Expert Systems with Applications 37 (2010) 8679–8684

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Ant clustering algorithm with *K*-harmonic means clustering

Hua Jiang *, Shenghe Yi, Jing Li, Fengqin Yang, Xin Hu

College of Computer Science, Northeast Normal University, Changchun, Jilin 130117, China

ARTICLE INFO

Keywords:
Clustering
K-means
K-harmonic means clustering
Ant clustering algorithm

ABSTRACT

Clustering is an unsupervised learning procedure and there is no a prior knowledge of data distribution. It organizes a set of objects/data into similar groups called clusters, and the objects within one cluster are highly similar and dissimilar with the objects in other clusters. The classic *K*-means algorithm (KM) is the most popular clustering algorithm for its easy implementation and fast working. But KM is very sensitive to initialization, the better centers we choose, the better results we get. Also, it is easily trapped in local optimal. The *K*-harmonic means algorithm (KHM) is less sensitive to the initialization than the KM algorithm. The Ant clustering algorithm (ACA) can avoid trapping in local optimal solution. In this paper, we will propose a new clustering algorithm using the Ant clustering algorithm with *K*-harmonic means clustering (ACAKHM). The experiment results on three well-known data sets like Iris and two other artificial data sets indicate the superiority of the ACAKHM algorithm. At last the performance of the ACAKHM algorithm is compared with the ACA and the KHM algorithm.

1. Introduction

Clustering is a popular data analysis method and plays an important role in data mining. So far it has been widely applied in many fields, like web mining, pattern recognition, machine-learning, spatial database analysis, artificial intelligence, and so on.

The existing clustering algorithms can be simply classified into the following two categories: hierarchical clustering and partitional clustering (Jain, Murty, & Flynn, 1999). The classic *K*-means algorithm (KM) is the most popular clustering algorithm due to its simplicity and efficiency.

Though the *K*-means algorithm is widely used to solve problems in many areas, KM is very sensitive to initialization, the better centers we choose, the better results we get. Also, it is easily trapped in local optimal (Khan & Ahmad, 2004). Recently much work was done to overcome these problems.

Simulated annealing (SA) algorithm was proposed to find the equilibrium configuration of a collection of atoms at a given temperature, and it is always used to solve the combinatorial problems. Simulated annealing heuristic was used with *K*-harmonic means to overcome local optimal problem (Güngör & Ünler, 2007).

Tabu search (TS) is a search method used to solve the combinatorial optimization problems, and the algorithm TabuKHM (Tabu *K*-harmonic means) was developed in 2008 (Güngör & Ünler, 2008).

A hybrid technique was proposed by Kao, Zahara, and Kao (2008). It is based on the *K*-means algorithm, Nelder–Mead simplex search, and particle swarm optimization (*K*–NM–PSO). The *K*–NM–PSO searches for cluster centers of an arbitrary data set as does the KM algorithm, but it can effectively find the global optima.

Particle swarm optimization (PSO) is a popular stochastic optimization technique developed by Kennedy and Eberhart, and a new hybrid algorithm based on PSO and KHM was proposed (Yang & Sun, 2009).

Moreover, some other hybrid heuristic methods like genetic simulated annealing or tabu-search with simulated annealing were ever used with clustering algorithm to solve local optimal problem (Chu & Roddick, 2003; Huang, Pan, Lu, Sun, & Hang, 2001).

In this paper we propose a new algorithm using the Ant clustering algorithm with *K*-harmonic means clustering (ACAKHM). This paper is organized as follows. Section 2 describes the clustering algorithms and gives prominence to the *K*-harmonic means clustering. Section 3 introduces the Ant clustering algorithm. In Section 4, our new algorithm, Ant clustering algorithm with *K*-harmonic means clustering, is presented. Section 5 explains the data sets and the experimental results. Finally, Section 6 summarizes the main conclusion of this study.

2. Clustering

Clustering is an unsupervised learning procedure and there is no a prior knowledge of data distribution (Liu, 2006). It is the

* Corresponding author. Fax: +86 0431 84536331.
E-mail address: jiangh289@nenu.edu.cn (H. Jiang).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.061

process of organizing a set of objects/data into groups called clusters, and the objects within one cluster are similar as much as possible according to a predefined criterion which is always defined with similarity measure.

There are two categories of clustering, the hierarchical clustering and the partitional clustering. The hierarchical clustering can be either agglomerative or divisive. The process of the hierarchical clustering is grouping a set of objects with a sequence of partitions, either from singleton clusters to a cluster including all individual or vice verse. In this paper we pay more attention to the partitional clustering.

2.1. The partitional clustering

The partitional clustering can be described as follows (Xu, 2005):

Given a set of input patterns $X = \{x_1, \ldots, x_i, \ldots, x_N\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \in R^d$ and x_i is the feature (attribute, dimension, or variable) of the data.

Partitional clustering attempts to seek a K-partition of $X, C = \{C_1, C_2, \ldots, C_k\}$, $(k \leqslant N)$, and

(1) $C_j \neq \phi, \quad j = 1, \ldots, k;$
(2) $\bigcup_{j=1}^{k} C_j = X, \quad j = 1, \ldots, k;$
(3) $C_i \cap C_j = \phi, \quad i,j = 1, \ldots, k, \quad i \neq j.$

The above partitional clustering is a kind of hard partitional which means each pattern only belongs to one cluster. However, in most cases the pattern may be allowed to belong to two or more clusters. This is known as fuzzy clustering which characteristic is that a pattern belongs to all clusters with a degree of membership $m(c_j/x_i)$, where c_j is the center of cluster C_j.

$m(c_j/x_i)$ presents the degree of membership of the object x_i belongs to the cluster j. And it satisfies the following constraints:

$$\sum_{j=1}^{k} m(c_j/x_i) = 1, \ \forall i; \quad \sum_{i=1}^{N} m(c_j/x_i) < N, \ \forall j.$$

K-means and K-harmonic means are both center-based clustering algorithms. Particularly, K-means algorithm was first presented over three decades ago. It makes use of minimizing the total mean-squared distance from each point of the data set to the point of closest center. It is hard partitional clustering. On the contrary, the K-harmonic means is fuzzy clustering, and it was presented recently by Zhang, Hsu, and Dayal (1999, 2000) and modified by Hammerly and Elkan (2002). It minimizes the harmonic average from all points in the data set to each center. It will be explained in detail in the following section.

2.2. The K-harmonic means clustering

The K-harmonic means was proposed by Zhang et al. (1999, 2000) and modified by Hammerly and Elkan (2002). It is a center-based clustering algorithm. The difference between KM and KHM is that the KM algorithm gives equal weight to all of the data points and the KHM algorithm every time gives dynamic weight to each data point with a harmonic average. The harmonic average assigns a large weight to a data point that is not close to any centers and a small weight to the data point that is close to one or more centers. Because of this principal, the KHM algorithm is less sensitive to the initialization than the KM algorithm.

Before we introduce the K-harmonic means clustering, we explain some notations used in the procedure of clustering at first (Güngör & Ünler, 2008; Hammerly & Elkan, 2002; Yang & Sun, 2009):

x_i: ith data point, $i = 1, \ldots, N$.
c_j: jth cluster center, $j = 1, \ldots, k$.
$KHM(X, C)$: The objective function of the KHM algorithm.
$m(c_j/x_i)$: The grade of membership value of the point x_i belongs to cluster j.
$w(x_i)$: The grade of influence value of the point x_i to the position of center c_j in the next iteration.

The detail of the K-harmonic means clustering algorithm is shown as follows:

1. Initialize the KHM algorithm by choosing the initial centers randomly.
2. Calculate objective function value according to

$$KHM(X, C) = \sum_{i=1}^{N} \frac{k}{\sum_{j=1}^{k} \frac{1}{\|x_i - c_j\|^p}}, \tag{1}$$

   p is an input parameter and it was proved that KHM works better with the value of $p > 2$.
3. Calculate the membership of each data point x_i to each center c_j according to

$$m(c_j/x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}, \quad m(c_j/x_i) \in [0, 1]. \tag{2}$$

4. Calculate the weight of each point according to

$$w(x_i) = \frac{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{k} \|x_i - c_j\|^{-p}\right)^2}. \tag{3}$$

5. Calculate the new center location with the membership and weight of each point according to

$$c_j = \frac{\sum_{i=1}^{N} m(c_j/x_i) \cdot w(x_i) \cdot x_i}{\sum_{i=1}^{N} m(c_j/x_i) \cdot w(x_i)}. \tag{4}$$

6. Repeat steps 2–5 until it reaches the predefined number of iterations or until the objective function $KHM(X, C)$ does not change significantly.
7. Assign the point x_i to the cluster j with the biggest $m(c_j/x_i)$.

Due to $m(c_j/x_i)$, the KHM algorithm is particularly useful when the boundaries of the clusters are not well separated and ambiguous. Also, the KHM algorithm is less sensitive to the initialization than the KM algorithm.

3. Ant clustering algorithm

The standard Ant clustering algorithm (ACA) was proposed by Lumer and Faieta (1994), and it closely mimics the ant behavior described in Ant-based clustering written by Deneubourg et al. (1991). The idea of the Ant-based clustering is gathering the corpses and sorting the larval of ants. The principle of gathering or sorting is the positive feedback of the behavior of the ants. The ACA technique provides a relevant partition of data without any knowledge of initial cluster centers, which is the merit of this technique. Given that agent ants perform random walks on a two-dimensional grid on which the objects are scattered randomly, and the size of grid is dependent on the number of objects. The agent ants are allowed to move throughout the grid, picking up and dropping the objects influenced by the similarity and density of the objects within the agent ant's immediate current neighborhood, as well as the agent ant's state (whether it is or is not loading an object) (Handl & Meyer, 2007).

The probability of picking up an object will be increased with low density neighborhoods, and decreased with high similarity

Algorithm ACAKHM

1. Scatter the objects randomly on the grid
2. Initialize the related parameters $p, k_p, k_d, \alpha, s, G_{NEW}, G_{ACA}, G_{KHM}$
3. While $(G_{NEW} < G_{NEWMAX})$

    (ACA algorithm)

    3.1 Initialize the position of the agent ants

    3.1.1 If $G_{NEW} = 1$ Place the agent ant with random position on the grid Else

    3.1.2 Place the agent ant on the grid based on the results of KHM algorithm

    3.2 While $(G_{ACA} < G_{ACAMAX})$

    3.2.1 For each agent ant Do

    3.2.1.1 Move the agent ant

    3.2.1.2 If the agent ant is loading an object *i*, then possibly drop the object *i* with p_{drop} Else

    3.2.1.3 Possibly pick up an object with p_{pick}

    3.3 Compute the heap centers c_j

    (KHM algorithm)

    3.4 Take the result of the ACA algorithm as the initial cluster centers of the KHM algorithm

    3.5 While $(G_{KHM} < G_{KHMMAX})$

    3.5.1 For each data point Do

    3.5.1.1 Compute the $KHM(X,C), w(x_i), m(c_j/x_i)$

    3.5.1.2 Update the c_j

    3.6 Assign the data point to the cluster *j* with the biggest $m(c_j/x_i)$

Fig. 1. The ACAKHM algorithm.

among objects in the surrounding area. On the contrary, the probability of dropping an object will be increased with high density neighborhoods. The agent ants and the objects on the grid may be in two situations: (a) one agent ant holds an object *i* and evaluates the probability of dropping it in its current position; (b) an agent ant is unload and evaluates the probability of picking up an object. Finally, the agent ants cluster the objects on the grid to form heaps, the objects in which are similar with each other (Handl, Knowles, & Dorigo, 2003; Kanade & Hall, 2003; Vizine & de Castro, 2005).

The picking up and dropping function is described as follows:

$$p_{pick}(i) = \left(\frac{k_p}{k_p + f(i)}\right)^2, \tag{5}$$

$$p_{drop}(i) = \begin{cases} 2f(i) & \text{if } f(i) < k_d, \\ 1 & \text{otherwise,} \end{cases} \tag{6}$$

where k_p and k_d are constants and that $f(i)$ is similarity density measure for object i in a particular grid locationτ. It is defined as:

$$f(i) = \begin{cases} \frac{1}{s^2} \sum_{j \in Neigh(\tau)} (1 - d(i,j)/\alpha) & \text{if } f(i) > 0, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where s^2 is the size of local neighborhood τ around the agent ant' current position and α is a constant explained the dissimilarity measure $d(i,j)$ (it is the Euclidean distance) between objects *i* and *j*.

4. Ant clustering algorithm with *K*-harmonic means clustering

Although the *K*-harmonic means algorithm (KHM) can overcome the drawback of the KM algorithm in some degree, it also easily runs into local optimal. The Ant clustering algorithm (ACA) provides a partition of data points without any prior knowledge. However, it may take a long time to get a better result. In this section we will propose and describe a new algorithm based on Ant clustering algorithm and *K*-harmonic means clustering (ACAKHM). Because of the characteristic of ACA, the ACAKHM algorithm will avoid trapping in local optimal solution. Meanwhile, the KHM algorithm can receive good initializations from the ACA, and provide better input to ACA in turn to accelerate it. So, the ACAKHM algorithm makes a better use of the advantage of both ACA and KHM algorithm.

The detail of our ACAKHM algorithm is explained in Fig. 1.

5. Experimental studies

We present a set of experiments with C++ on a Pentium (R) CPU 2.50 GHZ with 512 MB RAM. In order to prove the ACAKHM algorithm, five data sets are run with ACA, KHM and ACAKHM algorithm, and the results are evaluated and compared respectively in terms of the objective function of KHM and KM algorithm. The initialization of the parameters used in the ACAKHM algorithm is summarized in Table 1.

5.1. Data sets

The data sets we used, Artset1, Artset2, Glass, Iris, and Wine, are summarized in Table 2.

The Artset1 and Artset2 are artificial data sets, and the other three well-known data sets are all available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/. The details of data sets in the experiments is described as follows:

Table 1
The initialization of the parameters used in the ACAKHM algorithm.

Parameter	Value
k_p	0.15
k_d	0.15
s	5
α	4
G_{NEWMAX}	4

Table 2
Characteristics of data sets considered.

Data set	No. of data classes	No. of features	Size of data set
Artset1	3	2	300 (100,100,100)
Artset2	3	3	300 (100,100,100)
Glass	6	9	214 (70,17,76,13,9,29)
Iris	3	4	150 (50,50,50)
Wine	3	13	178 (59,71,48)

Artset1 ($n = 300$, $d = 2$, $k = 3$), this is an artificial data set. It is a two-featured problem with three unique classes. A total of 300 patterns are drawn from three independent bivariate normal distributions, where classes are distributed according to $N_2\left(\mu = \begin{pmatrix}\mu_{i1}\\ \mu_{i2}\end{pmatrix}, \sum = \begin{bmatrix}0.4 & 0.04\\ 0.04 & 0.4\end{bmatrix}\right)$, $i = 1,2,3$, $\mu_{11} = \mu_{12} = -2$, $\mu_{21} = \mu_{22} = 2$, $\mu_{31} = \mu_{32} = 6$, μ and $\sum$ being mean vector and covariance matrix, respectively. The data set Artset1 is illustrated in Fig. 2.

Artset2 ($n = 300$, $d = 3$, $k = 3$), this is an artificial data set. It is a three-featured problem with three classes and 300 patterns, where every feature of the classes is distributed according to Class1 ~ Uniform (10,25), Class2 ~ Uniform (25,40), Class3 ~ Uniform (40,55). The data set Artset2 is illustrated in Fig. 3.

Glass ($n = 214$, $d = 9$, $k = 6$), which consists of 214 objects and each has nine features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron. There are six different types of glass.

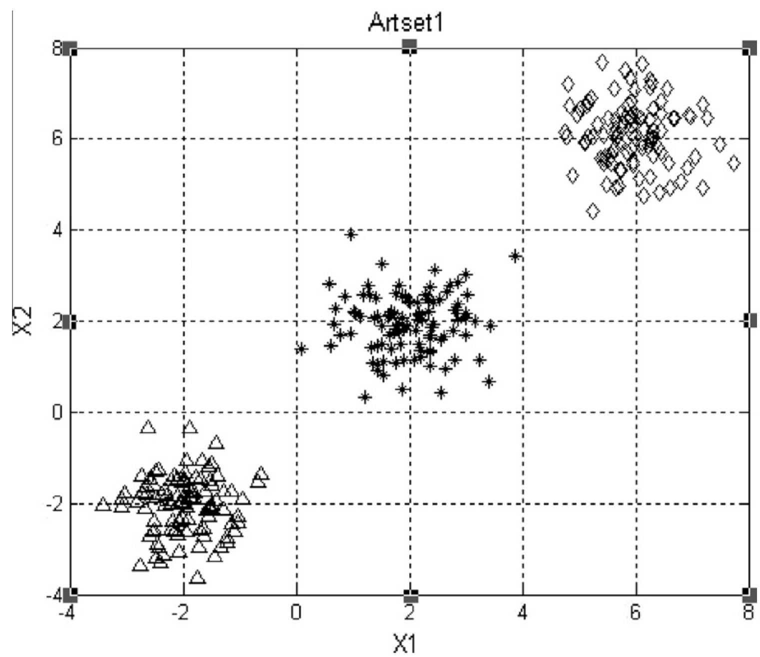

Fig. 2. The dataset of Artset1.

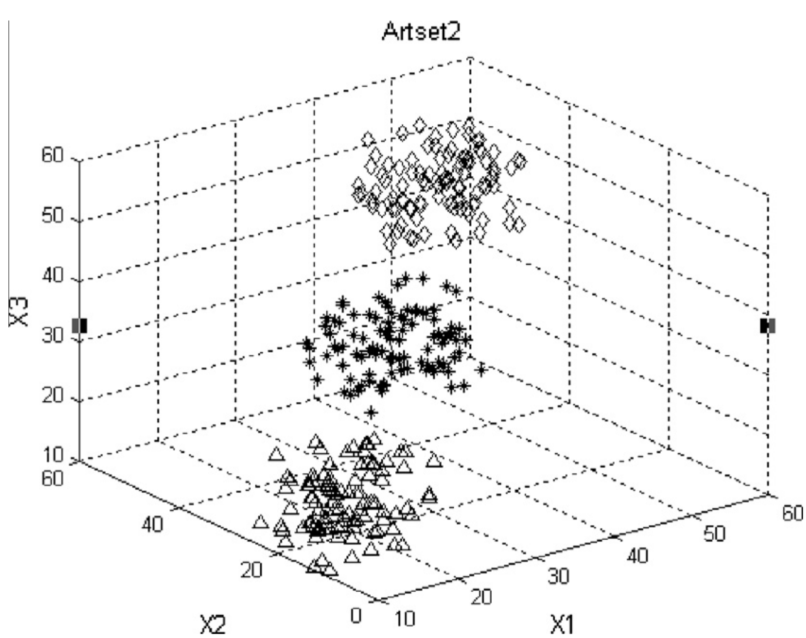

Fig. 3. The dataset of Artset2.

Iris ($n = 150$, $d = 4$, $k = 3$), which consists of three different species of iris flowers: Iris Setosa, Iris Versicolour and Iris Virginica. For each species, 50 samples with four features (sepal length, sepal width, petal length, and petal width) were collected.

Wine ($n = 178$, $d = 13$, $k = 3$), which is the result of a chemical analysis of wines grown in a region in Italy but derived from three different cultivars. The data set consists of 178 objects each with 13 continuous attributes.

5.2. Experimental results

We will compare the results of ACA, KHM, and ACAKHM algorithm in terms of the objective function of the KHM and KM algorithm respectively. In order to prove the performance of the algorithm, we employ the following two criteria:

(a) The objective function of the KHM and KM algorithm, $KHM(X,C)$, $KM(X,C)$. The smaller the objective function is, the higher the quality of clustering algorithm is.
(b) The *F*-Measure, which is related with the precision and the recall from the information retrieval (Dalli, 2003; Handl et al., 2003). The precision and the recall are defined as:

$$p(i,j) = \frac{n_{ij}}{n_j}, \quad r(i,j) = \frac{n_{ij}}{n_i}, \tag{8}$$

where each class i (given by the class labels of the used data set) is regarded as the set of n_i items desired for a query, and each cluster j (generated by the algorithm) is regarded as the set of n_j items retrieved for a query. n_{ij} is the number of data points of the class i within cluster j. For a class i and a cluster j, the *F*-Measure is defined as:

$$F(i,j) = \frac{(b^2+1)\cdot p(i,j)\cdot r(i,j)}{b^2\cdot p(i,j) + r(i,j)}, \tag{9}$$

where we choose $b = 1$ to obtain equal weighting for $p(i,j)$ and $r(i,j)$. The overall *F*-Measure for the data set of size n is given by

$$F = \sum_i \frac{n_i}{n} \max_j\{F(i,j)\}. \tag{10}$$

The bigger the *F*-Measure is, the better the clustering algorithm is.

Table 3
Results of ACA, KHM, and ACAKHM algorithm according to the objective function of KHM on five data sets when $p = 3.5$.

Source	ACA	KHM	ACAKHM
Artset1			
KHM	4849(14638)	807.548(0.08)	**807.514(0.06)**
F-Measure	0.31(0.12)	**1.00(0.00)**	**1.00(0.00)**
Runtime	5.39(0.015)	**0.19(0.005)**	12.53(0.017)
Artset2			
KHM	11,870,423(1157)	**697,005(0.00)**	**697,005(0.00)**
F-Measure	0.37(0.26)	**1.00(0.00)**	**1.00(0.00)**
Runtime	6.51(0.015)	**0.26(0.012)**	12.19(0.027)
Glass			
KHM	76125(1415)	**1871.811617(0.00)**	**1871.811617(0.00)**
F-Measure	0.27(0.3)	**0.40(0.00)**	**0.40(0.00)**
Runtime	4.67(0.010)	**4.27(0.070)**	16.28(0.037)
Iris			
KHM	2089.38(1619)	112.699(3.42)	**112.466(1.2)**
F-Measure	0.23(0.09)	0.77(0.06)	**0.80(0.07)**
Runtime	4.27(0.003)	**0.22(0.011)**	13.70(0.021)
Wine			
KHM	7.28E + 10(3,757,847)	8.56E + 9(490)	**3.54E + 9(169.98)**
F-Measure	0.21(0.2)	0.50(0.00)	**0.52(0.00)**
Runtime	5.46(0.031)	**2.32(0.022)**	15.57(0.050)

In this section, five data sets are employed to prove the effectiveness of the clustering algorithm. We run all data sets 10 times with the ACA, KHM, and ACAKHM algorithm, and the results are averages of 10 runs.

Table 3 summarizes the results of ACA, KHM, and ACAKHM algorithm on five data sets when $p = 3.5$. The quality of clustering is evaluated using the objective function of KHM algorithm and the *F*-Measure. The runtime of the algorithms are additionally provided. Besides, the figure in the brackets shows the standard deviations for 10 independent runs.

Because of the importance of the parameter p to the performance of KHM, we test the clustering algorithm respectively with different values of p. Table 4 shows the results with $p = 4$. The performance of ACA, KHM, and ACAKHM algorithm can be easily compared from Tables 3 and 4. For the data set of Artset1 and Artset2,

Table 4
Results of ACA, KHM, and ACAKHM algorithm according to the objective function of KHM on five data sets when $p = 4$.

Source	ACA	KHM	ACAKHM
Artset1			
KHM	21309.9(10051)	1025.58(29.52)	**993.096(30.8)**
F-Measure	0.35(0.2)	**1.00(0.00)**	**1.00(0.00)**
Runtime	5.44(0.013)	**0.21(0.013)**	12.47(0.018)
Artset2			
KHM	48,547,302(1065)	1,750,831(0.80)	**1,750,830 (0.53)**
F-Measure	0.39(0.37)	**1.00(0.00)**	**1.00(0.00)**
Runtime	6.31(0.012)	**0.25(0.021)**	12.37(0.029)
Glass			
KHM	51114(491)	2556.38(0.18)	**2556.28(0.00)**
F-Measure	0.27(0.17)	0.34 (0.01)	**0.35(0.00)**
Runtime	4.93(0.006)	**4.34(0.050)**	16.27(0.031)
Iris			
KHM	2432.2(138)	117.52(0.22)	**116.73(1.41)**
F-Measure	0.31(0.16)	0.78(0.01)	**0.79(0.01)**
Runtime	4.32(0.002)	**0.32(0.031)**	13.26(0.010)
Wine			
KHM	4.88E + 12 (3.89E + 11)	1.32E + 11 (6.22E + 9)	**8.39E + 10 (4.08E + 9)**
F-Measure	0.20(0.12)	0.50(0.03)	**0.53(0.02)**
Runtime	5.78(0.040)	**2.52(0.025)**	15.81(0.082)

Table 5
Results of ACA, KHM, and ACAKHM algorithm according to the objective function of KM on five data sets.

Source	ACA	KHM	ACAKHM
Artset1			
KM	721.75(822)	247.73(4.81)	**246.81(4.6)**
F-Measure	0.36(0.2)	**1.00(0.00)**	**1.00(0.00)**
Runtime	5.56(0.018)	**0.20(0.003)**	12.96(0.018)
Artset2			
KM	2985(5194)	**2193.13(0.00)**	**2193.13(0.00)**
F-Measure	0.34(0.26)	**1.00(0.00)**	**1.00(0.00)**
Runtime	6.61(0.011)	**0.25(0.011)**	12.08(0.029)
Glass			
KM	601(663)	694.27(251.26)	**572.9(0.00)**
F-Measure	0.28(0.3)	0.38 (0.01)	**0.40(0.00)**
Runtime	**4.35(0.008)**	4.39(0.053)	17.89(0.025)
Iris			
KM	189.9(250)	154.18(5.87)	**143.34(19.5)**
F-Measure	0.33(0.16)	0.78(0.01)	**0.79(0.01)**
Runtime	4.05(0.002)	**0.23(0.013)**	13.59(0.017)
Wine			
KM	32169.86(43447.1)	28449.72(1476.1)	**8764.78(923.2)**
F-Measure	0.28(0.12)	0.51(0.03)	**0.53(0.02)**
Runtime	5.89(0.026)	**2.32(0.019)**	16.28(0.050)

Table 6
Comparison of results in terms of the objective function of KM and KHM algorithm when the data set of Wine is normalized.

Data set (Wine)	ACA	KHM	ACAKHM
KM	111.61(59)	38.63(65.52)	**34.79(24.54)**
KHM	32.93(20)	2.76(4.37)	**2.49(0.6)**
F-Measure	0.23(0.24)	0.52(0.04)	**0.54(0.04)**

Table 7
Comparison of results in terms of the objective function of KM and KHM algorithm when the data set of Iris is normalized.

Data set (Iris)	ACA	KHM	ACAKHM
KM	56.21(29.22)	41.15(5.36)	**40.61(9.15)**
KHM	4.98(1.75)	0.93(0.03)	**0.92(0.02)**
F-Measure	0.28(0.11)	0.79(0.06)	**0.81(0.08)**

the average values of the objective function of KHM and ACAKHM algorithm are almost the same, even the value of the ACAKHM algorithm is smaller than the value of the KHM and both of the *F*-Measures are 1. For the data set of Glass, Iris and Wine, though the ACAKHM algorithm runs with more time than the ACA and the KHM, the $KHM(X,C)$ is lower and the *F*-Measure is higher than the other two algorithms. Particularly, the ACAKHM algorithm can effectively and efficiently find the global optima.

Table 5 summarizes the results of ACA, KHM, and ACAKHM algorithm on five data sets, and the quality of clustering is evaluated using the objective function of KM algorithm and the *F*-Measure. We can see clearly that the performance of the ACAKHM algorithm is better than the results of the other two algorithms.

Let the real data sets of Iris and Wine be normalized according to

$$x = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{11}$$

In terms of $KM(X,C)$, $KHM(X,C)$ and *F*-Measure, the results of ACA, KHM and ACAKHM algorithm are compared in Tables 6 and 7. It can be clearly seen that the performance of ACAKHM algorithm is superior to ACA and KHM when the data sets are normalized.

6. Conclusions

In this paper we present a new algorithm using the Ant clustering algorithm with *K*-harmonic means clustering (ACAKHM). This algorithm makes full use of the merits of the ACA and the KHM algorithm. It overcomes initialization sensitivity of KM and KHM, and reaches a global optimal effectively. Five data sets are employed to prove the performance of ACAKHM algorithm. The result of ACAKHM algorithm is better than the ACA and the KHM algorithm, especially when the data sets are normalized. But the runtime of the ACAKHM algorithm is a little longer than two others.

In the future, we intend to improve the Ant clustering algorithm so as to reduce the runtime of the ACAKHM algorithm. Moreover, we are planning to study the KHM algorithm with other combinatorial optimization techniques.

References

Chu, S., & Roddick, J. (2003). A clustering algorithm using Tabu search approach with simulated annealing for vector quantization. *Chinese Journal of Electronics, 12*(3), 349–353.

Dalli, A. (2003). Adaptation of the *F*-measure to cluster-based Lexicon quality evaluation. In *EACL 2003. Budapest*.

Deneubourg, J.-L., Goss,S., Franks, N., Sendova-Franks, A., Detrain, C., & Chretien, L. (1991). The dynamics of collective sorting: Robot-like ants and ant-like robots. In *Proceedings of the first international conference on simulation of adaptive behavior* (pp. 356–365).

Güngör, Z., & Ünler, A. (2007). *K*-harmonic means data clustering with simulated annealing heuristic. *Applied Mathematics and Computation, 184*, 199–209.

Güngör, Z., & Ünler, A. (2008). *K*-harmonic means data clustering with tabu-search method. *Applied Mathematical Modelling, 32*, 1115–1125.

Hammerly, G., & Elkan, C. (2002). Alternatives to the *k*-means algorithm that find better clusterings. In: *Proceedings of the 11th international conference on information and knowledge management* (pp. 600–607).

Handl, J., Knowles, J., & Dorigo, M. (2003). On the performance of ant-based clustering. Design and application of hybrid intelligent systems. *Frontiers in Artificial Intelligence and Applications, 104*, 204–213.

Handl, J., & Meyer, B. (2007). Ant-based and swarm-based clustering. *Swarm Intelligence, 1*, 95–113.

Huang, C. H., Pan, J. S., Lu, Z. H., Sun, S. H., & Hang, H. M. (2001). Vector quantization based on genetic simulated annealing. *Signal Processing, 81*, 1513–1523.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computational Survey, 31*(3), 264–323.

Kanade, P.M., & Hall, L.O. (2003). Fuzzy ants as a clustering concept. In *22nd International conference of the north american fuzzy information processing society (NAFIPS)* (pp. 227–232).

Kao, Y. T., Zahara, E., & Kao, I. W. (2008). A hybridized approach to data clustering. *Expert Systems with Applications, 34*, 1754–1762.

Khan, S. S., & Ahmad, A. (2004). Cluster center initialization algorithm for *K*-means clustering. *Pattern Recognition Letters, 25*, 1293–1302.

Liu, B. (2006). *Web Data Mining*. Chicago: Springer. Chapter 4.

Lumer, E.D, & Faieta, B. (1994). Diversity and adaptation in populations of clustering ants. In *Proceedings of the third international conference on the simulation of adaptive behavior (Vol. 3)* (pp. 499–508).

Vizine, A. L., & de Castro, L. N. (2005). Towards improving clustering ants: An adaptive ant clustering algorithm. *Informatica, 29*, 143–154.

Xu, R. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*, 645–678.

Yang, F. Q., & Sun, T. L. (2009). Particle swarm optimization based *K*-harmonic means data clustering. *Expert Systems with Applications, 36*, 9847–9852.

Zhang, B., Hsu, M., & Dayal, U. (1999). *K*-harmonic means – A data clustering algorithm. Technical Report HPL-1999-124. Hewlett-Packard Laboratories.

Zhang, B., Hsu, M., & Dayal, U. (2000). *K*-harmonic means. In *International workshop on temporal, spatial and spatio-temporal data mining, TSDM2000. Lyon, France*, September 12.