ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
DOI: 10.17559/TV-20150314115623

ADAPTIVE SEMI-SUPERVISED AFFINITY PROPAGATION CLUSTERING ALGORITHM BASED ON STRUCTURAL SIMILARITY

Limin Wang, Qiang Ji, Xuming Han

Original scientific paper
In view of the unsatisfying clustering effect of affinity propagation (AP) clustering algorithm when dealing with data sets of complex structures, an adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity (SAAP-SS) is proposed in this paper. First, a novel structural similarity is proposed by solving a non-linear, low-rank representation problem. Then we perform affinity propagation on the basis of adjusting the similarity matrix by utilizing the known pairwise constraints. Finally, the idea of fireworks explosion is introduced into the process of the algorithm. By adaptively searching the preference space bi-directionally, the algorithm's global and local searching abilities are balanced in order to find the optimal clustering structure. The results of the experiments with both synthetic and real data sets show performance improvements of the proposed algorithm compared with AP, FEO-SAP and K-means methods.

Keywords: *affinity propagation; fireworks explosion optimization; low rank representation; semi-supervised clustering; structural similarity*

Prilagodljivi polu-nadzirani algoritam grupiranja za srodno širenje utemeljen na strukturnoj sličnosti

Izvorni znanstveni članak
Uzimajući u obzir nezadovoljavajuće djelovanje grupiranja srodnog širenja algoritma grupiranja, kada se radi o nizovima podataka složenih struktura, u ovom se radu predlaže prilagodljivi nadzirani algoritam grupiranja srodnog širenja utemeljen na strukturnoj sličnosti (SAAP-SS). Najprije se predlaže nova strukturna sličnost rješavanjem nelinearnog problema zastupljenosti niskoga ranga. Zatim slijedi srodno širenje na temelju podešavanja matrice sličnosti primjenom poznatih udvojenih ograničenja. Na kraju se u postupak algoritma uvodi ideja eksplozija kod vatrometa. Prilagodljivo pretražujući preferencijalni prostor u dva smjera, uravnotežuju se globalne i lokalne pretraživačke sposobnosti algoritma u cilju pronalaženja optimalne strukture grupiranja. Rezultati eksperimenata i sa sintetičkim i s realnim nizovima podataka pokazuju poboljšanja u radu predloženog algoritma u usporedbi s AP, FEO-SAP i K-means metodama.

Ključne riječi: *optimizacija eksplozija vatrometa; polu-nadzirano grupiranje; srodno širenje; strukturna sličnost; zastupljenost niskoga ranga*

1 Introduction

Affinity Propagation (AP), proposed by Frey and Dueck, is a fast and efficient clustering algorithm. This distinctive clustering algorithm does not require the number of clusters to be predetermined like other clustering algorithms do, instead it considers all data points as potential exemplars and finds the optimal ones through continuous iteration [1]. Therefore, it is widely used in gene sequence analysis [2], text clustering [3], image processing [4, 5], facility location [6] and many other fields [7÷9]. So far, a great many scholars have carried out in-depth studies of AP and put forward improved versions of it. For instance, Japanese scholars Fujiwara Y et al. eliminated the unnecessary information exchange in iteration and proposed an AP clustering algorithm that has much better convergence rate without compromising the accuracy of the clustering result [10]. American scholars Givoni I. et al. extended AP in a principled way to solve the hierarchical clustering problem and proposed Hierarchical Affinity Propagation, which was successfully applied to actual HIV genetic sequence data [11]. By introducing the idea of manifold learning to AP, Feng Xiaolei et al. proposed a manifold distance-based semi-supervised AP clustering algorithm, which more accurately reflects the potential structure of actual data and has better clustering performance [12]. Wang Xianhui et al. combined AP with K-means clustering and put forward an AP-based cluster ensemble algorithm that effectively improves the accuracy, robustness and stability of K-means clustering [13].

The notion of similarity between observations is at the root of affinity propagation clustering. However, notably in many cases, for datasets with complex structures, Euclidean distance is not sufficient to represent the underlying structures. The original AP performs well for datasets with simple structures, but not for complex ones. This is because during the attempt to minimize the decision function, the original AP algorithm tends to produce excessive local clusters. Focusing on such problems, we propose a novel adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity (SAAP-SS) in this paper. In detail, this study consists of three critical steps: (1) a novel structured-based similarity is proposed to account for a low-dimensional global structure of the data. We solve a regularized low-rank representation problem of the observed data. Then we present the methods of constructing kernels for the design of a structured kernel similarity based on the low-rank representation. (2) In order to better reflect the similarity between data points, we use the known labelled data or pairwise constraints to adjust the similarity matrix. (3) During the running process of AP, a novel fireworks explosion optimization algorithm is adopted to select the preference parameter. In the early stage of the algorithm, the positions of fireworks and sparks are evaluated with the Silhouette index to search for the optimal preference space and promote the global searching ability of the algorithm. Afterward, according to the clustering structure, the radii of the fireworks explosion are adjusted adaptively to enhance the local searching ability of the algorithm and to determine the optimal clustering structure.

2 Affinity propagation clustering algorithm

Affinity propagation is a novel and high-efficiency algorithm that takes as input measures of similarity between pairs of data points and simultaneously considers all data points as potential exemplars. Real-valued messages are exchanged between data points until a high quality set of exemplars and corresponding clusters gradually emerges. Because of its simplicity, general applicability, and performance, we believe affinity propagation will prove to be of broad value in science and engineering [20]. Fig. 1 shows affinity propagation among a small set of two dimensional data points. Input consists of a collection of real-valued similarities between data points.

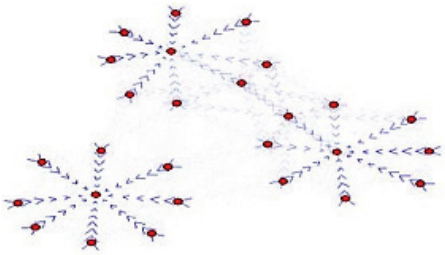

Figure 1 Affinity propagation

Affinity propagation takes as input a collection of real-valued similarities between data points, where the similarity $s(i, k)$ indicates how well the data point with index k is suited to be the exemplar for data point i. Each similarity is set to a negative Euclidean distance, as shown in Eq. (1):

$$s(i,k) = -d_{ik} = -\left\| x_i - x_k \right\|. \tag{1}$$

A real number $s(k, k)$ is taken as input for each data point k so that data points with larger values of $s(k, k)$ are more likely to be chosen as exemplars. These values are referred to as preference parameter; they play important roles in determining the number of exemplars. Initially all data points are equally suitable as exemplars, the preference parameter should be set to a common value p — this value can be varied to produce different numbers of clusters. In most cases, this shared value could be the median of the input similarities.

$$p = median(s(:)). \tag{2}$$

During the iteration, there are two types of messages exchanged between data points, and each takes into account a different kind of competition. Messages can be combined at any stage to decide which points are exemplars and, for every other point, which exemplar it belongs to. Fig. 2 shows affinity propagation is illustrated for two-dimensional data points, where negative Euclidean distance was used to measure similarity. Each point is coloured according to the current evidence that it is a cluster centre (exemplar). The darkness of the arrow directed from point i to point k corresponds to the strength of the transmitted message that point i belongs to exemplar point k [1].

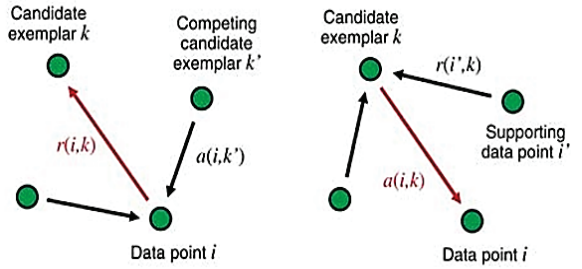

Figure 2 How affinity propagation works

"Responsibility" message $r(i, k)$, sent from data point i to candidate exemplar k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point i. "Availability" message $a(i, k)$, sent from candidate exemplar k to data point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar. The message-passing procedure is simulated by the updates of the two message matrices. A decision matrix E is calculated after each update. Decision matrix E represents whether point i chooses point k as its exemplar or not.

$$r(i,k) \leftarrow s(i,k) - \max_{k' s.t. k' \neq k} \left\{ a(i,k') + s(i,k') \right\}, \tag{3}$$

$$a(i,k) \leftarrow \begin{cases} \min\left\{0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} \max\left\{0, r(i',k)\right\}\right\} & i \neq k \\ \sum_{i' s.t. i' \neq k} \max\left\{0, r(i',k)\right\} & i = k \end{cases}, \tag{4}$$

$$E(k) = \arg\max_k \left(a(i,k) + r(i,k) \right). \tag{5}$$

When updating the messages, it is important that they are damped to avoid numerical oscillations that arise in some circumstances. Each message is set to λ times its value from the previous iteration plus $1 - \lambda$ times its prescribed updated value, where the range of the damping factor λ is between 0 and 1. A default damping factor of 0,5 was adopted, and each iteration of affinity propagation consisted of (i) updating all responsibilities given the availabilities, (ii) updating all availabilities given the responsibilities, and (iii) combining availabilities and responsibilities to monitor the exemplar decisions and terminate the algorithm when these decisions did not change for 10 iterations.

$$r^{(t+1)}(i,k) \leftarrow (1-\lambda) \cdot r^{(t+1)}(i,k) + \lambda \cdot r^{(t)}(i,k), \tag{6}$$

$$a^{(t+1)}(i,k) \leftarrow (1-\lambda) \cdot a^{(t+1)}(i,k) + \lambda \cdot a^{(t)}(i,k). \tag{7}$$

3 Adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity

3.1 Structure-based similarity measure

3.1.1 Low-rank data transformation

Consider a set of observations, $X = [x_1, x_2 \ldots, x_n]$, where $x_i \in \gamma^{d \times 1}$, approximately embedded on multiple

independent, low-dimensional manifolds, our goal is to discover these manifolds by using some techniques to learn low-rank representations of the data. In the cases where the observations are embedded on linear subspaces, the low-rank representation (LRR) problem can be formulated as:

$$Z = \min_Z \|X - XZ\|_F^2 . \quad s.t. \ \ \text{rank}(Z) = R \tag{8}$$

Where $\|\cdot\|_F$ is the Frobenius norm, and the solution, Z, is the minimum squared-error linear embedding on a R-dimensional subspace. Relaxing the constraint in Eq. (9), the minimization can be equivalently written as:

$$\min_Z \|X - XZ\|_F^2 + \lambda \cdot \text{rank}(Z), \tag{9}$$

The optimization of the rank of a matrix is non-convex and combinatorial. However, the convex relaxation of rank, the nuclear norm, can be substituted, resulting in the convex optimization:

$$\min_Z \|X - XZ\|_F^2 + \lambda \cdot \|Z\|_* . \tag{10}$$

This related problem was originally posed as a subspace segmentation approach by Liu et al. [14], who minimized the ℓ_2/ℓ_1 embedding error. Naturally, a kernel low-rank representation (KLRR) formulation of the problem is proposed for the cases where data is embedded on nonlinear subspaces:

$$\min_Z \frac{1}{2}\|\phi(X) - \phi(X)Z\|_F^2 + \lambda \cdot \|Z\|_* . \tag{11}$$

where $\phi(\cdot)$ is an expanded basis function with an associated kernel function.

$$K(i, j) = \phi(i)^T \phi(j) . \tag{12}$$

The form and parameters of the function $\phi(\cdot)$ are an assumption on the structure of the observations. Ideally, $\phi(\cdot)$ is chosen such that all observations are well approximated in the expanded basis space with a linear low-dimensional approximation while still maintaining the relationship between observations. As in all kernel methods, the accuracy of the approximation of manifolds is dependent on the ability of the kernel to fit the data.

3.1.2 Structured kernel similarity design for clustering

Based on the low-rank representation, the methods of constructing structured kernel similarity are presented in this section. The low-rank transformation of the raw data offers possibilities for designing kernels that incorporate the underlying structure of the data.

In order to exploit this structure we consider some specific PSD kernels, which are basically the dot product, i.e.

$$\omega_{ij} = \frac{z_i^T z_j}{\|z_i\|\|z_j\|} . \tag{13}$$

where ω_{ij} is the similarity between observations i and j and z_i and z_j are the i^{th} and j^{th} columns of Z, respectively. The value of ω_{ij} is the magnitude of the cosine of the angle between the vectors z_i and z_j. Given the near block-diagonal structure of the KLRR matrix, observations lying on independent subspaces have a very small similarity.

One issue with this similarity function is that it is undefined if either z_i or z_j is identically zero. Therefore, we can define $\omega_{ij} = 0$ if either z_i or z_j is zero. With this convention, it is possible to demonstrate that this similarity satisfies the properties of PSD, i.e.

$$\begin{aligned} \tilde{K}(z_i, z_j) &= z_i^T z_j \frac{1}{\|z_i\|}\frac{1}{\|z_j\|} = \tilde{K}_1(z_i, z_j)h(z_i)h(z_j) \\ &= \tilde{K}_1(z_i, z_j)\tilde{K}_2(z_i, z_j). \end{aligned} \tag{14}$$

Since both $\tilde{K}_1$ and $\tilde{K}_2$ are valid PSD kernels, $\tilde{K}$ is also a valid PSD kernel. The similarity proposed in (13) captures the structure of the observations, however, the scaling information is lost. In order to incorporate the structural information while preserving spatial relationships in the observation space, we propose the PSD kernel:

$$ss_{ij} = K(x_i, x_j) = \frac{z_i^T z_j}{\|z_i\|\|z_j\|} e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} . \tag{15}$$

Two observations have a large similarity only if they lie on the same manifold and have a small geometric distance. If x_i and x_j lie on independent manifolds, according to the structure of the KLRR matrix, the angle between the observations is small, and therefore the similarity is also small. Alternatively, if the observations lie on the same low-dimensional manifold but have a large geometric distance, the exponential term drives the similarity to a small value.

Based on the definition of similarity presented in (15), the following equation can be used for defining the distance between two observations:

$$d(x_i, x_j) = \sqrt{ss_{ii} + ss_{jj} - 2ss_{ij}} . \tag{16}$$

The distance between observations defined by (16) combines both the structural similarity and Euclidean distance of the data. If and only if two observations lie on the same low-dimensional manifold and have a small distance in the observations space, they are considered having a small distance.

According to the above definitions, a novel structural similarity between observations for AP is defined as follows:

$$s_{ij} = -d(x_i, x_j) = -\sqrt{s_{ii} + s_{jj} - 2s_{ij}} . \tag{17}$$

3.2 Semi-supervised affinity propagation clustering

In AP, the similarity matrix, whose definition directly affects the performance of the clustering algorithm, is an important input reflecting the similarity between data points. In view of this, the idea of semi-supervision was introduced into this study; pairwise constraints were used to perform logical extension of the unknown data points and guide the update of the similarity matrix. There are two kinds of pairwise constraints, must-link, where the two data points must belong to the same cluster, i.e. $M=\{(x_i, x_j)\}$, and cannot-link, where two data points should not be in the same cluster, i.e. $C=\{(x_i, x_j)\}$ [16]. The detailed rules for updating the matrix are as follows.

(1) For the data point pairs in priori information that meet the must-link constraint and the data point pairs newly accord with the must-link constraint after logical extension, perform similarity update as below.

$$(x_i, x_k) \notin M \delta (x_i, x_j) \in M \delta (x_j, x_k) \in M \Rightarrow (x_i, x_k) \in M. \quad (18)$$

(2) For the data point pairs in priori information that meet the cannot-link constraint, perform similarity update as below.

$$(x_i, x_j) \in C \Rightarrow s(i, j) = \text{-}\infty \delta s(j, i) = \text{-}\infty. \quad (19)$$

(3) Perform global adjustment to the unknown data points based on the principle of the shortest path according to the results of steps 1 and 2. If there is a data point connects to both data points in a data point pair pending for adjustment, and the sum of the similarities between this data point and the two data points in the pair is greater than the similarity of the data point pair, update the similarity of the data point pair to the sum.

$$(x_i, x_j) \notin \{M \cup C\} \Rightarrow s(i, j) = \max\{s(i, j), s(i, k) + s(k, j)\}. \quad (20)$$

(4) Correct the results of step 3 locally according to the cannot-link constraint.

$$(x_i, x_j) \notin \{M \cup C\} \delta (x_i, x_k) \in C \delta (x_k, x_j) \in M \Rightarrow (x_i, x_j) \in C. \quad (21)$$

The updated similarity matrix would more accurately reflect the similarity between data points and the clustering result would also be improved.

3.3 Fireworks explosion optimization algorithm

3.3.1 Basic idea

Fireworks explosion optimization algorithm is an adaptive bi-directional search optimization algorithm that was designed based on the idea of fireworks explosion. The algorithm generates a certain number of firework bombs in the search space and executes the operation of explosion to each fireworks bomb; the sparks generated by the explosion then explore the neighbourhood of the original fireworks bomb. The positions of the sparks are evaluated with the validity index, and positions with higher effectiveness are considered close to the optimal solution of the objective function. Then, the optimum position is selected as the origin of the next round of fireworks explosion. Meanwhile, according to the convergence rate, the algorithm adaptively adjusts the radius of fireworks explosion to balance its global searching and local searching abilities. Eventually, through continuous iteration, the sparks of fireworks explosion will concentrate near the optimal solution to the problem. When the termination condition is met, the spark position with the highest effectiveness is the optimal solution.

3.3.2 Algorithm description

The value of the preference parameter p of AP clustering algorithm is in the range of $(-\infty, 0]$, and the corresponding scope of the number of clusters is $[1, m]$, where m is the number of samples. In order to improve search efficiency and eliminate unnecessary computation, the upper limit of p was set to the median p_{mid} of all input similarities. Meanwhile, to ensure the quality of initial fireworks, the initial search space of the fireworks explosion optimization algorithm was set as $[p_{\min}, p_{\text{mid}}]$, where $p_{\min}$ is the minimum of the input similarities. By observing the results of a large number of experiments, as shown in Tab. 1, such configuration of the parameter was proved feasible.

Table 1 Clustering number of $p_{\min}$ and p_{mid}

Data Sets	Clustering number		
	Ture number	*The number in* p_{mid}	*The number in* $p_{\min}$
Balance	2	31	17
Pima	2	46	11
Iris	3	12	5
Wine	3	12	5
Ionosphere	2	47	9
Ecoli	5	28	10
Glass	6	14	9
Haberman	2	31	10
Seeds	3	17	9

As seen in Tab. 1, when the value of p is set to p_{mid}, the number of clusters produced by AP is much larger than the actual number of clusters. On the other hand, when the value of p is set to $p_{\min}$, the number of output clusters is less than the actual number. This way, not only the search efficiency is improved, but also the quality of initial fireworks is ensured, thereby avoiding omitting clustering structures.

In a word, the value of preference parameter p has a significant influence on the clustering results. Although there is no direct relationship between them, an obvious correlation can be observed, i.e. the number of clusters increases with increasing p value and decreases with the decrease of p value. Due to the diversity of data sets, the orders of magnitude of the similarities between data points may be different, and such difference directly affects the value of the preference parameter p. Therefore, this difference in the orders of magnitude should be considered for the selection of the fireworks explosion radius, so that the validity and rationality of the explosion can be ensured. Meanwhile, based on progressive cognition of the clustering structures, the algorithm adaptively adjusts the scopes of its forward and backward

searches during iteration to find the optimal clustering structure.

With the above considerations, the fireworks explosion range *er* was defined as follows.

$$er = \begin{Bmatrix} [p_{min}, p_{mid}] & t = 1 \\ [p_b + v \cdot (1 + range) \cdot r / t, & p_b - (1 - v) \cdot r / t] t > 1 \end{Bmatrix}. \quad (22)$$

where t is the number of iterations, p_{min} and p_{mid} are the minimum and median of the input similarities, respectively, p_b is the optimal position for the last explosion, r is the radius of the initial explosion, and v is the ratio of forward search. The values of r and v are calculated by (22) and (23).

$$r = (p_{min} + p_{mid}) / 2. \quad (23)$$

$$v = conv / (conv + div). \quad (24)$$

This definition of the fireworks explosion range allows the range to be reasonably determined according to specific data sets, thereby improving the algorithm's search performance. In addition, by recording the numbers of convergence and divergence during the iteration, the algorithm can adaptively adjust the scopes of its forward and backward searches, and locate the optimal preference space quickly and accurately.

In order to ensure the stability of the algorithm's results, it was set that the sparks generated by fireworks explosion are evenly scattered within the range of the explosion. This setup reduces the storage requirement, accelerates the algorithm's execution, and ensures that plenty of sparks are generated.

For certain numbers of clusters, the corresponding ranges of p value may be wider. In such case, multiple iterations are required before the number of clusters changes, and these iterations are often meaningless. By enlarging the forward searching scope and reduce the backward searching radius, the problem can be solved. An acceleration factor *range*, as defined in (24), was introduced to reduce the computation time.

$$range = \max_t K - \min_t K, \quad (25)$$

where K means the sets of candidate solutions (numbers of clusters) obtained in the t^{th} explosion. Larger range of the candidate solutions indicates more obvious convergent tendency. With the introduction of the acceleration factor, the optimal range of the preference parameter can be located quickly, thereby saving the time spent on unnecessary computation and prevent the algorithm from stagnating during computation.

3.4 The process of the proposed algorithm (SAAP-SS)

Input: Similarity matrix $S(i, j)$, The maximum number of iterations T, The initial explosion radius r, Number of Sparks m, Damping factor λ.
Output: Cluster sets $C=\{C_1,\ldots, C_k\}$,
Step 1 Initialization $r(i, k) = 0$, $a(i, k) = 0$, $\lambda = 0{,}5$, $m = 10$, $T=10000$.
Step 2 Perform low-rank data transformation to find a low rank matrix, Z.
Step 3 Constructing structured kernel similarity s_{ij}.
Step 4 Adjusting the similarity matrix by utilizing the known pairwise constraints.
Step 5 Fireworks exploding in the preferences space.
Step 6 Run AP algorithm and use silhouette to evaluate the position of sparks.
Step 7 Record the optimal position of sparks and its clustering result.
Step 8 Select the optimal position of sparks and return to step 4 until meeting the termination conditions.

4 Experimental results

In this section, we present a set of clustering experiments on many datasets, including four synthetic datasets, six UCI datasets. All experiments were performed with MATLAB 2012b on a computer with Inter (R) Pentium 2,9 GHz processer, 4 GB RAM, 500 GB hard drive, and operating system of Microsoft Windows 7 professional.

4.1 Experimental data

4.1.1 Synthetic datasets

In this part, we considered four synthetic datasets with complex non-spherical shapes clusters, as shown in Fig. 1. These datasets represent some difficult clustering instances because they contain clusters of arbitrary shape and varying densities.

4.1.2 UCI datasets

To verify the feasibility and efficiency of the proposed approach, we performed experiments on 6 UCI datasets, including Iris, Wine, Glass, Ecoli, Seeds and Haberman. The basic information of those UCI datasets is summarized in Tab. 2.

4.2 Validity index

In the experiments, we set the number of clusters equal to the true number of those for all the clustering algorithms. We use the following two popular validity indices to evaluate the performance for all the clustering algorithms.

4.2.1 Silhouette index

Assume a data set with n samples be divided into k clusters C_i (i = 1, 2, …, k), $a(t)$ is the average dissimilarity of sample t in C_j to all other samples in C_j, $d(t, C_i)$ is the average dissimilarity of sample t in C_j to all samples in another cluster C_i, then $b(t) = \min\{d(t, C_i)\}$, i = 1, 2,…, k, $i \neq j$. The formula to calculate the Silhouette index *Sil* of sample t is:

$$Sil(t) = \frac{[b(t) - a(t)]}{\max\{a(t), b(t)\}}. \quad (26)$$

The average *Sil* value of all the samples in a cluster reflects the clustering quality, where the largest average

Sil value represents the best clustering quality and the optimal number of clusters. With a series of *Sil* values corresponding to clustering solutions under different numbers of clusters calculated, the optimal clustering solution is found with the largest *Sil*.

4.2.2 F-measure index

F-measure measures a grammar's accuracy. It considers both the precision P and the recall R of the algorithm: P is the ratio of the number of correct results to the number of all returned results, and R is the ratio of the number of correct results to the number of results that should have been returned. P, R and F-measure (F) are defined as follows.

$$P(P_j, C_i) = \frac{|P_j \cap C_i|}{|C_i|}, \tag{27}$$

$$R(P_j, C_i) = \frac{|P_j \cap C_i|}{|P_j|}. \tag{28}$$

$$F(P_j, C_i) = \frac{2 \cdot P(P_j, C_i) \cdot R(P_j, C_i)}{P(P_j, C_i) + R(P_j, C_i)}, \tag{29}$$

$$F = \sum_j \frac{|P_j|}{N} \max_i F(P_j, C_i). \tag{30}$$

where N is the number of data points. Larger value of the F-measure index indicates that the algorithm is more accurate.

4.3 Comparison and analysis of the results

We compared the performance of the proposed algorithm with AP, FEO-SAP (fireworks explosion optimization-based semi-supervised affinity propagation, an improved approach without using structural similarity) and K-means algorithm. The priori information, accounting for 10 % of the entire data information, was randomly generated from the datasets.

The performances of the compared clustering algorithms on four synthetic datasets are shown in Figs. 4 to 7, where the best performance for each dataset is highlighted. Pictures numbered a, b, c and d represent the clustering results of AP, FEO-SAP, SAAP-SS and K-means respectively.

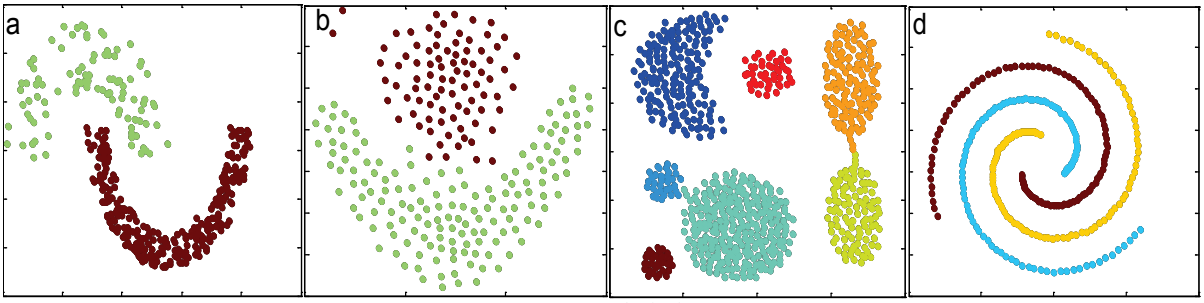

Figure 3 Four synthetic datasets:
(a) Jain dataset (b) Flame dataset (c) Aggregation dataset (d) Spirals dataset

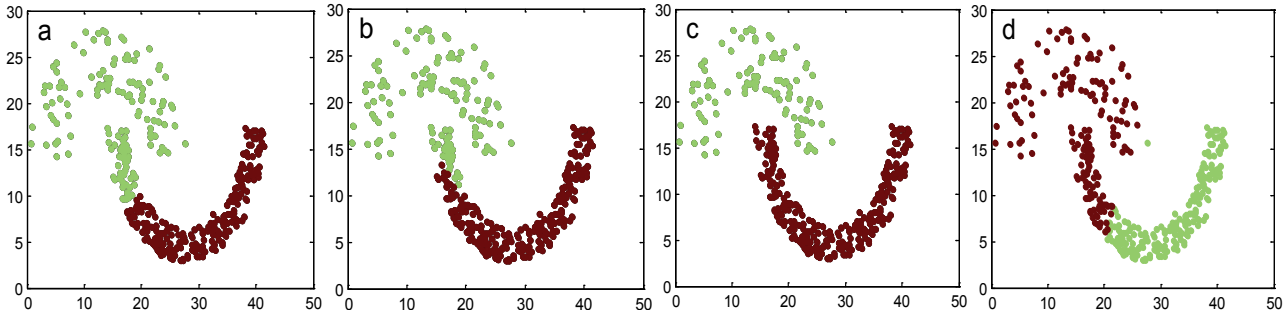

Figure 4 Clustering results on Jain dataset: (a) AP (b) FEO-SAP (c) SAAP-SS (d) K-means

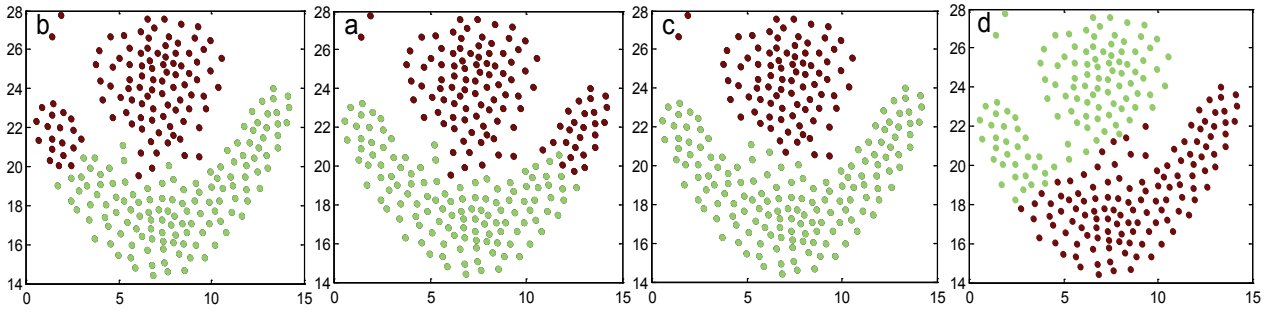

Figure 5 Clustering results on Aggregation dataset: (a) AP (b) FEO-SAP (c) SAAP-SS (d) K-means

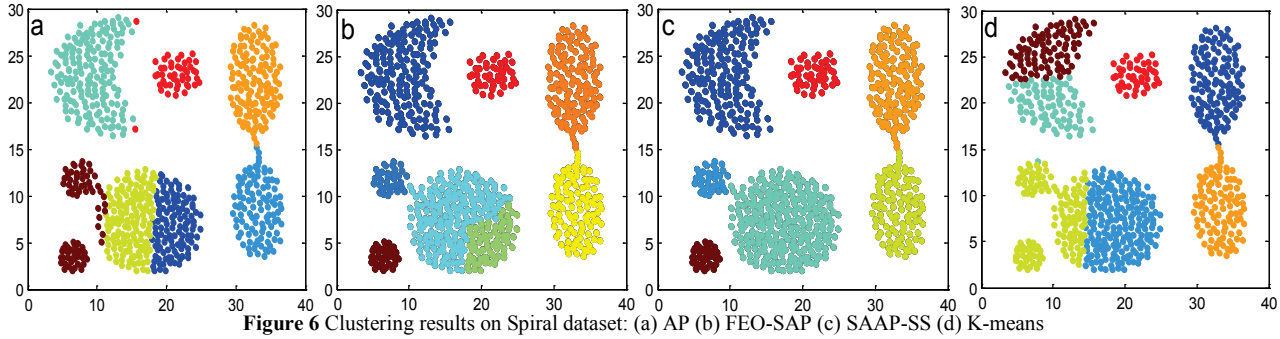

Figure 6 Clustering results on Spiral dataset: (a) AP (b) FEO-SAP (c) SAAP-SS (d) K-means

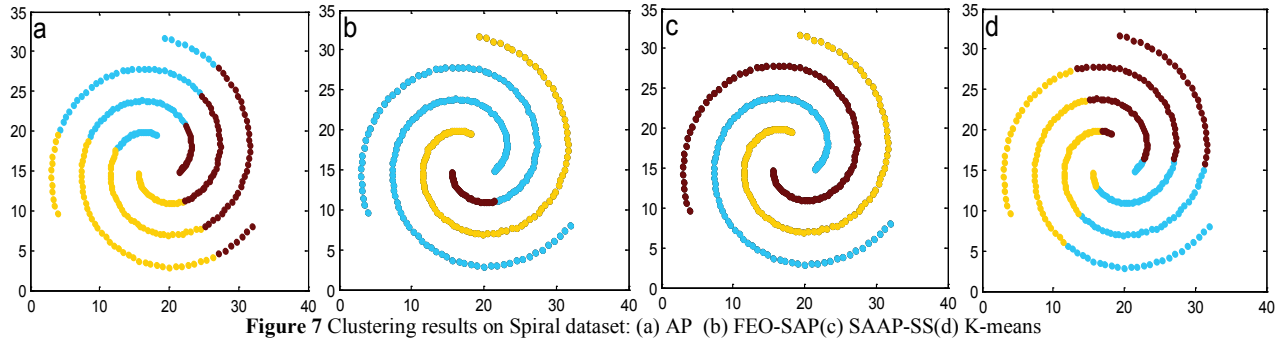

Figure 7 Clustering results on Spiral dataset: (a) AP (b) FEO-SAP(c) SAAP-SS(d) K-means

The following observations were made based on the above clustering results.

AP performs as well as K-means for the clusters with spherical or ellipsoidal structure. However, both algorithms failed to discover non-spherical clusters, as shown clearly in Figs. 4 to7.

The performance of AP is improved by using the priori known labelled data or pairwise constraints to adjust the similarity between data points. Moreover, to a certain extent, FEO-SAP is capable of identifying the underlying clustering structure. However, the improvement is somewhat constrained.

The proposed SAAP-SS algorithm outperforms AP, FEO-SAP and K-means on all the synthetic datasets. It is able to find the underlying structure of data and recognize arbitrary clusters.

Table 2 UCI Datasets

Data Sets	Clustering number			
	Number of samples	*Dimensions*	*True number*	*Sources*
Iris	150	4	3	UCI
Wine	178	13	3	UCI
Glass	214	10	6	UCI
Ecoli	327	7	5	UCI
Seeds	210	7	3	UCI
Haberman	306	3	2	UCI

In detail, the experiments on UCI datasets were performed in three steps. First, an experiment on AP, SAP (semi-supervised affinity propagation), and SAAP-SS algorithm was performed to test whether the best clustering number can be automatically identified. Then, the quality and accuracy of the proposed SAAP-SS were compared with those of AP, FEO-SAP and K-means. Finally, we performed an experiment on the six UCI datasets to analyse the relation between accuracy and the number of constraints and to compare the performance of the four methods above. The values of parameter p were set to the medians of the input similarities.

Table 3 Comparison of clustering number

Data Sets	Clustering number			
	True number	*AP*	*SAP*	*SAAP-SS*
Iris	3	12	8	3
Wine	3	12	6	3
Glass	6	14	9	6
Ecoli	5	28	11	5
Seeds	3	17	6	3
Haberman	2	31	17	2

As seen in Tab. 3, the numbers of clusters by the proposed SSAP-SS algorithm are in complete accordance with the real numbers in all the UCI datasets, while FEO-SAP and original AP failed to match the actual numbers. The results of F-measure index and Silhouette index show that SAAP-SS has superior clustering performance to FEO-SAP, K-means and original AP. This can be attributed to that the novel structural similarity can more accurately describe local neighbourhoods by explicitly incorporating the low-dimensional manifold structure of data. By bi-directionally search the preference space, the proposed algorithm can adaptively find the optimal clustering structure of the datasets and improve the clustering performance. Although the FEO-SAP algorithm has priori information to guide the updating of the similarity matrix, and this does improve the clustering quality and accuracy to a certain extent, it can only perform local adjustment to the similarity matrix due to the limitation of the amount of priori information. Therefore, the FEO-SAP algorithm is not able to comprehensively reflect the similarities among data points and to discover the global clustering structure of data.

Table 4 Comparison of clustering number

Data Sets	Clustering performance							
	AP		FEO-SAP		SAAP-SS		K-means	
	FM	*Sil*	*FM*	*Sil*	*FM*	*Sil*	*FM*	*Sil*
Iris	0,84	0,3848	0,90	0,5135	0,94	0,5233	0,84	0,3645
Wine	0,72	0,3680	0,84	0,5590	0,89	0,5890	0,67	0,3560
Glass	0,73	0,4977	0,76	0,5271	0,86	0,5436	0,72	0,4878
Ecoli	0,78	0,2437	0,84	0,2730	0,91	0,3730	0,72	0,2323
Seeds	0,79	0,3280	0,86	0,4346	0,93	0,5424	0,81	0,3460
Haberman	0,71	0,3390	0,76	0,4021	0,82	0,4921	0,68	0,3350

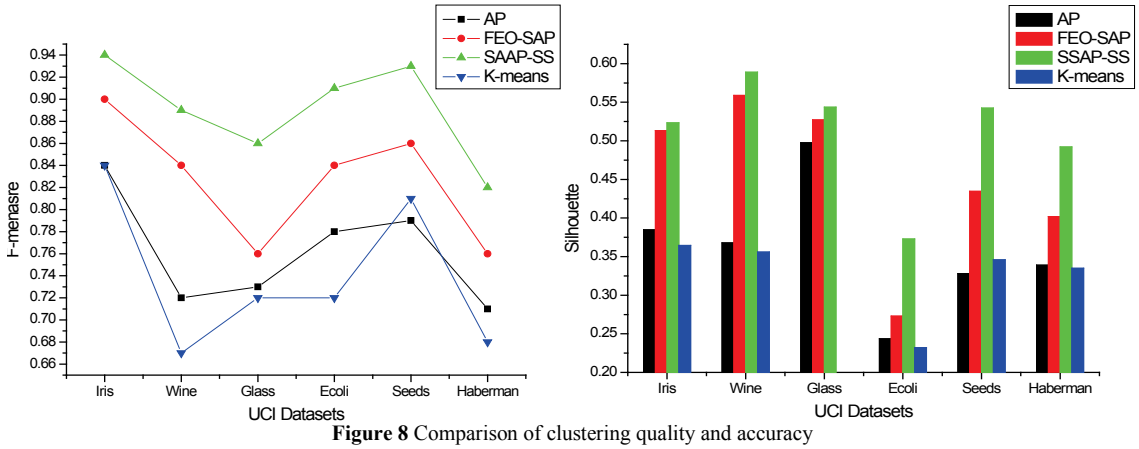

Figure 8 Comparison of clustering quality and accuracy

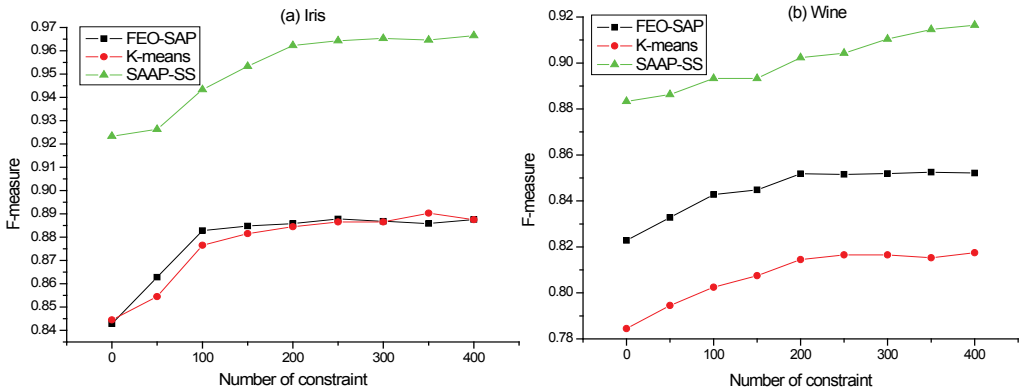

Figure 9 Clustering accuracy on Iris and Wine datasets

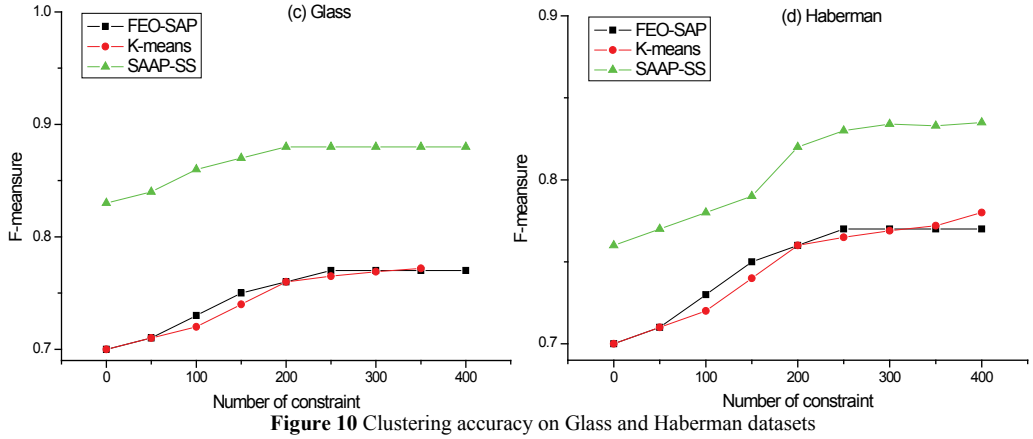

Figure 10 Clustering accuracy on Glass and Haberman datasets

The editor may be guided by the policies of the Journal's editorial board and constrained by such legal requirements as shall then be in force regarding libel, copyright infringement and plagiarism. The editor may confer with other editors or reviewers in making this decision.

Figs. 9, 10 and 11 show the clustering performance of different clustering methods with different numbers of constraints on the six UCI datasets. For all the datasets, the clustering performance of FEO-SAP, K-means and SAAP-SS algorithms gradually improve with the increase in the number of constraints. Moreover, in all cases, the overall performance of SAAP-SS algorithms is much better than those of FEO-SAP and K-means algorithms.

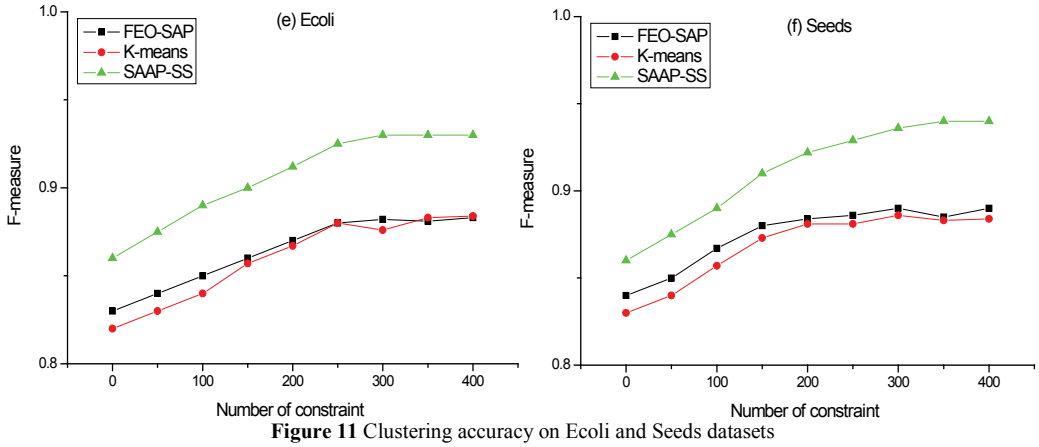

Figure 11 Clustering accuracy on Ecoli and Seeds datasets

4.4 Robustness analysis of the SAAP-SS

According to Eq. (6) and Eq. (7), the damping factor has a significant influence on the robustness of the algorithm. So we examine the robustness of the algorithm by setting different parameters of damping factor λ. The experimental results are shown in Figs. 12 to 14.

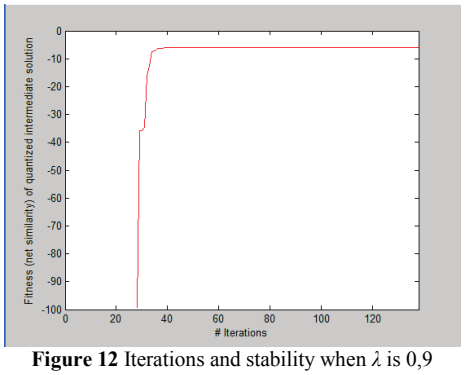

Figure 12 Iterations and stability when λ is 0,9

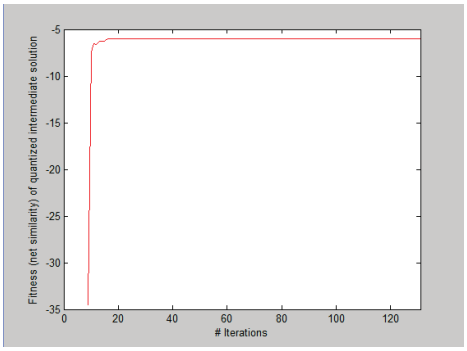

Figure 13 Iterations and stability when λ is 0,7

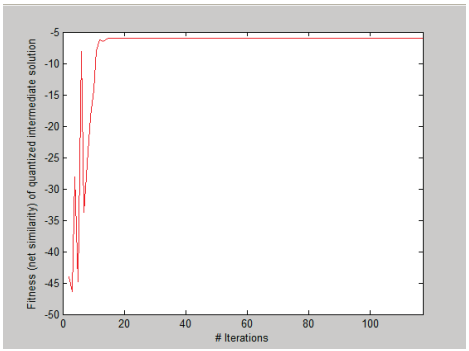

Figure 14 Iterations and stability when λ is 0,5

We randomly generated 200 points in the experiment where settings λ are 0,9, 0,7 and 0,5 respectively. From the results above, the greater the parameter λ is, the more robust the clustering results are. Although a slightly numerical oscillation occurs in the early iterations when λ is 0,5, soon the algorithm is tending to convergence. In conclusion, the proposed algorithm has highly robustness.

5 Conclusions

For the incapability of affinity propagation clustering algorithm to produce ideal clustering results when dealing with complex datasets, a novel adaptive semi-supervised affinity propagation clustering algorithm based on structural similarity (SAAP-SS) was proposed in this paper. We first solved a regularized low-rank representation problem of the observed data by deriving a computationally efficient closed-form solution that allows for handling large sets of observations. Then, we presented the methods of constructing kernels for designing a structured kernel similarity based on the low-rank representation. Moreover we used the priori known labelled data or pairwise constraints to adjust the similarity matrix in order to better reflect the similarity between data points. In addition, the proposed algorithm seeks the optimal clustering structure automatically by adjusting the forward and backward searching scope, and balancing the global and local searching abilities. The experimental results demonstrated that the clustering performance of the proposed SAAP-SS algorithm is superior to that of the original AP, FEO-SAP and K-means.

Acknowledgements

The research was supported by the National Science Foundation of China under grant No. 61202306, 61472049, 61572225 and 61402193, the Application Basis Foundation of Jilin Provincial Science & Technology Department under Grant No. 20100507, 201215119, 20130522177JH and 20130101072JC, the Science Technology Research Foundation of Jilin Province under Grant No. 2012185 and 2012189, the support project on the excellent talent of Jilin Province in colleges and universities under grant No. 2014159, the Society Science Research Foundation of Jilin Province under Grant No. 2013B224 and No. 2014B166.

6 References

[1] Frey, B. J.; Dueck, D. Clustering by Passing Messages Between Data Points. // Science. 315, 5814(2007), pp. 972-976. DOI: 10.1126/science.1136800

[2] Tang, Dongming; Zhu, Qingxin; Yang, Fan; Chen, Ke. Efficient Cluster Analysis Method for Protein Sequences. // Journal of Software. 22, 8(2011), pp. 1827-1837. DOI: 10.3724/SP.J.1001.2011.03848

[3] Wang, Jun; Wang, Shitong; Deng, Zhaohong. A Novel Text Clustering Algorithm Based on Feature Weighting Distance and Soft Subspace Learning. // Chinese Journal of Computers. 35, 8(2012), pp. 1655-1665. DOI: 10.3724/SP.J.1016.2012.01655

[4] Yang, C.; Bruzzone, L.; Guan, R. C. Incremental and Decremental Affinity Propagation for Semi-supervised Clustering in Multispectral Images. // Ieee Transactions on Geoscience And Remote Sensing. 3, (2013), pp. 1666-1679. DOI: 10.1109/TGRS.2012.2206818

[5] Xu, B.; Hu, R.; Guo, P. Combining affinity propagation with supervised dictionary learning for image classification. // Neural Computing and Applications. 22, 7-8(2013), pp. 1301-1308. DOI: 10.1007/s00521-012-0957-7

[6] Tang, Dongming; Zhu, Qingxin; Yang, Fan; Bai, Yong. Solving large scale location problem using affinity propagation clustering. // Application Research of Computers. 27, 3(2010), pp. 841-844.

[7] Saraçli, S. Performance of Rand's C statistics in clustering analysis: an application to clustering the regions of Turkey. // Journal of Inequalities and Applications. 1, (2013),pp. 1-9. DOI: 10.1186/1029-242x-2013-142

[8] Li, Lijuan; Song, Kun; Zhao, Yingkai. Modeling of ARA fermentation based on affinity propagation clustering. // CIESC Jorunal. 62, 8(2011), pp. 2116-2121.

[9] Wang, Weitao; Wang, Baoshan. Quick identification of multilevel similar earthquakes using hierarchical clustering method and its application to Wenchuan northeast aftershock sequence. // Chinese J. Geophys. 55, 6(2012), pp. 1952-1962.

[10] Fujiwara, Y.; Irie, G.; Kitahara, T. Fast algorithm for affinity propagation. // Proceedings of the Twenty-Second international joint conference on Artificial Intelligence / AAAI Press. 22, 3(2011), pp. 2238-2243.

[11] Givoni, I.; Chung, C.; Frey, B. J. Hierarchical affinity propagation. // arXiv preprint arXiv: 1202.3722, 2012.

[12] Feng, Xiaolei; Yu, Hongtao. Semi-supervised affinity propagation clustering based on manifold distance. // Application Research of Computers. 28, 10(2011), pp. 3656-3658.

[13] Wang, Xianhui; Qin, Zheng; Zhang, Xuanping; Gao, Hongjiang. Cluster Ensemble Algorithm Using Affinity Propagation. // Journal of Xi'an Jiaotong University. 45, 8(2011), pp. 1-6.

[14] Liu G, Lin Z, Yan S, et al. Robust recovery of subspace structures by low-rank representation. // Pattern Analysis and Machine Intelligence, IEEE Transactions on. 35, 1(2013), pp. 171-184. DOI: 10.1109/TPAMI.2012.88

[15] Shang, F.; Jiao, L. C.; Shi, J.; Wang, F.; Gong, M. Fast affinity propagation clustering: A multilevel approach. // Pattern recognition. 45, 1(2012), pp. 474-486. DOI: 10.1016/j.patcog.2011.04.032

[16] Gong, Maoguo; Wang, Shuang; Ma, Meng; Cao, Yu; Jiao, Licheng; Ma, Wenping. Two-Phase Clustering Algorithm for Complex Distributed Data. // Journal of Software. 22, 11(2011), pp. 2760-2772. DOI: 10.3724/SP.J.1001.2011.03903

[17] Xiao, Yu; Yu, Jian. Semi-Supervised Clustering Based on Affinity Propagation Algorithm. // Journal of Software. 19, 11(2008), pp. 2803-2813. DOI: 10.3724/SP.J.1001.2008.02803

[18] Chen, X.; Liu, W.; Qiu, H.; Lai, J. APSCAN: A parameter free algorithm for clustering. // Pattern Recognition Letters. 32, 7(2011), pp. 973-986. DOI: 10.1016/j.patrec.2011.02.001

[19] Patel, V. M.; Nguyen, H. V.; Vidal, R. Latent Space Sparse Subspace Clustering. // Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE. (2013), pp. 225-232.

[20] Napoleon, D.; Baskar, G.; Pavalakodi, S. An Efficient Clustering Technique for Message Passing Between Data Points using Affinity Propagation. // International Journal on Computer Science and Engineering. 3, 1(2011).

[21] Song, L.; Dai, B. Robust Low Rank Kernel Embeddings of Multivariate Distributions. // Advances in Neural Information Processing Systems. (2013), pp. 3228-3236.

[22] Santana, R.; Larrañaga, P.; Lozano, J. A. Learning factorizations in estimation of distribution algorithms using affinity propagation. // Evolutionary Computation. 18, 4(2010), pp. 515-546. DOI: 10.1162/EVCO_a_00002

[23] Adler, A.; Elad, M.; Hel-Or, Y. Probabilistic subspace clustering via sparse representations. // Signal Processing Letters, IEEE. 20, 1(2013), pp. 63-66. DOI: 10.1109/LSP.2012.2229705

[24] Yang, C.; Wang, L.; Zhang, S. et al. Accounting for non-genetic factors by low-rank representation and sparse regression for eQTL mapping. // Bioinformatics. 29, 8(2013), pp. 1026-1034. DOI: 10.1093/bioinformatics/btt075

[25] Shang, F.; Jiao, L. C.; Shi, J. et al. Fast density-weighted low-rank approximation spectral clustering. // Data Mining and Knowledge Discovery. 23, 2(2011), pp. 345-378. DOI: 10.1007/s10618-010-0207-5

[26] Wang, L.; Rege, M.; Dong, M. et al. Low-rank kernel matrix factorization for large-scale evolutionary clustering. // Knowledge and Data Engineering, IEEE Transactions on. 24, 6(2012), pp. 1036-1050. DOI: 10.1109/TKDE.2010.258

[27] Liu, G.; Yan, S. Latent low-rank representation for subspace segmentation and feature extraction. // Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE. (2011), pp. 1615-1622.

[28] Vidal, R.; Favaro, P. Low rank subspace clustering (LRSC). // Pattern Recognition Letters. 43, (20149, pp. 47-61.

[29] Zhuang, L.; Gao, H.; Lin, Z. et al. Non-negative low rank and sparse graph for semi-supervised learning. // Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE. (2012), pp. 2328-2335.

[30] Zhou, Shibing; Xu, Zhenyuan; Tang, Xuqing. Method for determining optimal number of clusters based on affinity propagation clustering. // Control and Decision. 26, 8(2011), pp. 1147-1152.

[31] Zhu, H.; Ding, S.; Xu, X.; Xu, L. A parallel attribute reduction algorithm based on Affinity Propagation clustering. // Journal of Computers. 8, 4(2013), pp. 990-997. DOI: 10.4304/jcp.8.4.990-997

[32] Savas, B.; Dhillon, I. S. Clustered low rank approximation of graphs in information science applications. // SDM. (2011), pp 164-175. DOI: 10.1137/1.9781611972818.15

[33] Zhang, H.; Lin, Z.; Zhang, C. et al. Robust latent low rank representation for subspace clustering. // Neurocomputing. (2014). DOI: 10.1016/j.neucom.2014.05.022

[34] Favaro, P.; Vidal, R.; Ravichandran, A. A closed form solution to robust subspace estimation and clustering. // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE. (2011), pp. 1801-1807.

[35] Elhamifar, E.; Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. // Pattern Analysis and Machine Intelligence, IEEE Transactions on. 35, 11(2013), pp. 2765-2781. DOI: 10.1109/TPAMI.2013.57

[36] Lee, K.; Gray, A.; Kim, H. Dependence maps, a dimensionality reduction with dependence distance for high-dimensional data. // Data Mining and Knowledge

Discovery. 26, 3(2013), pp. 512-532. DOI: 10.1007/s10618-012-0267-9

Authors' addresses

Limin Wang, Professor
School of Management Science and Information Engineering,
Jilin University of Finance and Economics,
Changchun 130117, China
E-mail: wlm_new@163.com

Qiang Ji, Postgraduate
School of Management Science and Information Engineering,
Jilin University of Finance and Economics,
Changchun 130117, China
E-mail: a83662830@126.com

Xuming Han, Professor, *Corresponding author*
School of Computer Science and Engineering,
Changchun University of Technology,
Changchun 130012, China
E-mail: hanxvming@163.com