Learning Repository. Iris is a four dimension dataset giving length and width of sepal and petal of 3 types of iris. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Wine dataset are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. This dataset has 13 dimensions. The value of each dimension variable is quantities of one of 13 constituents. There are totally 178 instances in the dataset. The Yeast cell cycle dataset [26] consists of 384 genes whose expression levels peak at different time points corresponding to the five phases (five classes) of cell cycle. The original version of this dataset has 17 dimensions, but 10th and 11th dimensions that were unreliably measured [28] were excluded when clustering. The Yeast galactose dataset contains 205 instances that belong to 4 classes. It has 80 dimensions. We used the version after imputing missing values in [27]. The number of initial centers for trimmed k-means was set equal to the number of real classes, or equal to the number of clusters yielded by molecular dynamics-like approach (abbr. MDA). Cluster purity [5] was used to evaluate results. "N_cluster" in Table 1 refers to the number of clusters yielded by MDA. Other columns list cluster purity. On Iris, Wine and Yeast galactose, MDA and trimmed k-means both achieved rather high purity. Additionally, N_cluster is very close (Iris) or exactly equal (Wine, Yeast galactose) to the number of real classes. On all of the datasets except for yeast galactose, purity from MDA is higher than that from trimmed k-means. In real world, some classes may be overlapped greatly. In such case, we cannot often possibly find the exact number of real classes. But MDA helps to find those important core areas where relatively more instances of the same class are settled, because points relatively closer to each other and of being in similar density areas tend to be of the same class, and they tend to fuse into cluster under the mechanism of MDA. As to relatively more separable data like Iris, etc., MDA helps to approximate the number of real classes, because classes of this kind of data have less overlapped parts so that most points of the same class tend to fuse into less number of clusters.

Table 1
Clustering results on supervised datasets.

	MDA (%)	*N_cluster*	Trimmed k-means (%) (k=*N_clusters*)	Trimmed k-means (%) (k=the number of real classes)
Iris	91.67	4	88.89	88.89
Wine	98.27	3	94.80	94.8
Yeast cell cycle	63.52	15	62.23	49.36
Yeast galactose	97.00	4	99.00	99.00

Compared with JP algorithm, the gravitational method and trimmed k-means, the main extra computational cost of MDA lay in the iterative fusion process, but better results were obtained in return in these experiments. The average time cost ratio of MDA, JP, the gravitational method and trimmed k-means in these experiments is about 1:0.0032:0.0041:0.105

3.1. The important parameters: r, Effect and noise_sensitive

Molecular dynamics-like clustering approach has three important parameters: r, *Effect*and *noise_sensitive*. Parameter r is for controlling fusion degree. As r increases, fusion will become deep when reaching balance state. Fig. 17 shows that groups of points in iterating space tended to become more compact as r increased with other parameters unchanged. When r increases to a certain degree, groups of points in iterating space will have been isolated distinctly from each other on reaching balance state and will mainly tend to shrink into themselves. In this situation, relative positions of point groups change little so that further increases of

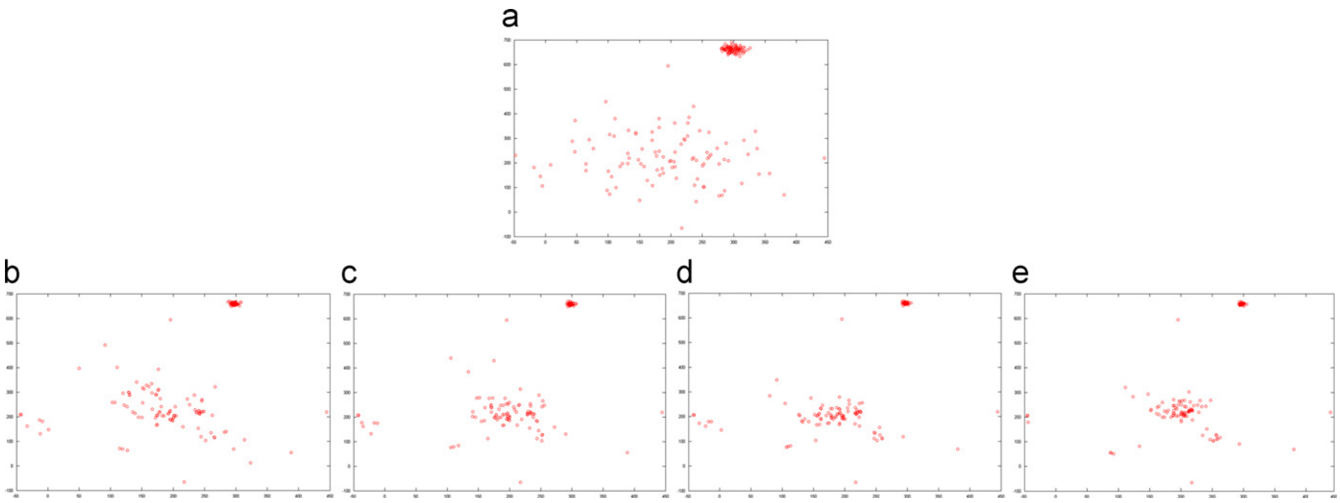

Fig. 17. Fusion degree is effected by r.

r have few influences on clustering results. It is unnecessary to set r with very big values.

Parameter *Effect* is for controlling the scope of molecular forces. Within the area where molecular forces are effective, data points interact with each other and fuse. When *Effect* increases, data points and point unions will tend to merger together into clusters of larger size. Taking the dataset of Fig. 8 for example, Fig. 18 displays clusters found with different values of *Effect* (0 indicates obtaining no cluster tag at present.). Other parameters were kept unchanged. In Fig. 18, a tendency is that, as *Effect* increases, clusters are gradually recognized from core (dense) areas, and points that are relatively close to each other tend to

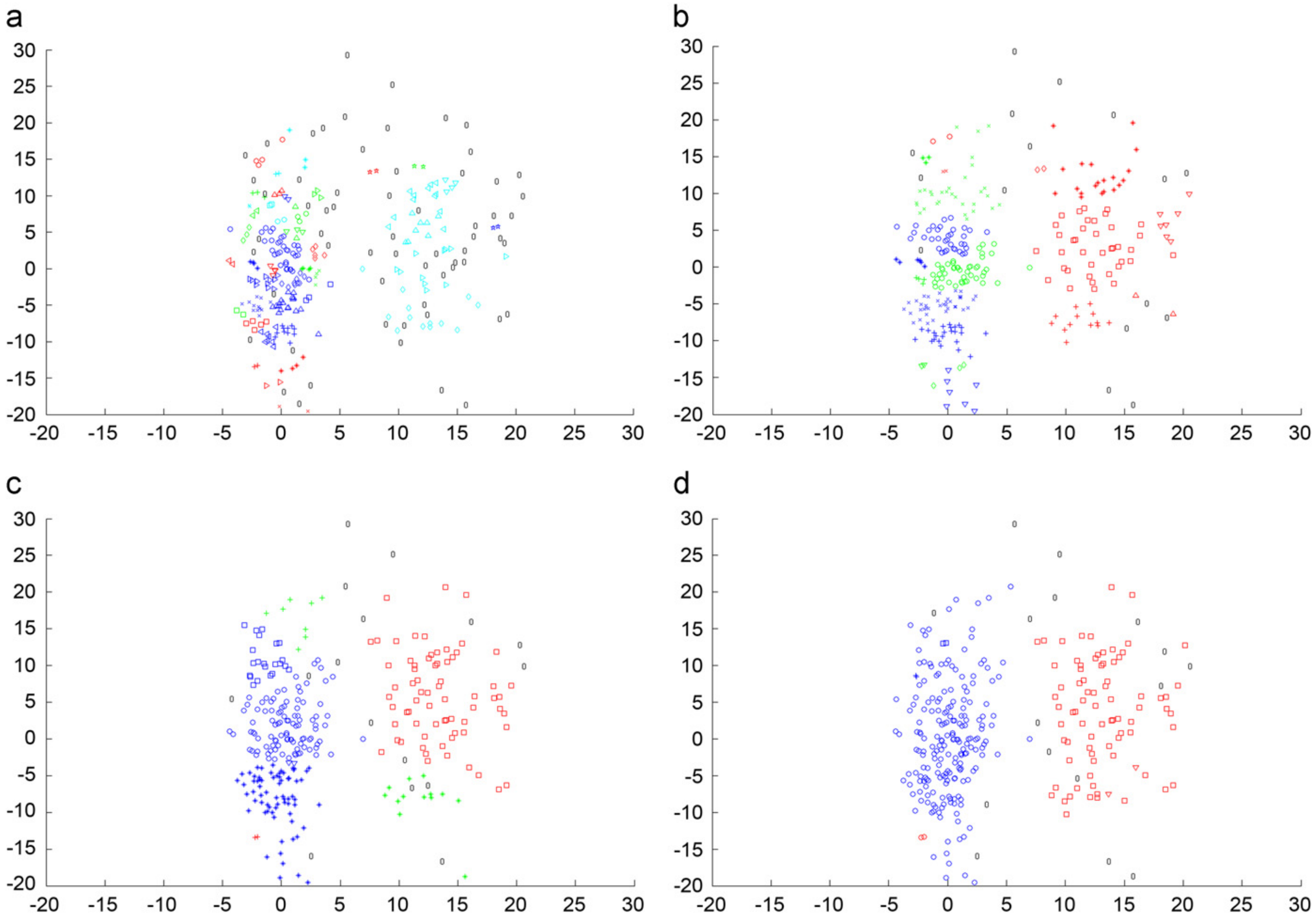

Fig. 18. Influences caused by Effect taking different values.

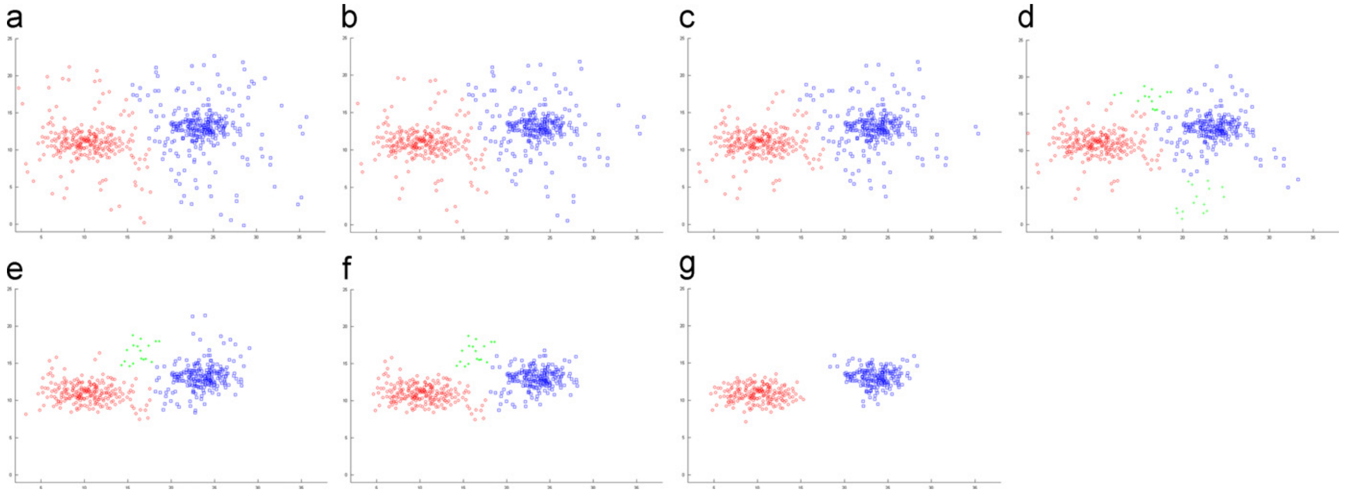

Fig. 19. Effects of parameter "noise_sensitive".

merge together to form bigger clusters and to reduce cluster number. We can increase *Effect* to improve clustering results if there are too many rare or nearly rare clusters.

noise_sensitive is related to sensitivity to noise. As this parameter increases, clusters tend to include more noises. So, for noise-polluted data, decreasing this parameter tends to improve clustering results. Taking the dataset of Fig. 5 for example, Fig. 19 displays the effects of *noise_sensitive*. As the value of *noise_sensitive* decreased, more and more noises around the two clusters were erased (Fig. 19a–g). In Fig. 19g, the two clusters almost show themselves.

Changes of these parameters more or less cause some changes of clustering results. In experiment, in order to get an idea of how much the change of parameter influences clustering results, we tested the method on datasets of Figs. 8, 11, Iris, Wine and Yeast cell cycle. Variance of silhouette coefficient [5], cluster purity or number of clusters was obtained in experiments that were conducted to cover both bad and good results on each dataset with the tested parameter increasing at a certain fixed step according to different datasets while keeping other parameters unchanged. The number of initial centers for trimmed *k*-means was set equal to the number of real classes. In Table 2, average variances of silhouette coefficient and cluster purity for parameter noise_sensitive are both bigger than those of parameter Effect or *r*. This showed that results were relatively more sensitive to noise_sensitive than to Effect or *r* on these five datasets. Additionally, parameter *r* caused a little more changes of results than Effect on these datasets. Compared with trimmed *k*-means, variances of results from noise_sensitive are generally bigger than those from trimming level *a*, while variances of results from Effect or *r* are less than or around those from trimming level *a* on these datasets. According to the variance of number of clusters in Table 2, noise_sensitive is still most influential while *r* caused smallest changes in number of clusters on these datasets. Generally, the sensitivity to noise_sensitive was greater than to Effect or *r* in these experiments.

If Effect is set to be infinite, all points will lie in each other's molecular force-effective area. In this case, the model's resistance against "black hole" problem is generally weakest and points tend to fuse towards the black hole cluster that is very compact and contains a large number of points. If Effect is set to be 1, point gives molecular force only to its nearest neighbor. In this case, there can possibly be many rare clusters and isolated points in clustering results, but points in core (dense) areas of a cluster tend to be recognized and clustered.

As has been discussed, few influences will be made on clustering results if *r* increases to infinite. When *r* is set to be very near zero, points always tend to repulse each other and points of the same cluster does not tend to fuse. So, $r > 1$ is defined for validity of the model.

If noise_sensitive is set infinite, noise points will tend to be attracted to move towards adjacent clusters. In this case, large number of noises (if there are many noises) tend to remain in clustering results. If noise_sensitive is set to be zero or Effect is set smaller than 1, fusion process will be skipped. This case also tends to produce many isolated points and rare clusters.

Generally, in order to get good clustering results, it is at least guaranteed that noise_sensitive $> 0, r > 1$ and Effect ≥ 1. When data contain clusters with large differences in density, relatively big value of Effect tend to yield good results. This is because in practical situations, some points can possibly be very close to their nearest neighbors though they belong to cluster of low density. This kind of points can often form mutual pair in which two points are each other's nearest neighbor. If this kind of points is expected to participate in interactions with more points in the same low density cluster, not limited to their nearest neighbors, it will require that the scope of force-effective area should be a certain number of times greater than the distance to the nearest neighbor. Recognition of low density cluster has this need. An empirical interval of *Effect* is$[1,4(\overline{d_o}/\overline{d})]$, where $\overline{d}=(1/N)\sum_{A=1}^{N} d_A$, $\overline{d_o}$ is the mean of the distances between points in a dataset (*N* is the total number of data points, $\overline{d}$ and $\overline{d_o}$ are calculated in original space). When clusters have large differences in density, tuning *Effect* to a relatively big value according to empirical interval can possibly lead to good clusters. Dataset of Fig. 11 is an example of such case.

When points settle in large areas in feature space and clusters are relatively far apart from each other, tuning noise_sensitive and *Effect* to relatively big value can extend the force-effective area of a point to possibly cover more points of the same clusters at low risks of involving points of other clusters that are far apart. The empirical interval of *noise_sensitive* is the same as that of

Table 2
Variance of silhouette coefficient, cluster purity or number of clusters.

	MDA						Trimmed *k*-means
	Variance of silhouette coefficient			Variance of number of clusters			Variance of silhouette coefficient
	Effect	*Noise_sensitive*	*r*	*Effect*	*Noise_sensitive*	*r*	Trimming level *a*
Fig. 8	0.0143	0.0350	0.0138	4.7400	5.7475	0.6600	0.0171
Fig. 11	0.0074	0.0036	0.0036	1.1556	0.7733	0.2600	0.0128
Iris	0.0041	0.0007	0.0004	10.1475	20.9900	0.1875	0.0094
Wine	0.0033	0.0182	0.0029	1.1100	8.8475	0.1875	0.0095
Yeast cell cycle	0.0006	0.0085	0.0031	1.1475	11.1250	3.5275	0.0022
	Average						
	0.0059	0.0132	0.0048	3.6601	9.4967	0.9645	0.0102
	Variance of purity						Variance of purity
	Effect	Noise_sensitive	*r*				Trimming level *a*
Iris	0.0003	0.0005	0.0034				0.0024
Wine	0.0065	0.0640	0.0158				0.0002
Yeast cell cycle	0.0006	0.0057	0.0021				0.0159
	Average						
	0.0025	0.0234	0.0071				0.0062

Effect. Results on dataset of Fig. 14 were from such tuning strategy according to empirical interval.

In the case like Fig. 5 that contained background noise, we first set *noise_sensitive* to a big value and then gradually decreased the value to eliminate noises. When clusters have small differences in density and are relatively close to each other, small values are first considered for *Effect* and *noise_sensitive*, and then increased according to clustering results.

An empirical interval of r is $(1{,}15+V_{nearest})$, where $V_{nearest}$ is the variance of nearest neighbor distance d_A. For dataset like Fig. 11 containing clusters with large density differences, in order to find low density clusters, we generally assigned r with a big value according to empirical interval. It aimed at raising fusion degree to make points of low density cluster shrink to a degree that they can be found. Contrarily for data containing clusters with small density differences, we preferred small values tuning in empirical interval in the experiments.

4. Conclusions

The clustering approach in this paper can find possible natural clusters in data by making use of the dynamical mechanism that is similar to the interaction between molecules through molecular forces. It does not require that the number of clusters be prespecified. The molecular dynamics-like clustering approach yielded better clustering results than trimmed k-means, JP algorithm, and the gravitational method of [13] on the datasets in experiments.

Acknowledgments

This work was partially supported by the China NKBRSF Project (Grant no. 2004CB318003), and the Knowledge Innovation Project of The Chinese Academy of Sciences (Grant no. KJCX2-YW-S02).

References

[1] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data. Prentice Hall Advanced Reference Series, Prentice Hall, 1988 March.

[2] S.M. Savaresi, D. Boley, A comparative analysis on the bisecting K-means and the PDDP clustering algorithms, Intelligent Data Analysis 8 (4) (2004) 345–362.

[3] P.H.A. Sneath, R.R. Sokal, Numerical Taxonomy, Freeman, San Francisco, 1971.

[4] A. Hinneburg, D.A..Keim, An efficient approach to clustering in large multimedia databases with noise, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. New York City, AAAI Press, 1998, pp. 58–65.

[5] Pang-Ning Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley Longman Publishing Co., Inc., 2005.

[6] I. Jonyer, D.J. Cook, L.B. Holder, Graph-based hierarchical conceptual clustering, Journal of Machine Learning Research 2 (2002) 19–43.

[7] U. Maulik, S. Bandyopadhyay, Genetic algorithm-based clustering technique, Pattern Recognition 33 (9) (2000) 1455–1465.

[8] Lei Wang, Huan Ji, An artificial immune cell model based C-means clustering algorithm, in: Proceedings of the Seventh World Congress on Intelligent Control and Automation, Chongqing, China, 2008, pp. 825–829.

[9] J. Gomez, D. Dasgupta, O. Nasraoui, A new gravitational clustering algorithm, in: Proceedings of the Third SIAM International Conference on Data Mining. San Francisco, CA, USA, 2003, pp. 83–94.

[10] W.E. Wright, Gravitational clustering, Pattern Recognition 9 (3) (1977) 151–166.

[11] T.V. Ravi, K.Chidananda Gowda, Clustering of symbolic objects using gravitational approach, IEEE Transactions on Systems, Man, and Cybernetics 29 (6) (1999) 888–894.

[12] U. Orhan, M. Hekim, Gravitational approach to supervised clustering for biclass datasets, in: Proceedings of Sixth International Conference on Electrical and Electronics Engineering, Bursa, Turkey,2009, pp. II-398–II-400.

[13] Sheng-yi Jiang, Qing-hua Li, Gravity-based clustering approach, Computer Applications 25 (2) (2005) 286–300.

[14] U. Orhan, M. Hekim, T. Ibrikci, Gravitational fuzzy clustering, MICAI 2008: advances in artificial intelligence, Lecture Notes in Computer Science, vol. 5317, Springer, 2008, pp. 524–531.

[15] Y. Endo, H. Iwata, Dynamic clustering based on universal gravitation model, Modeling Decisions for Artificial Intelligence, Lecture Notes in Computer Science 3558 (2005) 183–193.

[16] Li Junlin, Fu Hongguang, Data classification based on supporting data gravity, in: Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, China, 2009, pp. 22–28.

[17] L.A. Garcia-Escudero, A. Gordaliza, Robustness properties of k-means and trimmed k-means, Journal of the American Statistical Association 94 (447) (1999) 956–969.

[18] R.N. Dave, Characterization and detection of noise in clustering, Pattern Recognition Letters 12 (11) (1991) 657–664.

[19] J.A. Cuesta-albertos, A. Gordaliza, C. Matran, Trimmed k-means: an attempt to robustify quantizers, The Annals of Statistics 25 (2) (1997) 553–576.

[20] A.R. Leach.Molecular, Modelling: Principles and Applications, Prentice Hall, 2001.

[21] Kurt Binder, Monte Carlo and Molecular Dynamics Simulations in Polymer Science, Oxford University Press, 1995.

[22] Chen Zhenglong, Xu Weiren, Tang Lida, Theories and Practices of Molecular Simulation, Chemical Industry Press, China, 2007.

[23] Qing-Bao Liu, Su Deng, Chang-Hui Lu, Bo Wang, Yong-Feng Zhou, Relative density based k-nearest neighbors clustering algorithm, in: Proceedings of IEEE International Conference on Machine Learning and Cybernetics, vol. 1, 2003, pp. 133–137.

[24] Iris: UC Irvine Machine Learning Repository. On net: ⟨http://archive.ics.uci.edu/ml/datasets/Iris⟩.

[25] Wine: UC Irvine Machine Learning Repository. On net: ⟨http://archive.ics.uci.edu/ml/datasets/Wine⟩.

[26] K.Y. Yeung, C. Fraley, A. Murua, et al., Model-based clustering and data transformations for gene expression data, Bioinformatics 17 (10) (2001) 977–987.

[27] K.Y. Yeung, M. Medvedovic, R.E. Bumgarner, Clustering gene expression data with repeated measurements, Genome Biology 4 (5) (2003) R34.

[28] I.E. Tavazo, J.D. Hughes, M.J. Campbell, et al., Systematic determination of genetic network architecture, Natural Genetics 22 (3) (1999) 281–285.

Li Junlin received the master degree in software engineering from University of Electronic Science and Technology of China in 2007, and now is a doctoral student on computer software and theory. His research interests are in the fields of data mining, clustering and classification.

Fu Hongguang is a professor of School of Computer Science and Engineering, University of Electronic Science and Technology of China. His main research interests focus on the robotics,computer algebra, automated reasoning and knowledge engineering.