Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 34 (2008) 1754–1762

Expert Systems with Applications

www.elsevier.com/locate/eswa

A hybridized approach to data clustering

Yi-Tung Kao [a], Erwie Zahara [b,*], I-Wei Kao [b]

[a] Department of Computer Science and Engineering, Tatung University, Taipei City 104, Taiwan, ROC
[b] Department of Industrial Engineering and Management, St. John's University, Tamsui 251, Taiwan, ROC

Abstract

Data clustering helps one discern the structure of and simplify the complexity of massive quantities of data. It is a common technique for statistical data analysis and is used in many fields, including machine learning, data mining, pattern recognition, image analysis, and bioinformatics, in which the distribution of information can be of any size and shape. The well-known K-means algorithm, which has been successfully applied to many practical clustering problems, suffers from several drawbacks due to its choice of initializations. A hybrid technique based on combining the K-means algorithm, Nelder–Mead simplex search, and particle swarm optimization, called K–NM–PSO, is proposed in this research. The K–NM–PSO searches for cluster centers of an arbitrary data set as does the K-means algorithm, but it can effectively and efficiently find the global optima. The new K–NM–PSO algorithm is tested on nine data sets, and its performance is compared with those of PSO, NM–PSO, K–PSO and K-means clustering. Results show that K–NM–PSO is both robust and suitable for handling data clustering.

Keywords: Data clustering; K-means clustering; Nelder–Mead simplex search method; Particle swarm optimization

1. Introduction

Clustering is an important unsupervised classification technique. When used on a set of objects, it helps identify some inherent structures present in the objects by classifying them into subsets that have some meaning in the context of a particular problem. More specifically, objects with attributes that characterize them, usually represented as vectors in a multi-dimensional space, are grouped into some clusters. When the number of clusters, K, is known a priori, clustering may be formulated as distribution of n objects in N dimensional space among K groups in such a way that objects in the same cluster are more similar in some sense than those in different clusters. This involves minimization of some extrinsic optimization criterion.

The K-means algorithm, starting with k arbitrary cluster centers, partitions a set of objects into k subsets and is one of the most popular and widely used clustering techniques because it is easy to implement and very efficient, with linear time complexity (Chen & Ye, 2004). However, the K-means algorithm suffers from several drawbacks. The objective function of the K-means is not convex and hence it may contain many local minima. Consequently, in the process of minimizing the objective function, there exists a possibility of getting stuck at local minima, as well as at local maxima and saddle points (Selim & Ismail, 1984). The outcome of the K-means algorithm, therefore, heavily depends on the initial choice of the cluster centers.

Recently, many clustering algorithms based on evolutionary computing such as genetic algorithms have been introduced, and only a couple of applications opted for particle swarm optimization (Paterlini & Krink, 2006). Genetic algorithms typically start with some candidate solutions to the optimization problem and these candidates evolve towards a better solution through selection, crossover and mutation. Particle swarm optimization (PSO), a population-based algorithm (Kennedy & Eberhart, 1995),

[*] Corresponding author.
E-mail address: erwi@mail.sju.edu.tw (E. Zahara).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.028

simulates bird flocking or fish schooling behavior to build a self-evolving system. It searches automatically for the optimum solution in the search space, and the searching process is not carried out at random. Depending on the nature of the problem, a fitness function is employed to determine the best direction of search. Although evolutionary computation techniques do eventually locate the desired solution, practical use of these techniques in solving complex optimization problems is severely limited by the high computational cost of the slow convergence rate. The convergence rate of PSO is also typically slower than those of local search techniques (e.g. Hooke and Jeeves method, 1961; Nelder–Mead simplex search method, 1965, among others). To deal with the slow convergence of PSO, Fan, Liang, and Zahara (2004) proposed to combine Nelder–Mead simplex search method with PSO, the rationale behind it being that such a hybrid approach will enjoy the merits of both PSO and Nelder–Mead simplex search method. In this paper, we explore the applicability of the hybrid K-means algorithm, Nelder-Mead simplex search method, and particle swarm optimization (K–NM–PSO) to clustering data vectors. The objective of the paper is to show that the hybrid K–NM–PSO algorithm can be adapted to cluster arbitrary data by evolving the appropriate cluster centers in an attempt to optimize a given clustering metric. Results of conducting experimental studies on a variety of data sets provided from several artificial and real-life situations demonstrate that the hybrid K–NM–PSO is superior to the K-means, PSO, and K–PSO algorithms.

2. K-means algorithm

At the core of any clustering algorithm is the measure of similarity, the function of which is to determine how close two patterns are to each other. The K-means algorithm (Kaufman & Rousseeuw, 1990) groups data vectors into a predefined number of clusters on the basis of the Euclidean distance as the similarity measure. Euclidean distances among data vectors are small for data vectors within a cluster as compared with distances to other data vectors in different clusters. Vectors of the same cluster are associated with one centroid vector, which represents the "midpoint" of that cluster and is the mean of the data vectors that belong together. The standard K-means algorithm is summarized as follows:

1. Randomly initialize the k cluster centroid vectors
2. Repeat
   (a) For each data vector, assign the vector to the cluster with the closest centroid vector, where the distance to the centroid is determined using

$$D(\mathbf{x}_p \cdot \mathbf{z}_j) = \sqrt{\sum_{i=1}^{d} (x_{pi} - z_{ji})^2} \tag{1}$$

   where $\mathbf{x}_p$ denotes the pth data vector, $\mathbf{z}_j$ denotes the centroid vector of cluster j, and d subscripts the number of features of each centroid vector.
   (b) Recalculate the cluster centroid vectors, using

$$\mathbf{z}_j = \frac{1}{n_j} \sum_{\forall \mathbf{x}_p \in C_j} \mathbf{x}_p \tag{2}$$

   where n_j is the number of data vectors in cluster j and C_j is the subset of data vectors that form cluster j, until a stopping criterion is satisfied.

The K-means clustering process terminates when any one of the following criteria is satisfied: when the maximum number of iterations has been exceeded, when there is little change in the centroid vectors over a number of iterations, or when there are no cluster membership changes. For the purpose of this research, the algorithm terminates when a user-specified number of iterations has been exceeded.

3. Hybrid K-means, Nelder–Mead, and particle swarm optimization

A hybrid algorithm is developed in this study, which is intended to improve the performances of data clustering techniques currently used in practice. Nelder–Mead (NM) simplex method has the advantage of being a very efficient local search procedure but its convergence is extremely sensitive to the chosen starting point; particle swarm optimization (PSO) belongs to the class of global search procedures but requires much computational effort. The goal of integrating NM and PSO is to combine their advantages while avoiding shortcomings. Similar ideas have been discussed in hybrid methods using genetic algorithms and direct search techniques, and they emphasize the trade-offs between solution quality, reliability and computation time (Renders & Flasse (1996); Yen, Liao, Lee, & Randolph (1998)). This section starts by a brief introduction of NM and PSO, followed by a description of hybrid NM–PSO and our hybrid K-means and NM–PSO (denoted as K–NM–PSO).

3.1. The procedure of NM

This simplex search method, first proposed by Spendley, Hext, and Himsworth (1962) and later refined by Nelder and Mead (1965), is a derivative-free line search method that was particularly designed for traditional unconstrained minimization scenarios, such as the problems of nonlinear least squares, nonlinear simultaneous equations, and other types of function minimization (see, e.g., Olsson & Nelson (1975)). It proceeds as follows: first, evaluate function values at the $(N+1)$ vertices of an initial simplex, which is a polyhedron in the factor space of N input variables. Then, in the minimization case, the vertex with the highest function value is replaced by a newly reflected and better point, which can be approximately located in the negative gradient direction. Clearly, NM can be deemed as a direct line-search method of the steepest descent kind. The ingredients of the replacement process consist of four basic operations: reflection, expansion, contraction, and shrinkage. Through these operations, the simplex can improve itself and come closer and closer to a local optimum point successively. An example of minimization of a function of two variables ($N = 2$) will illustrate the basic procedure of NM. Starting with point B together with an initial step size, an initial simplex design shown as A, B and C is constructed, as illustrated in Fig. 1.

1. Sort the function values at A, B, and C. Suppose$f(C) < f(B) < f(A)$. $f(A)$ is the highest of the three function values and is to be replaced. In this case, a reflection is made through the centroid of $\overline{BC}$ at point D to point E.
2. If $f(E) < f(C)$, an expansion is made to point J. We then keep E or J as a replacement for A, depending on which function value is lower.
3. If $f(E) > f(C)$, a contraction is made to point G or H as a replacement for A, depending on which of $f(A)$ and $f(E)$ is lower, provided that $f(G)$ or $f(H)$ is smaller than $f(C)$.
4. If either $f(G)$ or $f(H)$ is larger than $f(C)$, the contraction has failed and we then perform a shrinkage operation. The shrinkage operation reduces the size of the simplex by moving all but the best point C halfway towards the best point C. We then have new points A and B. Go back to step 1.

The process terminates when either the number of iterations has exceeded a preset amount or the simplex size is smaller than a given value.

3.2. The procedure of PSO

Particle swarm optimization (PSO) is one of the latest evolutionary optimization techniques developed by Kennedy and Eberhart (1995). PSO concept is based on a metaphor of social interaction such as bird flocking and fish schooling. Similar to genetic algorithms, PSO is also population-based and evolutionary in nature, with one major difference from genetic algorithms, which is that it does not implement filtering, i.e., all members in the population survive through the entire search process. PSO simulates a commonly observed social behavior, where members of a group tend to follow the lead of the best of the group. The steps of PSO are outlined below:

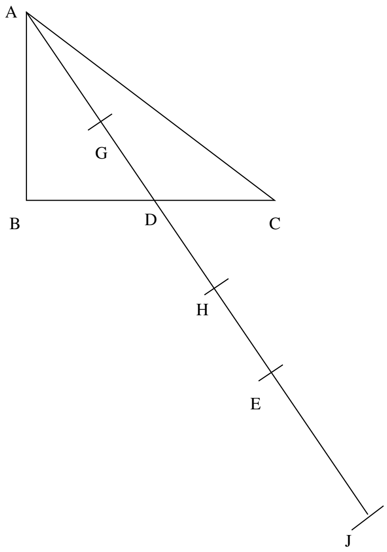


Fig. 1. NM operations on a two-dimensional case.

1. *Initialization*. Randomly generate $5N$ potential solutions, called "particles", N being the number of parameters to be optimized, and each particle is assigned a randomized velocity.
2. *Velocity update*. The particles then "fly" through hyperspace while updating their own velocity, which is accomplished by considering its own past flight and those of its companions'. The particle's velocity and position are dynamically updated by the following equations:

$$V_{\text{id}}^{\text{New}} = w \times V_{\text{id}}^{\text{old}} + c_1 \times \text{rand} \times (p_{\text{id}} - x_{\text{id}}^{\text{old}}) + c_2 \times \text{rand} \times (p_{\text{gd}} - x_{\text{id}}^{\text{old}}) \tag{3}$$

$$x_{\text{id}}^{\text{New}} = x_{\text{id}}^{\text{old}} + V_{\text{id}}^{\text{New}} \tag{4}$$

where c_1 and c_2 are two positive constants, w is an inertia weight, and rand is a uniformly generated random number. Eberhart and Shi (2001) and Hu and Eberhart (2001) suggested $c_1 = c_2 = 2$ and $w = [(0.5 + \text{rand}/2.0)]$. Eq. (3) shows that in calculating the new velocity for a particle, the previous velocity of the particle (V_{id}), the best location in the neighborhood about the particle (p_{id}), and the global best location (p_{gd}) all contribute some influence to the outcome of velocity update. Particles' velocities in each dimension are clamped to a maximum velocity $V_{\max}$, which is confined to the range of the search space in each dimension. Eq. (4) shows how each particle's position (x_{id}) is updated during the search in the solution space.

3.3. Hybrid NM–PSO

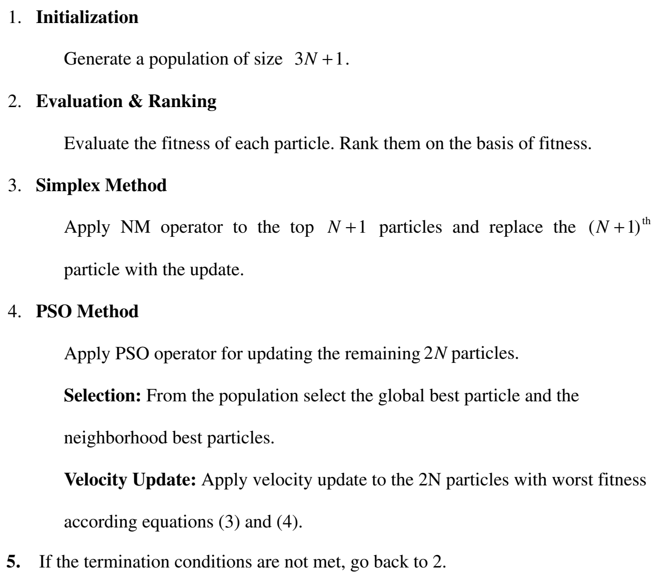
Having discussed NM and PSO separately, we will now look at their integrated form. The population size of this hybrid NM–PSO approach is set at $3N + 1$ when solving an N-dimensional problem. The initial $3N + 1$ particles are randomly generated and sorted by fitness, and the top $N + 1$ particles are then fed into the simplex search method to improve the $(N + 1)$th particle. The other $2N$ particles are adjusted by the PSO method by taking into account the positions of the $N + 1$ best particles. This step of adjusting the $2N$ particles involves selection of the global best particle, selection of the neighborhood best particles, and finally velocity updates. The global best particle of the population is determined according to the sorted fitness values. The neighborhood best particles are selected by first evenly dividing the $2N$ particles into N neighborhoods and designating the particle with the better fitness value in each neighborhood as the neighborhood best particle. By Eqs. (3) and (4), a velocity update for each of the $2N$ particles is then carried out. The $3N+1$ particles are sorted again in preparation for repeating the entire run. The process terminates when certain convergence criteria are met. Fig. 2 summarizes the hybrid NM–PSO algorithm. For more details, see Fan et al. (2004).

3.4. Hybrid K–NM–PSO

The K-means algorithm tends to converge faster than the PSO as it requires fewer function evaluations, but it usually results in less accurate clustering. One can take advantage of its speed at the inception of the clustering process and leave accuracy to be achieved by other methods at a later stage of the process. This statement shall be verified in later sections of this paper by showing that the results of clustering by PSO and NM–PSO can further be improved by seeding the initial population with the outcome of the K-means algorithm (denoted as K–PSO and K–NM–PSO). More specifically, the hybrid algorithm first executes the K-means algorithm, which terminates when there is no change in centroid vectors. In the case of K–PSO, the result of the K-means algorithm is used as one of the particles, while the remaining $5N$-1 particles are initialized randomly. The best PSO algorithm then proceeds as presented above. In the case of K–NM–PSO, randomly generate $3N$ particles, or vertices as termed in the earlier introduction of NM, and NM–PSO is then carried out to its completion.

4. Experimental results

The K-means clustering algorithm has been described in Section 2 and the objective function (1) of the algorithm will now be subjected to being minimized by PSO, NM–PSO, K–PSO and K–NM–PSO. Given a dataset with four features that is to be grouped into two clusters, for example, the number of parameters to be optimized is equal to the product of the number of clusters and the number of features, $N = k \times d = 2 \times 4 = 8$, in order to find the two optimal cluster centroid vectors.



```
1. Initialization
     Generate a population of size $3N+1$.
2. Evaluation & Ranking
     Evaluate the fitness of each particle. Rank them on the basis of fitness.
3. Simplex Method
     Apply NM operator to the top $N+1$ particles and replace the $(N+1)^{\text{th}}$
     particle with the update.
4. PSO Method
     Apply PSO operator for updating the remaining $2N$ particles.
     Selection: From the population select the global best particle and the
     neighborhood best particles.
     Velocity Update: Apply velocity update to the 2N particles with worst fitness
     according equations (3) and (4).
5. If the termination conditions are not met, go back to 2.
```

Fig. 2. The hybrid NM–PSO algorithm.

Table 1
Characteristics of the data sets considered

Name of data set	No. of classes	No. of features	Size of data set (size of classes in parentheses)
Art1	4	2	600 (150, 150, 150, 150)
Art2	5	3	250 (50, 50, 50, 50, 50)
Vowel	6	3	871 (72, 89, 172, 151, 207, 180)
Iris	3	4	150 (50, 50, 50)
Crude Oil	3	5	56 (7, 11, 38)
CMC	3	9	1473 (629, 334, 510)
Cancer	2	9	683 (444, 239)
Glass	6	9	214 (70, 17, 76, 13, 9, 29)
Wine	3	13	178 (59, 71, 48)

Nine data sets were employed to validate our method. These data sets, named Art1, Art2, Vowel, Iris, Crude oil, CMC, Cancer, Glass, and Wine, cover examples of data of low, medium and high dimensions. All data sets except Art1 and Art2 are available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/. Table 1 summarizes the characteristics of these data sets.

4.1. Data sets

(1) Artificial data set one ($n = 600$, $d = 2$, $k = 4$): This is a two-featured problem with four unique classes. A total of 600 patterns were drawn from four independent bivariate normal distributions, where classes were distributed according to $N_2\left(\mu = \begin{pmatrix} m_i \\ 0 \end{pmatrix}, \Sigma = \begin{bmatrix} 0.50 & 0.05 \\ 0.05 & 0.50 \end{bmatrix}\right)$, $i = 1, \ldots, 4$, $m_1 = -3$, $m_2 = 0$, $m_3 = 3$, $m_4 = 6$, μ being the mean vector and Σ being the covariance matrix. The data set is illustrated in Fig. 3a.

(2) Artificial data set two ($n = 250$, $d = 3$, $k = 5$): This is a three-featured problem with five classes, where every feature of the classes was distributed according to $Class1 \sim Uniform(85,100)$, $Class2 \sim Uniform(70,85)$, $Class3 \sim Uniform(55,70)$ $Class4 \sim Uniform(40,55)$, $Class5 \sim Uniform(25,40)$. The data set is illustrated in Fig. 3b.

(3) Vowel data set ($n = 871$, $d = 3$, $k = 6$), which consists of 871 Indian Telugu vowel sounds. The data set has three features corresponding to the first, second, and third vowel frequencies and six overlapping classes {δ (72 objects), a (89 objects),i (172 objects), u (151 objects), e (207 objects), o (180 objects)}.

(4) Fisher's iris data set ($n = 150$, $d = 4$, $k = 3$), which consists of three different species of iris flower: Iris setosa, Iris virginica, and Iris versicolour. For each

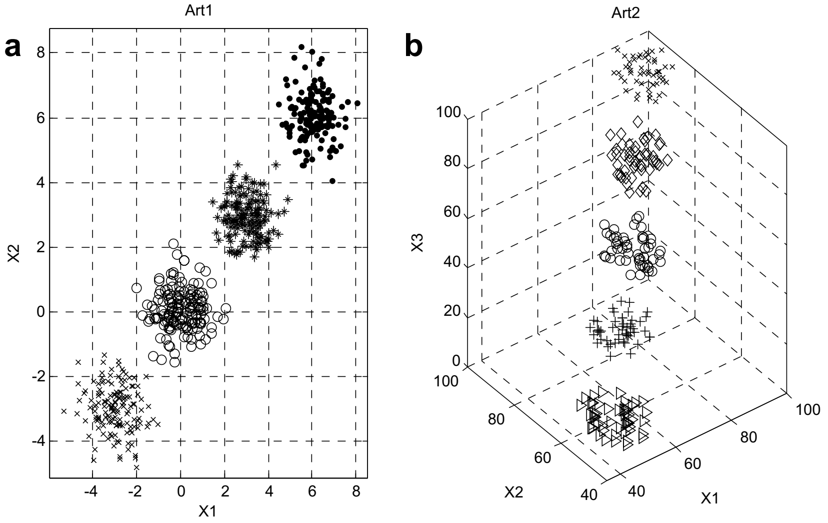


Fig. 3. Two artificial data sets.

species, 50 samples with four features each (sepal length, sepal width, petal length, and petal width) were collected.

(5) Crude oil data set ($n = 56$, $d = 5$, $k = 3$), which consists of 56 objects characterized by five features: vanadium, iron, beryllium, saturated hydrocarbons, and aromatic hydrocarbons. There are three crude-oil samples from three zones of sandstone (Wilhelm have 7 objects, Sub-Mulnia have 11 objects, and Upper have 38 objects).

(6) Contraceptive Method Choice (denoted as CMC with $n = 1473$, $d = 9$, $k = 3$): This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who either were not pregnant or did not know if they were at the time of interview. The problem is to predict the choice of current contraceptive method (no use has 629 objects, long-term methods have 334 objects, and short-term methods have 510 objects) of a woman based on her demographic and socioeconomic characteristics.

(7) Wisconsin breast cancer ($n = 683$, $d = 9$, $k = 2$), which consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects).

(8) Ripley's glass ($n = 214$, $d = 9$, $k = 6$), for which data were sampled from six different types of glass: building windows float processed (70 objects), building windows non-float processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects), and headlamps (29 objects), each with nine features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron.

(9) Wine ($n = 178$, $d = 13$, $k = 3$): These data, consisting of 178 objects characterized by 13 such features as alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines, and praline, are the results of a chemical analysis of wines brewed in the same region in Italy but derived from three different cultivars. The quantities of objects in the three categories of the data are: class 1 (59 objects), class 2 (71 objects), and class 3 (48 objects).

4.2. Results

In this section, we evaluate and compare the performances of the following methods: K-means, PSO, NM–PSO, K–PSO and K–NM–PSO algorithms as means of solution for the objective function of the K-means algorithm. The quality of the respective clustering will also be compared, where quality is measured by the following two criteria:

- the sum of the intra-cluster distances, i.e. the distances between data vectors within a cluster and the centroid of the cluster, as defined in Eq. (1). Clearly, the smaller the sum of the distances is, the higher the quality of clustering.
- error rate (ER): It is the number of misplaced points divided by the total number of points, as shown in Eq. (5):

$$\mathrm{ER} = \left(\sum_{i=1}^{n} (\text{if } (A_i = B_i) \text{ then } 0 \text{ else } 1) \div n \right) \times 100 \qquad (5)$$

where n denotes the total number of points, and A_i and B_i denote the data sets of which the ith point is a member before and after clustering, respectively. Table 2 serves as an example, in which, out of three clusters 1–3, two data points (2, 2) and (4, 10) are misplaced and the error rate is 2/6, which is 33.3%.

The reported results are averages of 20 runs of simulation as given below. The algorithms are implemented using Matlab on a Celeron 2.80 GHz with 504 MB RAM. For each run, $10 \times N$ iterations are carried out on each of the seven datasets for every algorithm when solving an N-dimensional problem. The criterion $10 \times N$ is adopted as it has been used in many previous experiments with great success in terms of both efficiency and effectiveness.

Table 3 summarizes the intra-cluster distances obtained from the five clustering algorithms for the data sets above. The values reported are averages of the sums of intra-cluster distances over 20 simulations, with standard deviations in parentheses to indicate the range of values that the algorithms span and the best solution of fitness from the 20 simulations. For Art1, the averages of the fitness for NM–PSO, K–PSO, and K–NM–PSO are almost identical to the best distance, and the standard deviations of the fitness for these three algorithms are less than 5.6E−05, significantly smaller than those of the other two methods, which is an indication that NM–PSO, K–PSO, and K–NM–PSO converge to the global optimum 515.8834 every time, while K-means and PSO may be trapped at local optimum solutions. For Art2, the average of the fitness for K–NM–PSO is near to the best distance, and the standard deviation of the fitness for this algorithm is 3.60, much smaller than those of the other four methods. For the other real life data sets, K–NM–PSO also outperforms the other four methods, as born out by a smaller difference between the average and the best solution and a small standard deviation. Please note that in terms of the best distance, although PSO, NM–PSO, and K–PSO may achieve the global optimum, they all have a larger standard deviation than does K–NM–PSO, meaning that PSO, NM–PSO, and K–PSO are less likely to reach the global optimum than K–NM–PSO if they all execute just once. It follows that K–NM–PSO is both effective and efficient for finding the global optimum solution as compared with the other four methods.

Table 2
Error rate calculations

i	Data point	A_i	B_i	Not misplaced (0)/Misplaced (1)
1	(3, 3)	1	1	0
2	(2, 2)	1	3	1
3	(10, 3)	2	2	0
4	(11, 3)	2	2	0
5	(4, 10)	3	1	1
6	(3, 11)	3	3	0
The number of misplaced point are:				2

Table 3
Comparison of intra-cluster distances for the five clustering algorithms

Data set	Criteria	K-means	PSO	NM–PSO	K–PSO	K–NM–PSO
Art1	Average	721.57	627.74	*515.88*	*515.88*	*515.88*
	(Std)	(295.84)	(180.24)	(*7.14E−08*)	(*5.60E−05*)	(*7.14E−08*)
	Best	516.04	515.93	*515.88*	*515.88*	*515.88*
Art2	Average	2762.00	2517.20	1910.40	2067.30	*1746.90*
	(Std)	(720.66)	(415.02)	(296.22)	(343.64)	(*3.60*)
	Best	1746.9	1743.20	1743.20	1743.20	*1743.20*
Vowel	Average	159242.87	168477.00	151983.91	149375.70	*149141.40*
	(Std)	(916)	(3715.73)	(4386.43)	(155.56)	(*120.38*)
	Best	149422.26	163882.00	149240.02	149206.10	*149005.00*
Iris	Average	106.05	103.51	100.72	96.76	*96.67*
	(Std)	(14.11)	(9.69)	(5.82)	(0.07)	(*0.008*)
	Best	97.33	96.66	96.66	96.66	*96.66*
Crude oil	Average	287.36	285.51	277.59	277.77	*277.29*
	(Std)	(25.41)	(10.31)	(0.37)	(0.33)	(*0.095*)
	Best	279.20	279.07	277.19	277.45	*277.15*
CMC	Average	5693.60	5734.20	5563.40	5532.90	*5532.70*
	(Std)	(473.14)	(289.00)	(30.27)	(0.09)	(*0.23*)
	Best	5542.20	5538.50	5537.30	5532.88	*5532.40*
Cancer	Average	2988.30	3334.60	2977.70	2965.80	*2964.70*
	(Std)	(0.46)	(357.66)	(13.73)	(1.63)	(*0.15*)
	Best	2987	2976.30	2965.59	2964.50	*2964.50*
Glass	Average	260.40	291.33	248.96	207.35	*200.50*
	(Std)	(36.82)	(12.33)	(6.82)	(5.12)	(*2.26*)
	Best	215.68	271.29	243.45	203.37	*199.68*
Wine	Average	18061.00	16311.00	16303.00	16294.00	*16293.00*
	(Std)	(793.21)	(22.98)	(4.28)	(1.70)	(*0.46*)
	Best	16555.68	16294.00	16292.00	16292.00	*16292.00*

The mean error rates, standard deviations, and the best solution of the error rates from the 20 simulations are shown in Table 4. For Art1, the mean, the standard deviation, and the best solution of the error rates are all 0% for NM–PSO, K–PSO, and K–NM–PSO, signifying that these methods classify this data set correctly. For Art2, K–NM–PSO correctly accomplishes the task, too. For the real life data sets, K–NM–PSO exhibits a significantly smaller mean and standard deviation compared with K-means, PSO, NM–PSO, and K–PSO. Again, K–NM–PSO is superior to the other four methods with respect to the intra-cluster distance. However, it does not compare favorably with NM–PSO and PSO for Vowel, Iris, CMC, and Glass data sets in terms of the best error rate, as there is no absolute correlation between the intra-cluster distance and the error rate. The fundamental mechanism of K-means algorithm has difficulty detecting the "natural clusters", that is, clusters with non-spherical shapes or widely different sizes or densities, and subsequent NM–PSO operations cannot be expected to gain much in accuracy following a somehow erroneous pre-clustering.

Table 5 lists the numbers of evaluating objective function (1) required of the five methods after $10 \times N$ iterations. For all the data sets, K-means needs the least number of function evaluations, but the results are less than satisfactory, seen in Tables 3 and 4, as it tends to be trapped at local optimum. K–NM–PSO uses less function evaluations than PSO, NM–PSO, and K–PSO and produces better outcomes than they do. All the evidence of the simulations demonstrates that K–NM–PSO converges to global optima with a smaller error rate and less function evaluations and leads naturally to the conclusion that K–NM–PSO is a viable and robust technique for data clustering.

Table 5
The number of function evaluations of each clustering algorithm

Data Set	K-means	PSO	NM–PSO	K–PSO	K–NM–PSO
Art1	80	3240	2265	2976	1996
Art2	150	11,325	7392	10,881	7051
Vowel	180	16,290	10,501	15,133	9291
Iris	120	7260	4836	6906	4556
Crude oil	150	11,325	7394	10,807	7057
CMC	270	36,585	23,027	34,843	21,597
Cancer	180	16,290	10,485	15,756	10,149
Glass	630	198,765	121,773	195,625	119,825
Wine	390	73,245	47,309	74,305	46,459

Figs. 4 and 5 provide more insight into the convergence behaviors of these five algorithms. Fig. 4a illustrates the trends of convergence of the algorithms for Art1. The K-means algorithm exhibits a fast but premature convergence to a local optimum. PSO converges near to the global

Table 4
Comparison of error rates for the five clustering algorithms

Data Set	Criteria	K-means (%)	PSO (%)	NM–PSO (%)	K–PSO (%)	K–NM–PSO (%)
Art1	Average	13.00	7.57	*0.00*	*0.00*	*0.00*
	(Std)	(17.78)	(12.18)	(*0.00*)	(*0.00*)	(*0.00*)
	Best	0.00	0.00	*0.00*	*0.00*	*0.00*
Art2	Average	34.00	22.00	4.04	10.00	*0.00*
	(Std)	(13.45)	(11.35)	(8.52)	(10.32)	(*0.00*)
	Best	20.00	0.00	0.00	0.00	*0.00*
Vowel	Average	44.26	44.65	41.96	42.24	*41.94*
	(Std)	(2.15)	(2.55)	(0.98)	(0.95)	(*0.95*)
	Best	42.02	41.45	*40.07*	40.64	40.64
Iris	Average	17.80	12.53	11.13	10.20	*10.07*
	(Std)	(10.72)	(5.38)	(3.02)	(0.32)	(*0.21*)
	Best	10.67	10.00	*8.00*	10.00	10.00
Crude oil	Average	24.46	24.64	24.29	24.29	*23.93*
	(Std)	(1.21)	(1.73)	(0.75)	(0.92)	(*0.72*)
	Best	23.21	23.21	23.21	23.21	*23.21*
CMC	Average	54.49	54.41	54.47	54.38	*54.38*
	(Std)	(0.04)	(0.13)	(0.06)	(0.00)	(*0.054*)
	Best	54.45	*54.24*	54.38	54.38	54.31
Cancer	Average	4.08	5.11	4.28	*3.66*	*3.66*
	(Std)	(0.46)	(1.32)	(1.10)	(*0.00*)	(*0.00*)
	Best	3.95	3.66	3.66	*3.66*	*3.66*
Glass	Average	37.71	45.59	40.89	32.17	*30.45*
	(Std)	(13.75)	(15.62)	(15.58)	(15.52)	(14.13)
	Best	11.00	11.21	*9.35*	12.62	12.62
Wine	Average	31.12	28.71	28.48	28.48	*28.37*
	(Std)	(0.71)	(0.41)	(0.27)	(0.40)	(*0.27*)
	Best	29.78	28.09	28.09	28.09	*28.09*

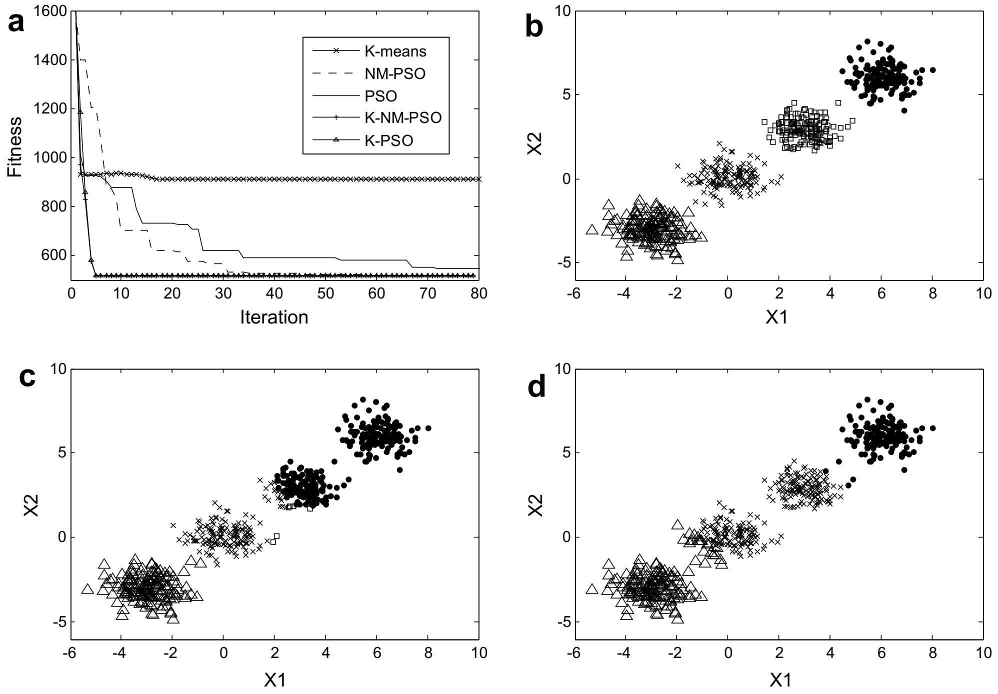


Fig. 4. Art1 data set: (a) algorithm convergence; (b) NM–PSO, K–PSO and K–NM–PSO result with 0% error rate; (c) PSO-Cluster result with 25% error rate; (d) K-means algorithm result with 25.67% error rate.

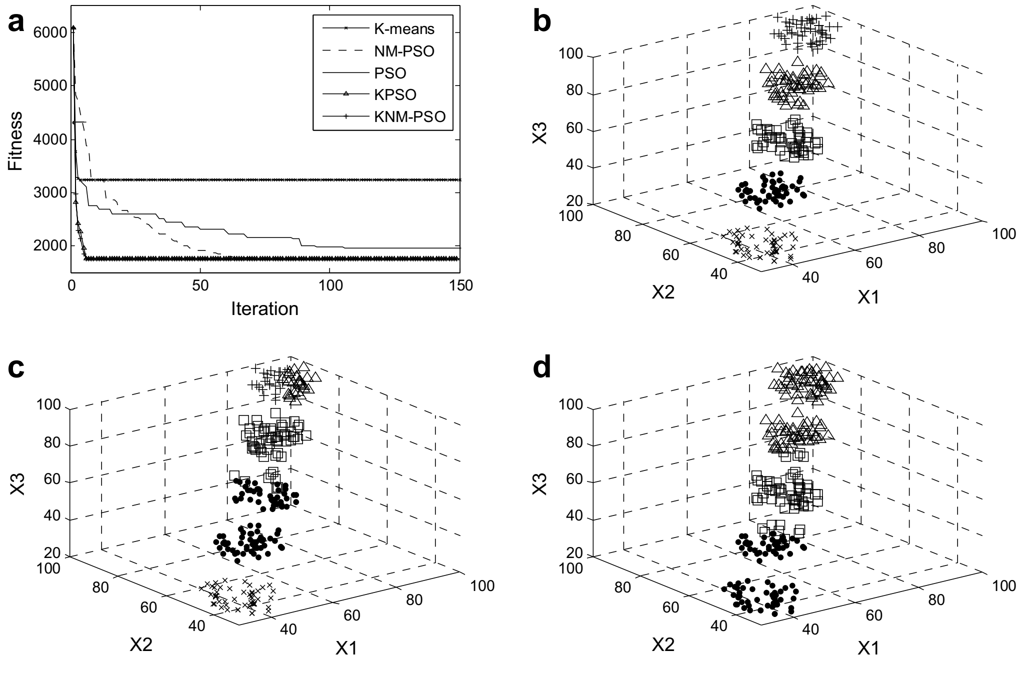


Fig. 5. Art2 data set: (a) algorithm convergence; (b) NM–PSO, K–PSO and K–NM–PSO result with 0% error rate; (c) PSO-Cluster result with 20% error rate; (d) K-means algorithm result with 40% error rate.

optimum and NM–PSO in about 50 iterations converges to the global optimum, whereas K–PSO and K–NM–PSO in about 10 iterations converge to the global optimum. Fig. 4b shows the clustering results for NM–PSO, K–PSO and K–NM–PSO, which correctly classify this data set into four clusters. Fig. 4c and d illustrates the final clusters for PSO and K-means, respectively. PSO classifies this data set with a 25% error rate and K-means algorithm classifies this data set into three clusters with a 25.67% error rate.

Table 6
Comparison of the intra-cluster distances for GA, KGA and K–NM–PSO

Data Set	Criteria	K–NM–PSO	GA	KGA
Vowel	Average	*149141.40*	390088.24	149368.45
	Best	*149005.00*	383484.15	149356.01
Iris	Average	*96.67*	135.40	97.10
	Best	*96.66*	124.13	97.10
Crude oil	Average	*277.29*	308.16	278.97
	Best	*277.15*	297.05	278.97

As in Figs. 4a, 5a depicts the trends of convergence of the algorithms for Art2. The K-means algorithm exhibits a fast but premature convergence to a local optimum. PSO converges to a better optimal solution but does not reach the global optimum, whereas NM–PSO in about 80 iterations converges to the global optimum. Fig. 5b shows the clustering results for NM–PSO, K–PSO and K–NM–PSO, which correctly classify this data set into five clusters. Fig. 5c and d illustrates the resulting clusters for PSO and K-means, respectively. PSO classifies the same data set into five clusters with a 20% error rate and K-means algorithm classifies this data set into three clusters with a 40% error rate.

Genetic algorithms have also been adopted to cluster data. Table 6 provides the performances of a genetic algorithm (Murthy & Chowdhury, 1996) and a hybrid genetic algorithm, KGA for hybrid K-means and genetic algorithm (Bandyopadhyay & Maulik, 2002) for data clustering in terms of the best and average intra-cluster distances. As seen from the table, K–NM–PSO has a smaller average and best intra-cluster distance for these three data sets compared with GA and KGA. The superiority of K–NM–PSO is again evident is these figures.

5. Conclusions

This paper investigates the application of the hybrid K–NM–PSO algorithm to clustering data vectors using nine data sets. K–NM–PSO, using the minimum intra-cluster distances as a metric, searches robustly the data cluster centers in an N-dimensional Euclidean space. Using the same metric, PSO, NM–PSO, and K–PSO are shown to need more iteration to achieve the global optimum, while the K-means algorithm may get stuck at a local optimum, depending on the choice of the initial cluster centers. The experimental results indicate, too, that K–NM–PSO is at least comparable to the other four algorithms in terms of the error rate.

Despite its robustness and efficiency, the K–NM–PSO algorithm developed in this paper is not applicable when the number of clusters is not known a priori, a topic that merits further research. Also, the algorithm needs to be modified in order to take care of situations where the partitioning is fuzzy.

References

Bandyopadhyay, S., & Maulik, U. (2002). An evolutionary technique based on K-means algorithm for optimal clustering in R^N. *Information Science, 146*, 221–237.

Chen, C.-Y., & Ye, F. (2004). Particle swarm optimization algorithm and its application to clustering analysis. In *Proceedings of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, Taiwan* (pp. 789–794).

Eberhart, R. C., & Shi, Y. (2001). Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the Congress on Evolutionary Computation, Seoul, Korea* (pp. 94-97).

Fan, S.-K. S., Liang, Y.-C., & Zahara, E. (2004). Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. *Engineering Optimization, 36*, 401–418.

Hooke, R., & Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of Association for Computing Machinery, 8*, 212–221.

Hu, X., & Eberhart, R. C. (2001). Tracking dynamic systems with PSO: where's the cheese? In *Proceedings of the Workshop on Particle Swarm Optimization, Indianapolis, IN, USA*.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. New York: John Wiley & Sons.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of the IEEE International Joint Conference on Neural Network, 4*, 1942–1948.

Murthy, C. A., & Chowdhury, N. (1996). In search of optimal cluaters using genetic algorithms. *Pattern Recognition Letters, 17*, 825–832.

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal, 7*, 308–313.

Olsson, D. M., & Nelson, L. S. (1975). The Nelder–Mead simplex procedure for function minimization. *Technometrics, 17*, 45–51.

Paterlini, S., & Krink, T. (2006). Differential evolution and particle swarm optimization in partitional clustering. *Computational Statistics and Data Analysis, 50*, 1220–1247.

Renders, J. M., & Flasse, S. P. (1996). Hybrid methods using genetic algorithms for global optimization. *IEEE Transaction on System Man and Cybernetic. Part B: Cybernetics, 26*, 243–258.

Selim, S. Z., & Ismail, M. A. (1984). K-means type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transaction of Pattern Analysis Machine Intelligent, 6*, 81–87.

Spendley, W., Hext, G. R., & Himsworth, F. R. (1962). Sequential application of simplex designs in optimization and evolutionary operation. *Technometrics, 4*, 441–461.

Yen, J., Liao, J. C., Lee, B., & Randolph, D. (1998). A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. *IEEE Transaction on System Man and Cybernetic Part B: Cybernetics, 28*, 173–191.