Journal of Information & Computational Science 12:10 (2015) 4113–4124 July 1, 2015
Available at http://www.joics.com

An Improved K-Means Algorithm Combined with Chaotic Particle Swarm Optimization Algorithm ⋆

Chunqin Gu [a], Qiang Chen [b,*], Qian Tao [b,*]

[a] *Department of Computer Science, Zhongkai University of Agriculture and Engineering, Guangzhou 510255, China*

[b] *Department of Computer Science, Guangdong University of Education, Guangzhou 510303, China*

Abstract

A clustering algorithm combining Chaotic Particle Swarm Optimization (CPSO) with K-Means (CPSO-KM) is proposed. It features better search efficiency than K-Means, PSO and CPSO. The K-Means algorithm cannot guarantee convergence to global optima and suffer in local optimal clusters centers because it is sensitive to initial clusters centers. CPSO can find global optimal solution; meanwhile K-Means can achieve local optima. The CPSO-KM algorithm utilizes both the global search capability of CPSO and the local search capability of K-Means. CPSO-KM algorithm has been tested with three synthetic data sets and four classical data sets from UCI. Experimental results show better performance of the CPSO-KM as compared to K-Means, PSO and CPSO.

Keywords: K-Means; Clustering; Particle Swarm Optimization; Chaotic

1 Introduction

Clustering is a common unsupervised learning method, which partitions a group of objects (instances) into groups (clusters) such that objects in the same cluster are similar to each other and dissimilar to the objects in other clusters. K-Means [1, 2] algorithm partitions the groups of given objects into Kclusters based on a distance metric. The K-Means algorithm is easy to implement and very efficient, so it is used widely in many application areas in recent years such as image processing [3-5], video processing [6, 7], document processing [8-10].

The main drawback of the K-Means algorithm is that the clustering result is sensitive to the initial clusters centers and may converge to the local optima [11]. In recent years, swarm intelligent

⋆This work is supported by the Postdoctoral Science Foundation of China (No. 2013M542219), Research Fund for the Doctoral Program of Higher Education of China (No. 20120171110034), Guangdong Province Natural Science Fund Projects (No. S2012020011067), Doctoral Foundation of Guangdong University of Education (No. 2012ARF05), and Guangdong Province Colleage Specialty Construction Projects (No. 2013LYM_0062), Guangdong Provincial Key Platform and Major Scientific Research Projects of Education (No. 2014KQNCX169, 2014KTSCX194).

*Corresponding author.
Email addresses: cq_c@gdei.edu.cn (Qiang Chen), tao66@139.com (Qian Tao).

1548–7741 / Copyright © 2015 Binary Information Press
DOI: 10.12733/jics20106797

algorithms have been combined with K-Means and applied on many clustering problems, because of the ability of global search of the swarm intelligent algorithms, such as GA [12-15], PSO [16-18], ACO [19, 20]. The combination of swarm intelligent algorithms and K-Means algorithm can take advantage of both global search ability of swarm intelligent algorithms and local search ability of K-Means algorithm. CPSO [21] in our previous literature has been verified that it has the advantage of more precise global search ability and more fast convergence speed. The CPSO is used to obtain better clusters centers for initial clusters centers, then K-Means algorithm is used based on the initial clusters centers found by CPSO.

The rest of the paper is organized as follows. Section 2 describes the model of clustering problem. Section 3 introduces the relative theory of clustering problem. Section 4 presents our clustering algorithm combining CPSO with K-Means (CPSO-KM). Section 5 illustrates experimental results. Finally, Section 6 makes conclusion.

2 Model of Clustering Problem

The clustering problem is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Given a data objects set $DS = \{X_1, X_2, ..., X_N\}$, where $X_j = (x_j^1, x_j^2, ..., x_j^L)$, L is the dimension of a data object, a clustering problem tries to find a K-partition of DS, $C = \{C_1, C_2, ..., C_K\}$, such that the similarity of the data objects in the same cluster is maximum and the difference of the data objects between different clusters center is maximum. The objective function of clustering problem is evaluated based on the Sum of Squared Error (SSE), which is defined as

$$\min \sum_{i=1}^{K} \sum_{\forall x_j \in C_i} D^2(x_j, \overline{C_i}), \overline{C_i} = \frac{1}{|C_i|} \sum_{\forall x_j \in C_i} x_j \tag{1}$$

$$\bigcup_{i=1}^{K} C_i = DS, \quad C_i \neq \phi \tag{2}$$

$$C_i \cap C_j = \phi, i \neq j \text{ and } i, j \in \{1, 2, ..., K\} \tag{3}$$

where $D(x, y)$ denotes the Euclidean distance between x and y, $\overline{C_i}$ is the mean of data objects in the cluster C_i.

3 Relative Theory

3.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO), which was developed by Kennedy and Eberhart in 1995 [22, 23], is a population-based swarm intelligence algorithm. The PSO simulates the social behavior of birds flocking and fish schooling. In PSO, each particle i represents a candidate solution in the solution space of D dimensions, which has two vectors: a position vector $X_i = [x_i^1, x_i^2, ..., x_i^D]$ and a velocity vector $V_i = [v_i^1, v_i^2, ..., v_i^D]$.

During the evolutionary process, the velocity vector and the position vector of particle i on

dimension d at iteration $t+1$ are updated as

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 \cdot r_1 \cdot (pBest_i^d(t) - x_i^d(t)) + c_2 \cdot r_2 \cdot (gBest^d(t) - x_i^d(t)) \quad (4)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (5)$$

where d=1, 2, ..., D represents each dimension of the search space, ω is inertia weight, c_1 and c_2 are cognitive learning coefficient and social learning coefficient, respectively, r_1 and r_2 are two uniform random numbers in the range of [0, 1], $pBest_i^d(t)$ is the position on dimension d with the best fitness found up to the tth iteration for the particle i, $gBest_i^d(t)$ is the best position on dimension d found by the whole particle swarm. The inertia weight ω in Eq. (4) is usually updated as

$$\omega = \omega_{max} - (\omega_{max} - \omega_{min}) \times g/g_{max} \quad (6)$$

where ω_{max} and ω_{min} are initial and final weight, and set to 0.9 and 0.4, respectively [24]. g is the current evolutionary generation number, g_{max} is the maximum number of generations, and set to 1000. c_1 and c_2 are set 1.8, respectively. During the evolutionary process, the velocity of each particle on dimension d is restricted to the range of $\left[-vmax^d, vmax^d\right]$, $vmax^d \in \Re^+$. Thus, if the velocity $v_i^d(t)$ exceeds $vmax^d$, it is reassigned to $vmax^d$. Otherwise, if the velocity $v_i^d(t)$ is lower than -max^d, it is reassigned to -max^d. If $vmax^d$ is too large, particles may miss good solutions. On the other hand, if $vmax^d$ is too small, particles may trap in local optima. The maximum velocity $vmax^d$ is usually set to 20% of the search range [25].

3.2 K-Means

The K-Means algorithm [26] is a straightforward and widely used clustering algorithm. It is simple to implement and run, relatively fast. The K-Means algorithm partitions a given dataset into a user-specified number of clusters.

The standard K-Means algorithm is described as follows:

(1) Initialize the K clusters centers $M = \{m_1, m_2, ..., m_K\}$ randomly;

(2) Assign each data point x_j to the closest cluster based on Euclidean distance. The distance between data point x_j to the closest cluster center m_k is defined as

$$D(x_j, m_k) = \sqrt{\|x_j - m_k\|^2} \quad (7)$$

(3) Recalculate the means of K cluster centers, and obtain new K centers;

(4) Repeat step 2 and 3 until the average change in centroid vectors is less than a predefined value.

4 Clustering Algorithm Combining CPSO with K-Means

In this paper, we develop a hybrid algorithm for solving clustering problems. It can be observed that CPSO is combined into the clustering algorithm. Detailed implementations of the algorithm are proposed and described in this section.

4.1 Particle Representation

To find good initial clusters centers, the clusters centers should be encoded into particle's representation. We employ a representation that each particle is characterized by a $D = Num\_Attri \times Num\_Center$ dimensional real number vector. Num_Attri is the number of attribution of instances in the clustering problems. Num_Center is the number of the center of instances in the clustering problems. Thus the particle is encoded as a sequence of clusters centers. The ith particle in the swarm is in the form of

$$X_i = [x_i^{1,1}, x_i^{1,2}, ..., x_i^{1,Num\_Attri}, x_i^{2,1}, x_i^{2,2}, ..., x_i^{2,Num\_Attri}, ..., x_i^{Num\_Center,1}, x_i^{Num\_Center,2}, ..., x_i^{Num\_Center,Num\_Attri}] \quad (8)$$

where $x_i^{j,k}$ is the kth attribution value of the jth center, $j = 1, 2, ..., Num\_Center$, $k = 1, 2, ..., Num\_Attri$, Fig. 1 shows the particle representation of the ith particle in a 2-D space.

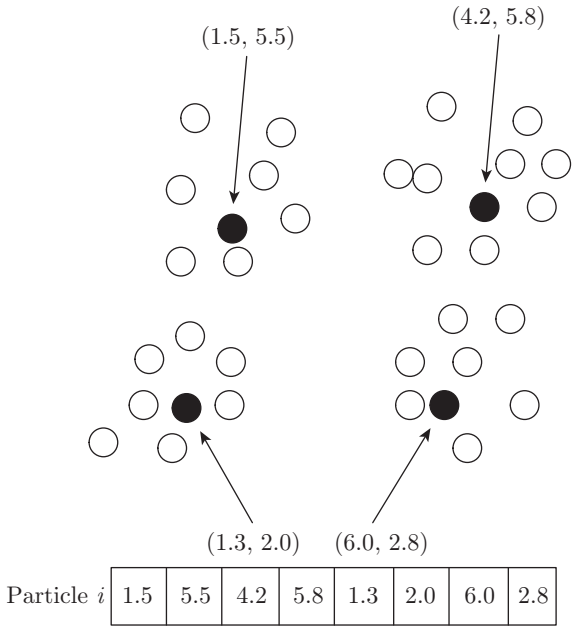

Fig. 1: Particle representation in a 2-D space

4.2 Initialization and Fitness Calculation

In the initialization, the position of all particles are randomly generated, with $(x_i^{j,1}, x_i^{j,2}, ..., x_i^{j,Num\_Attri})$ is the attribution value of the jth center $(j = 1, 2, \ldots, K)$ of the ith particle. The velocity of each particle is also randomly generated, while the maximum velocity of each dimension is set to be 20% of the search range. As described in Section 2, the objective of clustering problem is the minimizing of the SSE.

4.3 CPSO-KM

CPSO-KM is proposed for solving clustering problem, whose flowchart is shown in Fig. 2. During the evolutionary process, the PSO may trap into a local optima solution, then the exploration performance will not be improved. CPSO introduces chaotic mapping with certainty and stochastic

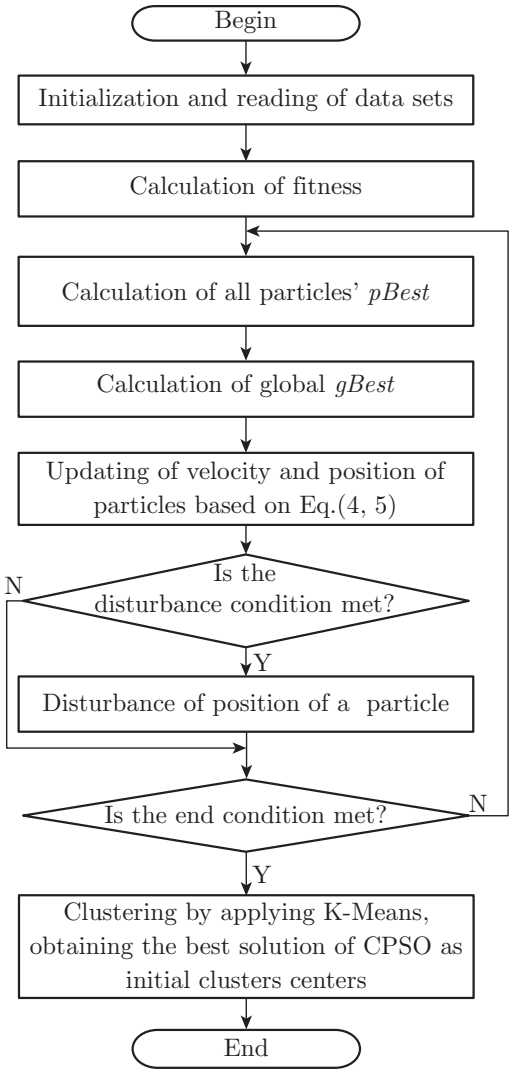

Fig. 2: Flowchart of CPSO-KM

property into PSO in order to improve the global convergence ability [27]. The chaotic mapping method is the chaotic logistic sequence, which is defined as follows:

$$x(t+1) = r \cdot x(t) \cdot (1 - x(t)), r \in N, x(0) \in [0, 1] \tag{9}$$

where r is the control parameter, y is a variable, $r = 4$, $x \notin \{0, 0.25, 0.5, 0.75, 1\}$ and $t = 0, 1, 2, \cdots$.

The radius of chaotic searching region is computed to avoid destroy the excellent solution. The radius can be adaptively adjusted by the distance between *pBest* and *gBest* and variance.

The chaotic searching time is defined as

$$\sigma(pbest, gbest) = \frac{1}{PNum} \sum_{i=1}^{PNum} \sqrt{\sum_{j=1}^{d} (pbest(particle_{i,j}) - gbest_j)^2} \leq \varepsilon_1, \tag{10}$$

where ε_1 is the threshold of the premature convergence.

5 Experiments and Comparisons

In this section, seven data sets are employed to validate the proposed CPSO-KM clustering algorithm. CPSO-KM is compared with conventional K-Means, PSO and CPSO.

5.1 Data Sets and Algorithm Configuration

The seven data sets include three synthetic data sets (SynSet1, SynSet2, SynSet3) and four real data sets from UCI [27]. The three synthetic data sets are generated based on normal distribution with mean vector μ and covariance matrix $\sum$. The parameters for the three synthetic data sets are shown in Table 1. The synthetic data sets are relatively simple, and the real number and centers position of their clusters are known in advance, so experiments on these data sets can reveal the virtues and defects of the algorithms. Fig. 3, 4 and 5 graphically presents the three synthetic data sets.

Table 1: The parameters for three synthetic data sets

Data sets	No. of clusters	No. of dimensions	Size of clusters	Parameters of clusters
SynSet1	3	2	50	$\mu_1 = [10, 10], \sum_1 = 0.4$ $\mu_2 = [7, 7], \sum_2 = 0.4$ $\mu_3 = [3, 3], \sum_3 = 0.4$
SynSet2	4	3	100	$\mu_1 = [30, 30, 30], \sum_1 = 4$ $\mu_2 = [20, 20, 20], \sum_2 = 4$ $\mu_3 = [14, 14, 14], \sum_3 = 4$ $\mu_4 = [5, 5, 5], \sum_4 = 4$
SynSet3	5	3	300	$\mu_1 = [20, 20, 20], \sum_1 = 0.4$ $\mu_2 = [10, 10, 10], \sum_2 = 0.4$ $\mu_3 = [0, 0, 0], \sum_3 = 0.4$ $\mu_4 = [-5, -5, -5], \sum_4 = 0.4$ $\mu_5 = [-10, -10, -10], \sum_4 = 0.4$

The four real data sets that we considered are Iris, Haberman, Hayes-Roth and Wine, which are available online [27]. These data sets are very classical and often used to examine and compare the performances of algorithms in the fields of classification.

(1) The Iris data set consists of three species (Iris Setosa, Iris Versicolour, and Iris Virginica). Each species contains 50 cases with four features (sepal length, sepal width, petal length, and petal width).

(2) The Haberman data set has 306 samples featured by three attributes (Age of patient at time of operation, Patient's year of operation from 1900, Number of positive axillary nodes detected). The data set consists of two survival status: class 1 (225 patients survived 5 years or longer), class 2 (81 patients died within 5 year).

(3) The Hayes-Roth data set consists of 132 samples with four attribute information (Hobby, Age, Educational level, Marital status). The data set has three categories: class 1 (51 persons), class 2 (51 persons), class 3 (30 other persons).

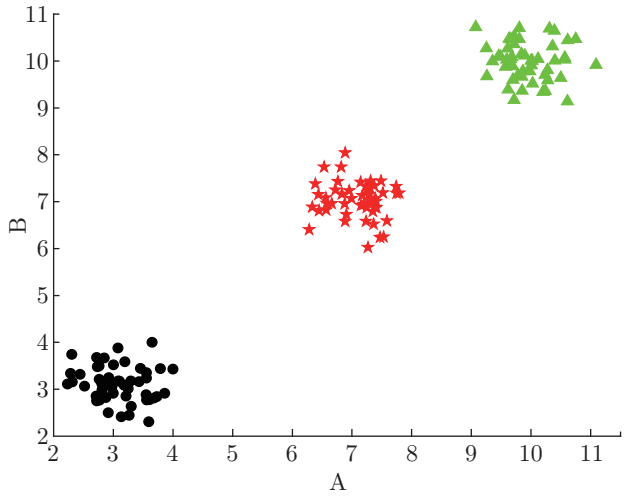

Fig. 3: Clusters of the SynSet1

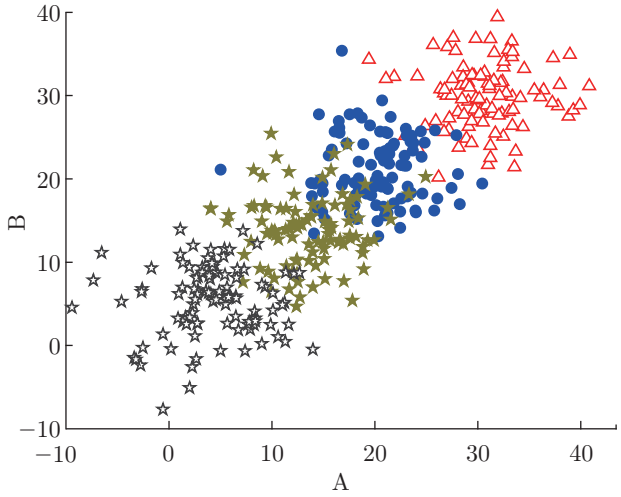

Fig. 4: Clusters of the SynSet2

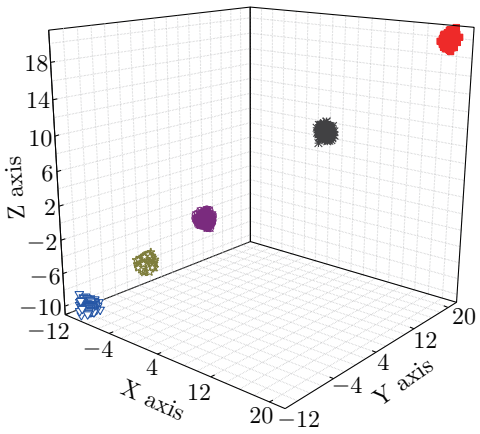

Fig. 5: Clusters of the SynSet3

(4) The Wine data set contains 178 cases characterized by thirteen features (Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, and Proline). The data set consists of three classes: class 1 (59 objects), class 2 (71 objects), and class 3 (48 objects).

The PSO, CPSO and CPSO-KM use the same population size of 20 and the same number of 1000 fitness evaluations (FEs) for a fair comparison. All the experiments are carried out using Visual C++ on the same machine with a Pentium(R) Dual-Core CPU 3.20 GHz, 2.0 GB RAM, and Windows 7 operation system. All the data sets are independently simulated 30 times, and their mean results are compared in order to reduce the statistical errors.

5.2 Performance Comparison

To validate the proposed CPSO-KM, we compare the CPSO-KM with K-Means, PSO and CPSO. Each data set is independently simulated 30 times for purpose of reducing statistical errors. The

statistical results are shown in Table 2 in terms of the mean and standard deviation of the fitness, and boldface in the table indicates the best result obtained.

Table 2: Performance (fitness) comparison of CPSO-KM with other algorithms

Datasets		K-Means	PSO	CPSO	CPSO-KM
SynSet1	Mean fitness	**48.4797**	**48.4797**	**48.4797**	**48.4797**
$(N = 150, C = 3, D = 2)$	Std. Dev	**0**	**0**	**0**	**0**
SynSet2	Mean fitness	1.8643E+04	1.8642E+04	1.8642E+04	**1.8641E+04**
$(N = 400, C = 4, D = 3)$	Std. Dev	2.8918	2.7051	2.3612	**1.7709**
SynSet3	Mean fitness	7.4299E+03	6.3194E+03	5.1954E+03	**2.9608E+03**
$(N = 1500, C = 5, D = 3)$	Std. Dev	5.7699E+03	5.8959E+03	5.7698E+03	**4.7110E+03**
Iris	Mean fitness	92.2927	91.8262	85.4199	**78.9657**
$(N = 150, C = 3, D = 4)$	Std. Dev	26.8443	27.0535	20.3712	**0.2539**
Haberman	Mean fitness	3.0573E+04	3.0563E+04	3.0554E+04	**3.0546E+04**
$(N = 306, C = 2, D = 3)$	Std. Dev	56.8038	57.4285	54.6581	**50.1978**
Hayes-Roth	Mean fitness	288.2780	286.0900	285.6593	**282.5963**
$(N = 132, C = 3, D = 4)$	Std. Dev	11.1871	8.1570	8.1873	**3.9336**
Wine	Mean fitness	2.8104E+06	2.7946E+06	2.7787E+06	**2.7629E+06**
$(N = 178, C = 3, D = 13)$	Std. Dev	7.9610E+04	8.0672E+04	7.8231E+04	**7.1930E+04**

An interesting result is that all algorithms have most reliably achieved the minimal fitness of SysSet1. It is shown that CPSO-KM attains better fitness than K-Means, PSO and CPSO. The CPSO-KM is also observed to obtain the smallest *SD* of the fitness.

5.3 External Criteria Comparison

Because the real partitions of the data sets considered here are already known, the performances of CPSO-KM algorithm can be evaluated by comparing the resulting cluster with the real structures in terms of external criteria. Some commonly used criteria include the Rand index (RI), Jaccard coefficient (JC) and F-Measure ($F - M$).

Assuming that S is a prespecified partition of data sets DS, C is the resulting clusters by applying the CPSO-KM algorithm.

(1) TP is the numbers of pairs of data points (x_i, x_j), where $x_i, x_j \in C_m, x_i, x_j \in S_n, i \neq j$.

(2) FP is the numbers of pairs of data points (x_i, x_j), where $x_i, x_j \in C_m, x_i, \in S_{n1}, x_j \in S_{n2}, i \neq j, n1 \neq n2$.

(3) TN is the numbers of pairs of data points (x_i, x_j), where $x_i \in C_{m1}, x_j \in C_{m2}, x_i \in S_{n1}, x_j \in S_{n2}, i \neq j, m1 \neq m2, n1 \neq n2$.

(4) FN is the numbers of pairs of data points (x_i, x_j), where $x_i \in C_{m1}, x_j \in C_{m2}, x_i, x_j \in S_n, i \neq j, m1 \neq m2$.

Table 3: External criteria comparison of CPSO-KM with other algorithms, in terms of the Rand index and Jaccard coefficient and F-Measure. Given are the mean and standard deviation based on 30 runs

Datasets		K-Means	PSO	CPSO	CPSO-KM
SynSet1 ($N = 150, C = 3, D = 2$)	RI	**1**	**1**	**1**	**1**
	JC	**1**	**1**	**1**	**1**
	F-M	**1**	**1**	**1**	**1**
SynSet2 ($N = 400, C = 4, D = 3$)	RI	0.9445±0.0000	0.9445±0.0000	0.9445±0.0000	**0.9445±0.0000**
	JC	0.7989±0.0002	0.7989±0.0002	0.7989±0.0001	**0.7990±0.0001**
	F-M	0.8882±0.0001	0.8882±0.0001	0.8882±0.0001	**0.8882±0.0001**
SynSet3 ($N = 1500, C = 5, D = 3$)	RI	0.9404±0.0513	0.9503±0.0523	0.9603±0.0513	**0.9801±0.0419**
	JC	0.7264±0.1205	0.7619±0.1438	0.7974±0.1551	**0.8685±0.1496**
	F-M	0.8703±0.1116	0.8920±0.1139	0.9136±0.1116	**0.9568±0.0911**
Iris ($N = 150, C = 3, D = 4$)	RI	0.8817±0.0086	0.8842±0.0113	0.8868±0.013	**0.9094±0.0139**
	JC	0.6996±0.0179	0.7049±0.0237	0.7103±0.0271	**0.7156±0.0292**
	F-M	0.8231±0.0121	0.8239±0.0115	0.8266±0.0160	**0.8339±0.0198**
Haberman ($N = 306, C = 2, D = 3$)	RI	0.7984±0.0135	0.8021±0.0145	0.8089±0.0140	**0.8191±0.0146**
	JC	0.6775±0.0223	0.6787±0.0242	0.6797±0.0254	**0.6874±0.0261**
	F-M	0.8481±0.0147	0.8494±0.0147	08504±0.0149	**0.8480±0.0150**
Hayes-Roth ($N = 132, C = 3, D = 4$)	RI	0.8785±0.0191	0.8822±0.0141	0.8841±0.0126	**0.8925±0.0154**
	JC	0.6504±0.0383	0.6580±0.0300	0.6575±0.0303	**0.6774±0.0334**
	F-M	0.6992±0.0486	0.6093±0.0371	0.6087±0.0375	**0.6333±0.0412**
Wine ($N = 178, C = 3, D = 13$)	RI	0.6945±0.0157	0.6950±0.0145	0.7187±0.0121	**0.7239±0.0156**
	JC	0.4475±0.0232	0.4346±0.0243	0.4120±0.0250	**0.4191±0.0254**
	F-M	0.6183±0.0143	0.6059±0.0142	0.5835±0.0151	**0.5907±0.0158**

(5) P and R are the precision and recall of cluster i.

The two external criteria used in our experiments can then be defined as follows, with larger values indicating a greater similarity of C and S.

(1) Rand index

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

(2) Jaccard coefficient

$$JC = \frac{TP}{TP + FP + FN}$$

(3) F-Measue

$$F - M = \frac{(\beta^2 + 1.0)(P \times R)}{(\beta^2 \times P) + R}, \text{ where } P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}$$

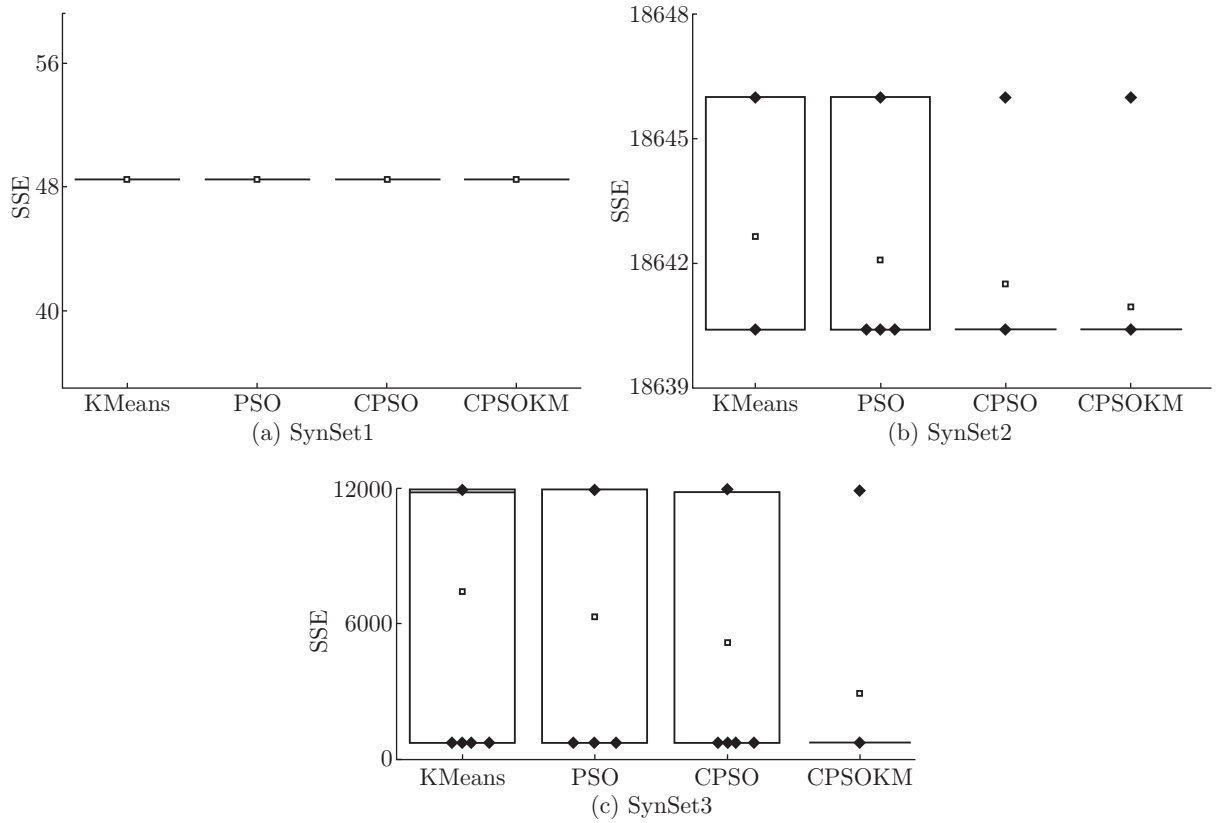

Fig. 6: The boxplots of SSE on the Synset1, Synset2 and Synset3

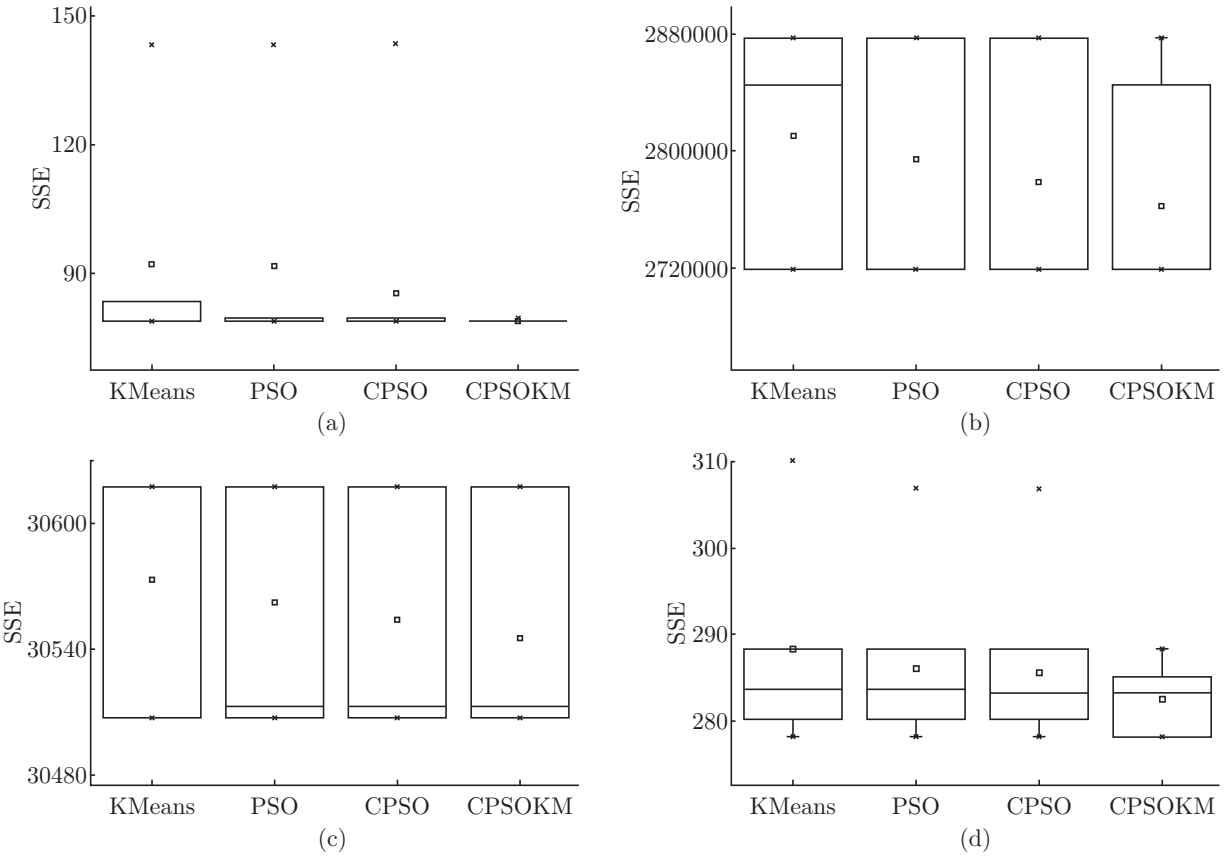

Fig. 7: The boxplots of SSE (a) Iris; (b) Haberman; (c) Hayes-Roth; (d) Wine

In experiments, we set $\beta = 1.0$, which weights precision and recall equally. The performance comparison between CPSO-KM and other algorithms in terms of RI, JC and $F-M$ are illustrated in Table 3. The corresponding boxplots are shown in Fig. 6 and Fig. 7. CPSO-KM is also observed to find the best similarity of S and C.

6 Conclusions

In this paper, a CPSO-KM algorithm is proposed to solve the clustering problem, which is based on the combination of the Particle Swarm Optimization (PSO) and the K-Means algorithm. CPSO and K-Means are run sequentially and the cluster centers obtained by CPSO are the initial centers of the K-Means. The combination method solves the problem of sensitivity to the choice of initial cluster centers. The fitness experiments on three synthetic and four real data sets show that the CPSO-KM algorithm can obtain better performance than K-Means, PSO and CPSO. The experimental results also show that CPSO-KM can attain the better clusters than other algorithms in terms of Rand index, Jaccard coefficient and F-Measure. Future work includes research into the application of the CPSO-KM algorithm to solve more demanding data sets with more complex structure, such as document clustering.

References

[1] E. Forgy, Cluster analysis of multivariate data: Efficiency vs. inter-pretability of classifications, Biometrics, 21(3), 1965, 768-780

[2] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium, 1967, 281-297

[3] S. N. Sulaiman, N. A. M. Isa, Adaptive fuzzy-K-means clustering algorithm for image segmentation, IEEE Transactions on Consumer Electronics, 56(4), 2010, 2661-2668

[4] N. A. M. Isa, S. A. Salamah, U. K. Ngah, Adaptive fuzzy moving K-means clustering algorithm for image segmentation, IEEE Transactions on Consumer Electronics, 55(4), 2009, 2145-2153

[5] M. D. Cuong, B. Javidi, 3D integral imaging reconstruction of occluded objects using independent component analysis-based K-means clustering, Journal of Display Technology, 6(7), 2010, 257-262

[6] B. Maliatski, O. Yadid-Pecht, Hardware-driven adaptive k-means clustering for real-time video imaging, IEEE Transactions on Circuits and Systems for Video Technology, 15(1), 2005, 164-166

[7] T. Chen, S. Chien, Bandwidth adaptive hardware architecture of K-Means clustering for video analysis, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18(6), 2010, 957-966

[8] S. Gowtham, M. Goswami, K. Balachandran, B. S. Purkayastha, An approach for document pre-processing and K-means algorithm implementation, in: Proceedings of the 2014 Fourth International Conference on Advances in Computing and Communications (ICACC), 2014, 162-166

[9] P. Jaganathan, S. Jaiganesh, An improved K-means algorithm combined with Particle Swarm Optimization approach for efficient web document clustering, in: Proceedings of the 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), 2013, 772-776

[10] Y. Liu, Z. Liu, An improved hierarchical K-means algorithm for web document clustering, in: Proceedings of the International Conference on Computer Science and Information Technology, 2008, 606-610

[11] S. Z. Selim, M. A. Ismail, K-means type algorithms: A generalized convergence theorem and characterization of local optimality, IEEE Transaction Pattern Analysis Machine Intelligence, 6, 1984, 81-87

[12] P. Sunantasaengtong, S. Chivapreecha, Mixed K-means and GA-based weighted distance fingerprint algorithm for indoor localization system, in: Proceedings of the 2014 IEEE Region 10 Conference TENCON, 2014, 1-5

[13] W. Zhao, Y. Wang, D. Li, A dynamic feature selection method based on combination of GA with K-means, in: Proceedings of the 2010 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), 2010, 271-274

[14] K. Krishna, M. N. Murty, Genetic K-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 29(3), 1999, 433-439

[15] M. Laszlo, S. Mukherjee, A genetic algorithm using hyper-quadtrees for low-dimensional K-means clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4), 2006, 533-543

[16] M. Davodi, H. R. Modares, E. Reihani, M. Davodi, A. Sarikhani, Coherency approach by hybrid PSO, K-Means clustering method in power system, in: Proceedings of the IEEE 2nd International Power and Energy Conference, 2008, 1203-1207

[17] A. Ahmadyfard, H. Modares, Combining PSO and K-means to enhance data clustering, International Symposium on Telecommunications, 2008, 688-691

[18] S. Sharifian-R, A. Baharian, E. Asgarian, A. Rasooli, A combination of PSO and K-means methods to solve haplotype reconstruction problem, in: Proceedings of the International Conference on Innovations in Information Technology, 2009, 190-194

[19] X. Sun, B. Zhang, L. Yang, S. Li, L. Gao, Hyperspectal image clustering using Ant Colony Optimization (ACO) improved by K-means algorithm, in: Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2, 2010, V2-474-V2-478

[20] K. Aparna, M. K. Nair, Enhancement of K-means algorithm using ACO as an optimization technique on high dimensional data, in: Proceedings of the 2014 International Conference on Electronics and Communication Systems (ICECS), 2014, 1-5

[21] Q. Tao, Z. Huang, C. Gu et al., Optimization of green agri-food supply chain network using chaotic PSO algorithm, in: Proceedings of the 2013 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2013, 462-467

[22] J. Kennedy, R. C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conference Neural Network, 1995, 4, 1942-1948

[23] R. C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the 6th International Symposium on Micromachine Human Science, 1995, 39-43

[24] Y. Shi, R. C. Eberhart, A modified particle swarm optimizer, in Proceedings IEEE World Congress Computer Intelligence, 1998, 69-73

[25] R. C. Eberhart, Y. H. Shi, Particle swarm optimization: Developments, applications and resources, in: Proceedings IEEE Congress Evolutionary Computation, 2001, 81-86

[26] S. Z. Selim, M. A. Ismail, K-means type algorithms: A generalized convergence theorem and characterization of local optimality, IEEE Transaction of Pattern Analysis Machine Intelligent, 6, 81-87

[27] L. Zhang, Y. Zhang, Y. Yang, A decorrelation algorithm based on chaos adaptive, Journal of Electronics & Information Technology, 2009, 31(8), 1825-1829

[28] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/