Automatically Affinity Propagation Clustering using Particle Swarm

Xian-hui Wang
School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China
Email: wangxhpaper@gmail.com

[1, 2]Zheng Qin, [1]Xuan-ping Zhang
1. School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, China
2. School of Software, Tsinghua University, Beijing, China

***Abstract*—Affinity propagation (AP) is a clustering algorithm which has much better performance than traditional clustering approach such as *K*-means algorithm. AP can usually find a moderate clustering number, but "moderate" usually may not be the "optimal". If we have found the optimal clustering number of AP, to estimate the input "preferences" (*p*) and the effective corresponding "preferences" (*p*) interval from the data sets is hard. In this paper, we propose a new approach called Automatically Affinity Propagation Clustering (AAP).Our AAP method is absolutely "automatic". AAP represents the issue of finding the optimal AP clustering and the corresponding "preferences" (*p*) interval as an optimization problem of searching optimal solution of the input "preferences" (*p*).AAP searches the "preferences" (*p*) space using Particle Swarm Optimization (PSO) algorithm, and evaluates the particles' fitness using clustering validation indexes. In order to prevent particles from flying out of defined region, we used Boundary Checking (BC) rule to check the validity of particles' positions of PSO. According to lots of AAP's independent runs results, we can find AP's optimal clustering number and estimate the corresponding "preferences" (*p*) interval. One artificial data set and several real-world data sets are presented to illustrate the simplicity and effectiveness of AAP.**

***Index Terms* — Automatically Affinity Propagation Clustering; Particle Swarm Optimization; Clustering Validation Indexes; Boundary Checking (BC) rule; "preferences" (*p*) interval;**

I. INTRODUCTION

Data clustering is the process of identifying natural groupings or clusters within multidimensional data, based on some similarity measure (e.g. Euclidean distance) [1, 2]. It is an important process in Artificial Intelligence (AI), Pattern Recognition and Machine Learning. In the past few decades, clustering has played a central role in a variety of fields, such as data mining, image segmentation, compression, etc [1, 2].

Data clustering techniques can be *hierarchical* or *partitional* clustering[3, 4]. Within each of the types, there exists a wealth of subtypes and different algorithms for finding the clusters.

In *hierarchical* clustering, the output is a tree showing a sequence of clustering, with each luster being a partition of the data set. *Hierarchical* algorithms have two basic advantages. Firstly, the number of cluster need not be specified a *priori*. Secondly, they are independent of the initial conditions. However, *hierarchical* clustering techniques have two main drawbacks. Firstly, they are static. That is, data points assigned to a cluster cannot move to another cluster. In addition to that , they may fail to separate overlapping clusters due to lack of information about the global shape or size of the clusters[5].

Partitional clustering attempt to partition the data set directly into a specified number of clusters. They try to optimize certain criteria (e.g., square-error function).The criterion function may emphasize the local structure of the data, such as by assigning clusters to peaks in the probability density function, or the global structure. Typically, the global criteria involve minimizing some measure of dissimilarity in the samples within each cluster, while maximizing the dissimilarity of different clusters.

The advantages of the *hierarchical* algorithms are the disadvantages of the *partitional* algorithms, and vice versa. *Partitional* clustering techniques are more popular than *hierarchical* techniques. Classical *partitional* clustering technique, such as *K-means* clustering, starts with *K* arbitrary cluster centers, partitions a set of objects into *K* subsets. *K-means* clustering is quite sensitive to the initial selection of data centers, so it is usually rerun many times with different initializations in order to obtain a satisfactory result. Nevertheless, this works well only when the number of clusters is small and chances are good that at least one random initialization is close to a good solution.

In 2007, Brendan J. Frey and Delbert Dueck[6] proposed a new cluster method in ***Science*** called Affinity Propagation Clustering (AP). Affinity propagation has been used to cluster images of faces, detect genes in microarray data, identify representative sentences in this manuscript, and identify cities that are efficiently accessed by airline travel. Affinity propagation can find clusters with much lower error than other methods (like *K-means* cluster and *K-centers* cluster et al). Affinity propagation considers all data points as potential exemplars and it depends on the message passing among

doi:10.4304/jcp.5.11.1731-1738

data points to find cluster. Because of its simplicity, general applicability, and performance, this paper will focus on Affinity Propagation Clustering.

However, AP clustering suffers from the following drawbacks in practical applications:

- When the input "preferences" (***p***) value can be the median of the input similarities of data sets($p=p_m$, default value[6]), AP can find a moderate number of clustering [6], but "moderate" clustering number usually may not be the "optimal" clustering number.
- If we want to find the optimal clustering number, the value of the input "preferences" (***p***) is unknown and to estimate the "preferences" (***p***) from the data sets is hard.
- From Fig .1. (D) of Ref.[6] , we can know that: clustering number is monotonically increasing with the input "preferences" (***p***), but not strictly monotonic increasing .Namely: the relationship between clustering number and input "preferences" (***p***) is not one-to-one mapping, there will be situation that all input "preferences" (***p***) values in an effective "preferences"(***p***) interval correspond to a fixed clustering number. If we want to find the effective input "preferences" (***p***) interval in which the corresponding optimal clustering number is found, to estimate the effective input "preferences" (***p***) interval from the data sets is hard.

In this paper, we propose a new approach called Automatically Affinity Propagation Clustering (AAP). Our AAP method is absolutely "automatic". AAP represents the issue of finding the optimal AP clustering and the corresponding "preferences" (*p*) interval as an optimization problem of searching optimal solution of the input "preferences" (*p*).It searches the "preferences" (*p*) space using Particle Swarm Optimization (PSO) algorithm, and evaluates the particles' fitness using clustering validation indexes. In order to prevent particles from flying out of defined region, we used Boundary Checking(**BC**) rule[7] to check the validity of particles' positions of PSO. Because of PSO's random initialization, we can obtain a lot of results from lots of PSO's independent runs(with different seeds of the random number generator).According to these run results, we can compute the mean values and standard deviations of AP's optimal clustering number. Furthermore, we can figure out the maximum and minimum values of the input "preferences" (*p*) and estimate the corresponding "preferences" (*p*) interval.

The rest of this paper is organized as follows. Section II describes Affinity Propagation Clustering algorithm. Section III describes PSO algorithm. Section IV outlines the proposed Automatically Affinity Propagation Clustering algorithm (AAP), and related parameter setup. Section V describes one artificial data set and four real data sets used for experiments, the compared algorithms and experiments' result. Finally, Section VI concludes the paper.

II. AFFINITY PROPAGATION CLUSTERING(AP)

Affinity Propagation Clustering is a new method, which depends on the message passing among data points to find cluster. It takes the similarities of a group of data as an input and form the best exemplars and a cluster gradually through by exchanges of the true valuable information among data points. The genus of each data point is the sort of its nearest representative points.

The given ***d*** dimension data sets: $X = \{X_1, X_2,...,X_N\}$.

$X_i = \{X_{i1}, X_{i2},...,X_{id}\}(i = 1,2,...,N)$ is a data point, ***N*** is the number of the data points. AP depends on the similar matrix among ***N*** data points $S = [s(i,k)]_{N*N}$ to cluster. In this paper, we take Euclidean distance to measure the similarity.

The similarities $s(i,k)$ between any two points are the negative of Euclidean distance square. For instance, point X_i and point X_k :

$$s(i,k) = -\|X_i - X_k\|^2 \ (i,k = 1,2,...,N; i \neq k) \quad (1)$$

AP set preference $p_k(k = 1,2,...,N)$ to every data point as an input. At the very beginning, $s(k,k) = p_k$. AP assumes that all the data points have the same opportunity to be chosen as representative points of a cluster, therefore, when p_k has the same value ***p***, $s(k,k)$ also have the same value ***p***.

AP can be viewed as searching over valid configurations of the labels $C = \{c_1, c_2,..., c_n\}$ to minimize the energy:

$$E(c) = -\sum_{i=1}^{N} s(i, c_i) \quad (2)$$

Each label c_i indicates the exemplar of the data point ***i***, while $s(i, c_i)$ is the similarity between data point ***i*** and its exemplar c_i.

There have two sorts of information exchange between data points : one is responsibility matrix $R = [r(i,k)]_{N\times N}$ and the other is availability matrix $A = [a(i,k)]_{N\times N}$. The cyclic iteration process of AP is the cyclic renew process of these two kinds of information. Each sort of information includes different kinds of competition.

Algorithm 1 .Affinity Propagation clustering (AP)

Input:

$s(i,k)(i,k = 1,2,...,N; i \neq k)$: The similarity of point ***i*** to point ***k***.

$p = s(j, j) = p_j(j = 1,2,...,N)$: The preferences that data point ***j*** are chosen as a cluster center.

Output:

idx (j)*:** the index of the cluster center for data point ***j.
***dpsim*:** the sum of the similarities of the data points to their cluster centers.
***expref*:** the sum of the preferences of the identified cluster centers.
netsim*:** the net similarity. ***netsim= dpsim + expref.

Process:

Step 1:

Initialize the availabilities matrix $A=\left[a(i,k)\right]_{N\times N}$ to zero:

$$a\left(i,k\right)=0 \qquad (3)$$

Step 2:

Update the responsibilities matrix $R=\left[r(i,k)\right]_{N\times N}$ using the rule:

$$r(i,k)\leftarrow s(i,k)-\max_{k's.t.k'\neq k}\left\{a(i,k')+s(i,k')\right\} \qquad (4)$$

Step 3:

Update the availabilities matrix $A=\left[a(i,k)\right]_{N\times N}$ $(i\neq k)$ using the rule:

$\mathrm{IF} i\neq k,$

$$a(i,k)\leftarrow \min\left\{0,r(k,k)+\sum_{i's.t.i'\notin\{i,k\}}\max\left\{0,r(i',k)\right\}\right\} \qquad (5)$$

Step 4:

Update the self-availability $A=\left[a(i,k)\right]_{N\times N}$ $(i=k)$ using the rule:

$$a(k,k)\leftarrow \sum_{i's.t.i'\neq k}\max\left\{0,r(i',k)\right\} \qquad (6)$$

Step 5:

The message-passing procedure may be terminated after a fixed number of iterations, after changes in the messages fall below a threshold or after the local decisions stay constant for some number of iterations.

Step 6:

Find out the representative points of the cluster. If satisfy $(r(k,k)+a(k,k))>0$, then X_k is the representative points of the cluster. Assign other each data points to an exemplar which it is most similar to.

There have two important input parameters in AP, one is the deviate parameters ***p***, and another is the damping factor λ .The damping factor is avoid of vibration in the renew process of AP, increases can eliminate vibration. λ 's default value is ***0.5***. In message passing, $r(i,k)$ and $a(i,k)$ can be set to λ times its value from the previous iteration plus $(1-\lambda)$ times its prescribed updated value.

III. PARTICLE SWARM OPTIMIZATION(PSO)

Kennedy and Eberhart[8, 9] first introduced particle Swarm Optimization (PSO), a new population-based evolutionary computation technique inspired by social behavior simulation, in 1995. Since it is simple and effective, it has been successfully applied to many fields such as non-linear optimization, neural network and pattern recognition. PSO is a population based iterative search algorithm.

A swarm consists of ***m*** particles moving around in a ***D***-dimensional search space. Each particle adjusts its flying toward a promising area according to its own flying experience and shared social information among particles.

The position of ***i*** particle at ***t*** iteration is represented as: $X_i^{(t)}=(x_{i1},x_{i2},...,x_{iD})$.

During the search process the particle successively adjusts its position toward the global optimum according to two factors:

*(a)*the best position encountered by itself (***pBest***) denoted as $P_i=(p_{i1},p_{i2},...,p_{iD})$;

(b) the best position encountered by the whole swarm (***gBest***) denoted as $P_g=(p_{g1},p_{g2},...,p_{gD})$.

The velocity of ***i*** particle at ***t*** iteration is represented as: $V_i^{(t)}=(v_{i1},v_{i2},...,v_{iD})$.

PSO algorithm updates velocities of all particles using the formula:

$$V_i^{(t+1)}=\omega*V_i^{(t)}+c_1*rand_1()*(P_i-X_i^{(t)})+c_2*rand_2()*(P_g-X_i^{(t)}) \qquad (7)$$

PSO algorithm updates positions of all particles using the formula:

$$X_i^{(t+1)}=X_i^{(t)}+V_i^{(t+1)} \qquad (8)$$

Algorithm 2 .Particle Swarm Optimization (PSO)

Input:

ω : Inertia weight;

c_1 : Cognitive acceleration;

c_2 : Social acceleration;

$rand_1()\sim U(0,1), rand_2\sim U(0,1)$;

V_{min} : Minimum velocity of the particle;

V_{max} : Maximum velocity of the particle;

$[X_{min},X_{max}]$: search space of all particles;

Output:

X_{opt} : The best position of the particles;

$F_{X_{opt}}$: The best fitness value of the particles;

Process:

Step 1:

Initialize (); Initialize m particles' velocity and position;

Step 2:

Evaluation Fitness ($X_i^{(t)}$): Evaluation each particle's quality using ***Fitness function F***;

Step 3:

If needed, update P_i and P_g .

Step 4:

Update Velocities of all particles using above formula (7);

Step 5:

Limit Velocities of all particles:

$$\begin{aligned} &If \quad V_i^{(t+1)} > V_{max}, then\, V_i^{(t+1)} = V_{max}; \\ &If \quad V_i^{(t+1)} < V_{min}, then \quad V_i^{(t+1)} = V_{min}; \end{aligned} \tag{9}$$

Step 6:

Update Positions of all particles using above formula (8);

Step 7:

The iterations procedure may be terminated if (P_g meets problem requirements); or else, return ***Step 2***;

IV. Automatically Affinity Propagation Clustering(AAP)

A. Problem Definition

A pattern is a physical or abstract structure of objects. It is distinguished from others by a collective set of attributes called features, which together represent a pattern.

Let $X = \{X_1,\ X_2,...,X_N\}$ is a set of ***N*** patterns or data points; $X_i = \{X_{i1},\ X_{i2},...,X_{id}\}\ (i = 1,2,...,N)$ is a point of ***d*** dimension. Let each element X_{ij} in X_i corresponds to the ***j***th real-value feature $(j = 1,2,...,d)$ of the ***i***th data points $(i = 1,2,...,N)$.

Given such a set $X = \{X_1,\ X_2,...,X_N\}$, a hard partitional clustering algorithm tries to find a partition $Clu = \{C_1, C_2, \ldots C_K\}$. In the partition, $K(1 \le K \le N)$ is the clustering number of data points. The partitions should maintain three properties:

$$\begin{aligned} &\text{(I)} \quad C_i \neq \varnothing, i = 1,...,K; \\ &\text{(II)} \ \bigcup_{i=1}^{K} C_i = X; \\ &\text{(III)} \ C_i \bigcap C_j = \varnothing,\ \ i, j = 1,...,K \ \text{ and } \ i \neq j; \end{aligned} \tag{10}$$

Since the given data set can be partitioned in a number of ways, maintaining all of the above mentioned properties, we define:

$$\boldsymbol{CA} = \left\{ y \left| \begin{array}{l} y = clu = \{C_1, C_2, \ldots C_K\}, \\ \text{And } clu \text{ satisfy (I),(II),(III)}, K \in [1, N] \end{array} \right. \right\} \tag{11}$$

$\boldsymbol{CA}$ is a set including a variety of partitions. The problem turns out to be one of finding a partition C^* of optimal or near-optimal in $\boldsymbol{CA}$. This is the same as:

$$\boldsymbol{C}^* = \arg_{clu} Optimize \ \ f(X, clu) \tag{12}$$

Where clu is a single partition from the set ***CA***, and f is a statistical-mathematical function that quantifies the goodness of a partition on the basis of the distance measure of the patterns. It has been shown in Ref.[10]that the clustering problem is ***NP-hard*** when the number of clusters exceeds ***3***. Let N_{opt} be the optimum number of clusters, then:

$$N_{opt} = \arg_K \left(\arg_{clu} Optimize \ \ f(X,\ \ clu) \right) \tag{13}$$

Affinity Propagation Clustering is indicated as:

$$[idx,\ netsim,\ dpsim,\ expref] = \text{apcluster}(s,\ p) \tag{14}$$

According to ***formula*** (1), ***formula*** (14) may be indicated as:

$$[idx,\ netsim,\ dpsim,\ expref] = \text{apcluster}'(X,\ p) \tag{15}$$

Each ***idx*** corresponds to a cluster solution. So, a single cluster clu is indicated as:

$$clu = h(idx) \tag{16}$$

Where h is a mapping function that can compute a cluster solution from each ***idx***.

According to ***formula*** (12), (15), (16), then:

$$\boldsymbol{c}^* = \arg_{h(\text{apcluster}(X,p))} Optimize \ f\left(X,\ h\left(\text{apcluster}'(X,p)\right)\right) \tag{17}$$

Let $\boldsymbol{p}^*$ be the value of the input "preferences" (***p***) when finding the optimal clustering number, then:

$$\boldsymbol{p}^* = \arg_p Optimize \ f\left(X,\ h\left(\text{apcluster}'(X,p)\right)\right) \tag{18}$$

B. Clustering Validation Indexes

The main subject of cluster validation is "the evaluation of clustering results to find the partitioning that best fits the underlying data"[11]. Hence, cluster validity approaches are used to quantitatively evaluate the result of a clustering algorithm.

The traditional approach to determine the "optimum "number of clusters is to run the algorithm repetitively using different input values and select the partitioning of data resulting into the best validity measure. Two criteria

that have been widely considered sufficient in measuring the quality of partitioning a data set into a number of clusters are:

1) ***Compactness***: samples in one cluster should be similar to each other and different from samples in other clusters. An example of this would be the variance of a cluster;

2) ***Separation***: clusters should be well separated from each other. An example of this criterion is the Euclidean distance between the cancroids of clusters.

There are several relative validity indices, such as: *variance ratio criterion* (VRC), *Dais-Bouldin* (DB) index, *silhouette index* etc [12]. *Silhouette index* reflects the structure of within-cluster compactness and different-clusters separation. It can not only be used to estimate the optimal number of clustering, but also be applied to evaluate clustering quality. Therefore, we use the *silhouette index* to find the optimal clustering solution.

Suppose a cluster:

$Clu = \{C_1, C_2, \ldots C_K\}, C_j \in Clu,\ j = 1,2,...,K$.

Let $X_i \in C_j,\ X_i = \{X_{i1},\ X_{i2},...,X_{id}\}(i = 1,2,...,N)$.

The average dissimilarity of X_i to all other objects of C_j is denoted by $a(X_i)$.

Now consider another $C_m \in Clu$, $m = 1,2,...,K$, $m \neq j$. The average dissimilarity of X_i to all objects of C_m will be denoted by $d(X_i, C_m)$.

Let $b(X_i) = \min\{d(X_i, C_m)\}(m \neq j)$. The value represents the dissimilarity of X_i to its neighbor cluster, and the silhouette $S_{il}(X_i)$ is given by:

$$S_{il}(X_i) = \frac{b(X_i) - a(X_i)}{\max\{a(X_i), b(X_i)\}} \quad (19)$$

According to $S_{il}(X_i)$, we can compute the average silhouette index value $S_{il}(clu)$ of all X_i. A greater $S_{il}(clu)$ value means the better clustering quality.

C. Particles coding scheme

The particles coding scheme is simple. We encode only ***p***. So the length of a particle is ***N***. If $p_i = p_j$ $(i, j = 1,2,...,N, i \neq j)$, then the length is one. In order to speed up the search, the effective search space of ***p*** is denoted as **[*pmin, pmax*]**. When ***N*** data points are clustered, it is reasonable to consider the maximum of the optimum cluster number as $\sqrt{N}$ [13]. Experimental results show that when the preference value is set to half of the median of ***p***, the number of exemplars identified by AP will always be equal to or larger than $\sqrt{N}$ [14]. So $p_{\max} = p_m / 2$ is comfortable (p_m: the median value of ***p***). ***pmin*** may be set to a lower bound which is computed by **Function *preferenceRange(s)*** [15].

D. The design of fitness function

Because silhouette index can be used to estimate the optimal number of clustering, also be applied to evaluate clustering quality, **fitness function *F*** is set to silhouette index of clusters, i.e.

$$F = S_{il}(clu). \quad (20)$$

According to ***formula*** (15), (16), then:

$$F = S_{il}(clu) = S_{il}\left(h(idx)\right) = S_{il}\left(h\left(\text{apcluster}'(X, p)\right)\right) \quad (21)$$

E. Boundary Checking(BC) rule of PSO

$[p\min, p\max]$ is the valid search space, all feasible solutions must be in this range. However, if there is no direct measure that limits the positions of the particles, the particles still have a chance to fly out of the search space; particle will directly fly towards it and not care about the boundary. In order to prevent particles from flying out of defined region, we used Boundary Checking(BC) rule[7] to check the validity of particles' positions of PSO. Through Boundary Checking (BC) rule, we can restrict particles within defined search space.

The new positions of particles are checked in each iteration. If an element of a new position vector is smaller than X_{min}, then it is set to X_{min}; while greater than X_{max}, and then set to X_{max}.

Boundary Checking (BC) rule of PSO is represented as:

$$\begin{aligned} &If \quad X_i^{(t+1)} > X_{max}, then\ X_i^{(t+1)} = X_{max}; \\ &If \quad X_i^{(t+1)} < X_{min}, then\ X_i^{(t+1)} = X_{min}; \end{aligned} \quad (22)$$

F. The steps of AAP

The steps of AAP as follows:

Algorithm 3 .Automatically Affinity Propagation Clustering (AAP)

Input:

ω: Inertia weight;

c_1: Cognitive acceleration;

c_2: Social acceleration;

$rand_1() \sim U(0,1),\ rand_2 \sim U(0,1)$;

$[X_{min}, X_{max}]$: search space of all particles;

V_{min}: Minimum velocity of the particle;

$V_{min} = X_{min}$;

V_{max}: Maximum velocity of the particle;

$V_{max} = X_{max}$;

$X_{min} = p\min, X_{max} = p\max = = p_m / 2$;

Output:

$X_{opt} = \boldsymbol{p}^*$: Value of the input "preferences" (*p*) when finding the optimal clustering number;

$F_{X_{opt}}$: The best $S_{il}(clu)$ of the optimal clustering number.

Process:

Step 1: Initialize ();

Step 2: Evaluation Fitness ($X_i^{(t)}$): Evaluation each particle's quality using Fitness function $\boldsymbol{F}$= $S_{il}(clu)$;

Step 3: If needed, update P_i and P_g .

Step 4:

Update Velocities of all particles using ***formula*** (7);

Step 5:

Limit Velocities of all particles using ***formula*** (9);

Step 6:

Update Positions of all particles using ***formula*** (8);

Step 7:

Limit Positions of all particles using Boundary Checking (BC) rule of PSO, i.e. using ***formula*** (22);

Step 8: The iterations procedure may be terminated if (P_g meets problem requirements); or else, return ***Step 2***;

V. Experiments and Result

To evaluate the AAP algorithm, we carried out a number of experiments on one artificially generated data set, as well as several real-world data sets. Unless otherwise mentioned, in the following experiments, the AP's input "preferences" (*p*) is set to default value ($p = p_m$) ; PSO' parameters are set as follows: $m = 20, c_1 = 2, c_2 = 2$ .PSO' termination criteria *TC* can be a user-defined maximum number of iterations. In this paper, *TC*=40. Because PSO' random initialization, we have taken 40 independent runs (with different seeds of the random number generator) of AP and AAP algorithms. The results have been stated in terms of the mean values and standard deviations over the 40 runs in each case.

A. Artificial data set-Toy-Problem Data

The Toy-Problem data was twenty-five 2D data points [6, 15].The similarity between every pair of 2D data points was set to the negative squared distance between the points. Table I shows the experimental results of AP and AAP '40 times run respectively and corresponding "preferences" interval in Toy-Problem Data. The experimental results include: algorithm's cluster number, algorithm' preferences (*p*), algorithm's silhouette index. They are expressed by mean and standard deviation. The corresponding "preferences" interval is expressed by maximum and minimum of AAP's preferences.

TABLE I. The results (Mean and Standard Deviation) of AP and AAP'40 times run respectively in Toy-problem and corresponding "preferences" interval (Maximum and Minimum)

Data set		Toy-Problem
Algorithm's Cluster number	AP	3.000±0.000
	AAP	3.000±0.000
Algorithm's Silhouette	AP	0.438 ±0.000
	AAP	0.438 ±0.000
Algorithm's Preferences(*p*)	AP	-16.159±0.000
	AAP	-47.016±15.944
AAP's corresponding "preferences"(*p*) interval	Maximum of AAP's "preferences"	-14.086
	Minimum of AAP's "preferences"	-65.035

Table I shows that both AAP and AP found three clusters and their silhouette index values are same for 0.438. It shows AP and AAP was found with the same quality of clustering results. Although AAP' average preferences value was different with AP, AAP can still find the same quality clustering results in AAP's corresponding "preferences" (*p*) interval. It indicates that AAP is effective.

Figure. 1 shows AP and AAP's preferences value and corresponding clustering number. 'a' is minimum of AAP's preferences. 'b' is maximum of AAP's preferences. 'c' is average value of AP's preferences. 'd' is average value of AAP's preferences.

Figure.1 shows that AP and AAP can find the same clustering number on a, b, c, d of four preferences value. 'c' of AP is between AAP's maximum 'b' and AAP's minimum 'a'. So, we can estimate AAP's preferences interval: [a, b].In the interval of [a, b], AAP can find the same optimal clustering number.

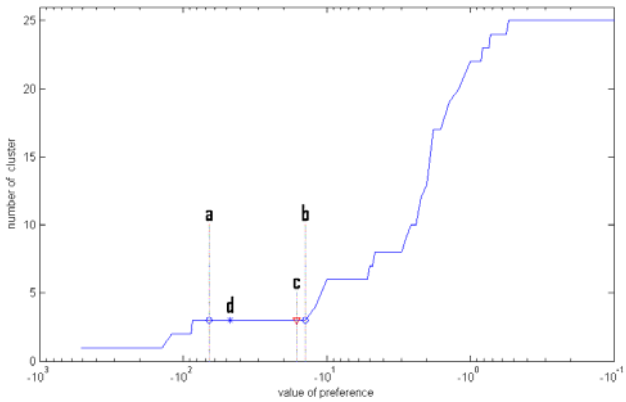

Figure 1. AP and AAP's preferences value and corresponding clustering number

B. *Real world data sets*

We use four real-world data sets from UCI database to compare AP and AAP algorithms. Table II demonstrates the relevant information of the data sets. Because AP and AAP algorithms use Euclidean distance to measure the similarity, so the experiment data sets have be pretreated. If data has missing attribute values, the attribute's average values are filled. In voting data set, votes were numerically encoded as 0.5 for "*yea*", -0.5 for "*nay*", and 0 for "*unknown disposition*".

TABLE II. INFORMATION OF FOUR REAL-WORLD DATA SET FROM UCI

Data set	#Instances	#Features	#Clusters
Iris	150	4	3
Wine	178	13	3
Voting	435	16	2
Glass	214	9	6

Table III shows the experimental results of AP and AAP '40 times run respectively and corresponding "preferences" interval in UCI data sets. The experimental results include: algorithm's cluster number, algorithm' preferences (*p*), algorithm's silhouette index. They are expressed by mean and standard deviation. The corresponding "preferences" interval is expressed by maximum and minimum of AAP's preferences.

Figure 2 shows that comparison of AP and AAP's cluster number in different UCI data sets. Figure 3 shows that comparison of AP and AAP's silhouette index in different UCI data sets.

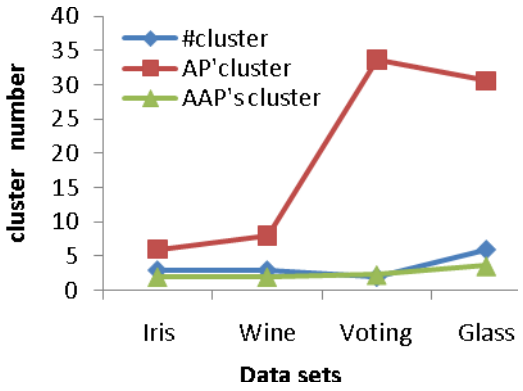

Figure 2. Comparison of AP and AAP's cluster number in data sets

It is indicated that AP usually cannot find the optimum cluster in four UCI data sets from Figure.2. However, AAP actually can find the nearly optimum cluster.

A greater silhouette index means the better clustering quality. So，it is indicated from Figure.3 that the quality of clustering results of AAP is always better than AP by comparison of silhouette index in four UCI data sets.

From Table III, we can get the input preferences' average value while finding the optimal clustering number. Furthermore, we can get the corresponding preferences interval. In the corresponding preferences interval, AAP can always find the nearly optimal clustering number.

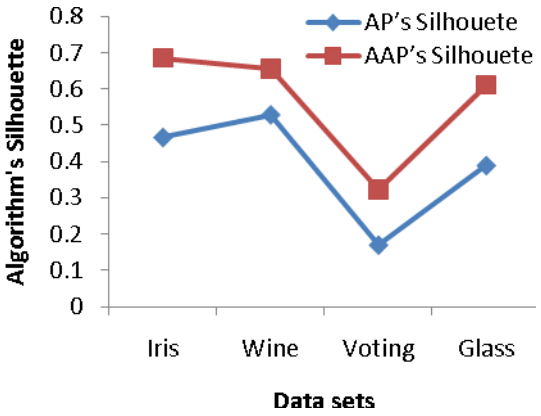

Figure 3. Comparison of AP and AAP's Silhouette index in data sets

VI. CONCLUSION

The paper proposes a new approach called Automatically Affinity Propagation Clustering (AAP).Our AAP method is absolutely "automatic". AAP searches the "preferences" (p) space using Particle Swarm Optimization (PSO) algorithm, and evaluates the particles' fitness using clustering validation indexes. In order to prevent particles from flying out of defined region, we used Boundary Checking (BC) rule to check the validity of particles' positions of PSO. The experiments in one artificial data set and several real-world data sets demonstrate our AAP method's simplicity and effectiveness. AAP can not only find the nearly optimal clustering number, but also estimate the corresponding "preferences" (*p*) interval.

ACKNOWLEDGMENT

The research reported in this paper is funded by National Natural Science Foundation of China (No. 60673024) and also supported by the National Defense 11th-Five-Year Preliminary Research fund.

REFERENCE

[1] R. Xu, D. Wunsch,“ Survey of clustering algorithms,” IEEE Trans Neural Netw, vol.16, pp. 645-678, 2005.

[2] M. G. H. Omran, A. P. Engelbrecht, A. Salman,“ An overview of clustering methods,” Intell Data Anal, vol.11, pp. 583-605, 2007.

[3] L. Yee, Z. Jiang-She, X. Zong-Ben,“ Clustering by scale-space filtering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.22, pp. 1396-1410, 2000.

[4] H. Frigui, R. Krishnapuram,“ A robust competitive clustering algorithm with applications incomputer vision,” Ieee Transactions on Pattern Analysis and Machine Intelligence, vol.21, pp. 450-465, 1999.

[5] A. K. Jain, M. N. Murty, P. J. Flynn,“ Data clustering: A review,” Acm Computing Surveys, vol.31, pp. 264-323, 1999.

[6] B. J. Frey, D. Dueck,“ Clustering by passing messages between data points,” Science, vol.315, pp. 972-976, 2007.
[7] Y. Liu, Z. Qin, X. Wang, et al.,“ Momentum particle swarm optimizer,” Journal of Systems Engineering and Electronics, vol.16, pp. 941-946, 2005.
[8] J. Kennedy, R. Eberhart,“Particle swarm optimization,” Proceedings of IEEE international conference on neural networks.Piscataway,NJ:IEEE, pp.1942-1948,1995.
[9] R. Eberhart, J. Kennedy,“A new optimizer using particle swarm theory,” Proceedings Sixth Symposium on Micro Machine and Human Science.pp.39–43,1995.
[10] P. Brucker,“ On the complexity of clustering problems,” Optimization and operations research, vol.157, pp., 1978.
[11] M. Halkidi, Y. Batistakis, M. Vazirgiannis,“On clustering validation techniques,” 13th International Conference on Scientific and Statistical Database Management (SSDBM 2001).pp.107-145,2001.
[12] E. R. Hruschka, R. Campello, A. A. Freitas, et al.,“ A Survey of Evolutionary Algorithms for Clustering,” IEEE Trans Syst Man Cybern Part C-Appl Rev, vol.39, pp. 133-155, 2009.
[13] J. Yu, Q. Cheng,“ The upper bound of the optimal number of clusters in fuzzy clustering,” Science in China Series F: Information Sciences, vol.44, pp. 119-125, 2001.
[14] X. Zhang, F. Wu, Y. Zhuang,“Clustering by Evidence Accumulation on Affinity Propagation,” pp.1-4,2008.
[15] B.Frey, D.Dueck , “ Affinity Propagation , ” . http://www.psitoronto.edu/affinitypropagation/, 2007.

Xian-hui Wang was born in 1980. He received his M.S. from Xi'an Jiaotong University in Shaanxi of China in 2005. And now he is a Ph.D. candidate of Department of Computer Science & Technology, Xi'an Jiaotong University, Xi'an, Shaanxi in China. His research interests include evolutionary computing, artificial intelligence, pattern recognition etc.

Zheng Qin is a Professor of Tsinghua University and Xi'an Jiaotong University, China. His major research interest includes artificial intelligence, software architecture, data synthesis etc. He has published dozens of research papers on key journals and conferences both at home and abroad, and some books as well.

Xuan-ping Zhang is an associate professor of Xi'an Jiaotong University, China. His major research interest includes artificial intelligence etc. He has published some research papers on key journals and conferences both at home and abroad.

TABLE III. THE RESULTS OF AP AND AAP ’ 40 TIMES RUN RESPECTIVELY（MEAN AND STANDARD DEVIATION）IN REAL-WORLD DATA SETS AND CORRESPONDING “PREFERENCES” INTERVAL

Data set		**Iris**	**Wine**	**Voting**	**Glass**
#Clusters		3	3	2	6
Algorithm’s cluster	AP	6.000 ±0.000	8.000 ±0.000	33.650 ±0.489	30.600 ±0.503
	AAP	2.000 ±0.000	2.000 ±0.000	2.300 ±0.470	3.600 ±0.940
Algorithm’s Silhouette	AP	0.466±0.000	0.528±0.000	0.171±0.001	0.389±0.000
	AAP	0.685±0.002	0.656±0.002	0.322±0.048	0.611±0.011
Algorithm’s Preferences(p)	AP	-5.570 ±0.000	-79620.939±0.000	-7.000±0.000	-5.491±0.000
	AAP	-180.011±65.772	-7183079.934±2543672.904	-254.637±96.117	-150.953±36.722
AAP’s corresponding “preferences”(*p*) interval	Maximum of AAP’s “preferences”	-101.673	-2793305.734	-115.916	-93.888
	Minimum of AAP’s “preferences”	-315.381	-9757343.377	-394.543	-223.886