2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Accelerated Chaotic Particle Swarm Optimization for Data Clustering

Cheng-Hong Yang [1 +], Chih-Jen Hsiao[1] and Li-Yeh Chuang [2], Member, IEEE

[1] Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan

[2] Department of Chemical Engineering, I-Shou University, Kaohsiung, Taiwan

Abstract. Data clustering is a powerful technique for discerning the structure of and simplifying the complexity of large scale data. An improved technique combining chaotic map particle swarm optimization (CPSO) with an acceleration strategy, is proposed in this paper. Accelerated chaotic particle swarm optimization (ACPSO) searches for cluster centers of an arbitrary data set and can effectively find the global optima. ACPSO is tested on six experimental data sets, and its performance is compared to the performance of PSO, NM-PSO, K-PSO, K-NM-PSO and K-means clustering. Results indicated that ACPSO is both robust and suitable for solving data clustering problem.

Keywords: Data Clustering, Chaotic Map, Particle Swarm Optimization.

1. Introduction

Clustering analysis is a very popular data mining techniques. It is the process of grouping a set of objects into clusters so that objects within a cluster are similar to each other but are dissimilar to objects in other clusters [1]. When used on a set of objects, it helps identify some inherent structures present in the objects. The purpose of cluster analysis is to classify the clusters into subsets that have some meaning in the context of a particular problem. More specifically, a set of patterns, usually 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 the distribution of *n* objects in an *N*-dimensional space among *K* groups, such that the objects in the same group are more similar in some sense than those in different groups [2]. This involves minimization of some extrinsic optimization criteria. In recent years, many clustering algorithms based on evolutionary computing techniques such as particle swarm optimization have been introduced [3]. Particle swarm optimization (PSO) is a population-based algorithm [4]. It simulates bird flocking or fish schooling behavior to achieve a self-evolving system. It searches automatically for the optimum solution in the search space, and the involucel search process is not random. Depending on the nature of different problems, a fitness function decides the best way to conduct the 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.

PSO applied to the clustering of multi-dimensional space has shown outstanding performance. However, the rate of convergence when searching for global optima is still not sufficient [5]. For this reason, we combine chaotic map particle swarm optimization (CPSO) with an accelerated convergence rate strategy, and introduce this accelerated chaotic map particle swarm optimization (ACPSO) in this research. The choice of chaotic sequences is justified theoretically by their unpredictability, i.e., by the spread-spectrum characteristics and ergodic properties of these sequences [6]. We used this characteristica on chaotic map and adaptive action to avoid the entrapment of PSO in a local optimum [7]. This paper proposes that the ACPSO algorithm can be adapted to cluster arbitrary data by evolving the appropriate cluster centers in an attempt to

+ Corresponding author. Tel.: + 886 7 381 4526 ext 5639; fax: +886 7 383 6844.
E-mail address: chyang@cc.kuas.edu.tw

optimize a given clustering metric. Results of the conducted experimental studies on a variety of data sets provided from several real-life situations demonstrate that ACPSO is superior to the K-means, PSO, K-PSO, and K-NM-PSO algorithms [5].

2. Methods

Although PSO has been successfully applied to many practical clustering problems, its convergence rate is still rather slow and the global search ability for optimum solutions needs to be improved. We propose a combination of chaotic map and an acceleration strategy to improve the performance of PSO. ACPSO consists of four major steps, namely the encoding and initialization of the particle, the acceleration strategy, the velocity and position update, and the fitness evaluation. The ACPSO procedure for data clustering is described below:

1) Initial population and encoding**:** Randomly generate 3*N* particles, where each particle represents a feasible solution (cluster center) of the problem. *N* is computed as follows:

$$N = K \times d \tag{1}$$

where *d* is the data set dimension and *K* is the anticipated number of clusters.

A possible encoding of a particle for a two-dimensional problem with four clusters is illustrated in Fig. 1. The four cluster centers represented by this particle are (-3, 1.2), (8, 4.6), (1.1, 3.6), and (-7, -4).

2) Acceleration strategy: In an initial steps, one-third of the particles is used to accelerate the convergence rate of the particles. The distances between data vectors within a cluster and the center of the cluster are defined in Eq. (2). The acceleration strategy recalculates the cluster center vectors using Eq. (3), and yields mean centers. The mean clusters then replace the original centers. This is the new position of the particle.

$$D(x_p \cdot z_j) = \sqrt{\sum_{i=1}^{d} (x_{pi} - z_{ji})^2} \tag{2}$$

$$z_j = \frac{1}{n_j} \sum_{\forall x_p \in c_j} x_p \tag{3}$$

where z_j denotes the center vector of cluster *j*, x_p denotes the p^{th} data vector, *d* subscripts the number of features of each center vector, n_j is the number of data vectors in cluster *j* and C_j is the subset of data vectors that form cluster *j*.

3) Velocity and position update: The particles are moving through the search space in each iteration. In chaotic particle swarm optimization, sequences generated by the logistic map [8] substitute the random parameters r_1 and r_2 of PSO. The parameters r_1 and r_2 are modified by the logistic map based on the following equation.

$$Cr_{n+1} = 4.0 \times Cr_n \times (1 - Cr_n) \tag{4}$$

In Eq. (4), $Cr_{(0)}$ is generated randomly for each independent run, with $Cr_{(0)}$ not being equal to {0, 0.25, 0.5, 0.75, 1}. The chaotic sequence value *Cr* is bounded within (0, 1), and $Cr_{(0)}$ = 0.001. The particle's velocity and position are dynamically updated as follows:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times Cr \times (pbest_{id} - x_{id}^{old}) + c_2 \times (1 - Cr) \times (gbest_d - x_{id}^{old}) \tag{5}$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \tag{6}$$

where r_1 and r_2 are random numbers between (0, 1), w is an inertia weight, and c_1 and c_2 are two positive

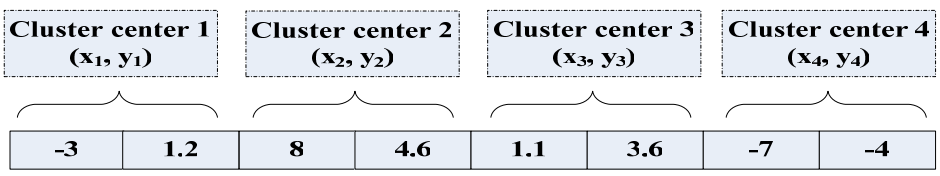

Fig. 1: Encoding of a single particle in PSO

constants. The best previously encountered position of the ith particle is denoted its individual best position pi = (pi1, pi2, …, piD), a value called pbesti. The best value of the all individual pbesti values is denoted the global best position g = (g1, g2, …, gD) and called gbest. Eberhart et al. [9, 10] suggested c1 = c2 = 2 and w = [(0.5 + rand/2.0)]. Velocities v_{id}^{new} and v_{id}^{old} denote the velocities of the new and old particle, respectively. x_{id}^{old} is the current particle position, and x_{id}^{new} is the new, updated particle position.

4) Fitness evaluation: The fitness value of each particle can be computed by following the fitness function.

$$\text{fitness} = \sum \left\| X_j - Z_i \right\|, \ i = 1, ..., K, \ j = 1, ..., n \tag{7}$$

where K and n are the numbers of clusters and data sets, respectively. Zi is cluster center i and Xj is data point j. The pbesti and gbest values are updated if the new value is better than the old one. Step 3) is repeated until the termination condition is met. The pseudo-code of ACPSO is shown below:

***ACPSO* pseudo-code**

```
01: begin
02:   Randomly initialize particles swarm
03:   Accelerate strategy of one-third of particle on initial swarm
04:   Randomly generate Cr(0)
05:   while (number of iterations, or the stopping criterion is not met)
06:     Evaluate fitness of particle swarm
07:     for n = 1 to number of particles
08:       Find pbest
09:       Find gbest
10:       for d = 1 to number of dimension of particle
11:         Update the Chaotic Cr value by Eq. 3
12:         Update the position of particles by Eq. 4 and Eq. 5
13:       next d
14:     next n
15:   next generation until stopping criterion
16: end
```

3. Experimental results and discussions

Six experimental data sets were employed to validate our method. These data sets, named Vowel, Iris, Crude oil, CMC, Cancer, and Wine, cover examples of data of low, medium and high dimensions. All data sets are available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/. Table 1 summarizes the characteristics of these data sets. Given a data set with three features that is to be grouped into two clusters, the number of parameters to be optimized in order to find the two optimal cluster centres vectors is equal to the product of the number of clusters and the number of features, N = k × d = 2 × 3 = 6.

In order to demonstrate the power of ACPSO, we compared the results to results obtained with the following methods: K-means, PSO, NM-PSO, K-PSO, K-NM-PSO and ACPSO. The quality of the respective clustering was also compared, where quality is measured by the following two criteria:

1. 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. (2). A higher quality of clustering is achieved if the sum is relatively small.

2. Error rate: the number of misplaced points divided by the total number of points, as shown in Eq.(8):

$$\text{error rate} = \left(\sum_{i=1}^{n} (if \quad (A_i = B_i) \quad then \quad 0 \quad else \quad 1) \div n \right) \times 100 \tag{8}$$

where n denotes the total number of points. Ai and Bi denote the data sets of which the ith point is a member before and after clustering, respectively.

The reported results are averages of 20 simulation runs details of which are given below. The algorithms were implemented using Java. For each run, 10 × N iterations were carried out on each of the six data sets for every

Table 1: Characteristics of the considered data sets

Name of data set	Number of classes	Number of features	Size of data set (size of classes in parentheses)
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)
Wine	3	13	178 (59, 71, 48)

Table 2: Comparison of intra-cluster distances for the six clustering algorithms

Data set	Criteria	K-means[5]	PSO[5]	NM-PSO[5]	K-PSO[5]	K-NM-PSO[5]	ACPSO
Vowel	Average	159242.87	168477.00	151983.91	149375.70	149141.40	**149051.84**
	(Std)	(916)	(3715.73)	(4386.43)	(155.56)	(120.38)	(67.27)
	Best	149422.26	163882.00	149240.02	149206.10	149005.00	148970.84
Iris	Average	106.05	103.51	100.72	96.76	96.67	**96.66**
	(Std)	(14.11)	(9.69)	(5.82)	(0.07)	(0.008)	(0.001)
	Best	97.33	96.66	96.66	96.66	96.66	96.66
Crude Oil	Average	287.36	285.51	277.59	277.77	277.29	**277.24**
	(Std)	(25.41)	(10.31)	(0.37)	(0.33)	(0.095)	(0.04)
	Best	279.20	279.07	277.19	277.45	277.15	277.21
CMC	Average	5693.60	5734.20	5563.40	5532.90	5532.70	**5532.20**
	(Std)	(473.14)	(289.00)	(30.27)	(0.09)	(0.23)	(0.01)
	Best	5542.20	5538.50	5537.30	5532.88	5532.40	5532.19
Cancer	Average	2988.30	3334.60	2977.70	2965.80	2964.70	**2964.42**
	(Std)	(0.46)	(357.66)	(13.73)	(1.63)	(0.15)	(0.03)
	Best	2987	2976.30	2965.59	2964.50	2964.50	2964.39
Wine	Average	18061.00	16311.00	16303.00	16294.00	16293.00	**16292.31**
	(Std)	(793.21)	(22.98)	(4.28)	(1.70)	(0.46)	(0.03)
	Best	16555.68	16294.00	16292.00	16292.00	16292.00	16292.18

Table 3: Comparison of error rates for the six clustering algorithms

Data set	Criteria	K-means (%) [5]	PSO (%) [5]	NM-PSO (%) [5]	K-PSO (%) [5]	K-NM-PSO (%) [5]	ACPSO (%**)**
Vowel	Average	44.26	44.65	41.96	42.24	41.94	**41.69**
	(Std)	(2.15)	(2.55)	(0.98)	(0.95)	(0.95)	(0.31)
	Best	42.02	41.45	40.07	40.64	40.64	41.10
Iris	Average	17.80	12.53	11.13	10.20	10.07	**10.00**
	(Std)	(10.72)	(5.38)	(3.02)	(0.32)	(0.21)	(0.00)
	Best	10.67	10.00	8.00	10.00	10.00	10.00
Crude Oil	Average	24.46	24.64	24.29	24.29	**23.93**	26.25
	(Std)	(1.21)	(1.73)	(0.75)	(0.92)	(0.72)	(0.84)
	Best	23.21	23.21	23.21	23.21	23.21	25.00
CMC	Average	54.49	54.41	54.47	54.38	54.38	**54.38**
	(Std)	(0.04)	(0.13)	(0.06)	(0.00)	(0.054)	(0.00)
	Best	54.45	54.24	54.38	54.38	54.31	54.38
Cancer	Average	4.08	5.11	4.28	3.66	3.66	**3.51**
	(Std)	(0.46)	(1.32)	(1.10)	(0.00)	(0.00)	(0.00)
	Best	3.95	3.66	3.66	3.66	3.66	3.51
Wine	Average	31.12	28.71	28.48	28.48	28.37	**28.23**
	(Std)	(0.71)	(0.27)	(0.27)	(0.40)	(0.27)	(0.25)
	Best	29.78	28.09	28.09	28.09	28.09	28.09

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 effectiveness [5].

Table 2 summarizes the intra-cluster distances obtained from the six 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 the all experimental data sets, ACPSO outperformed the other five methods, as born out by a smaller difference between the averages and a small standard deviation. Please note that in terms of the best distance, PSO, NM-PSO, K-PSO, and K-NM-PSO all have a larger standard deviation than does ACPSO even though the may achieve a global optimum. This means that PSO, NM-PSO, K-PSO, and K-NM-PSO are weaker search tools for global optima than ACPSO if all executed just once. It follows that ACPSO is both more effective in efficient for finding the global optimum solution than the other five methods.

Table 3 shows the mean error rates, standard deviations, and the best solution of the error rates from the 20 simulations. For all the real life data sets except Crude Oil, ACPSO exhibited a significantly smaller mean and standard deviation compared to K-means, PSO, NM-PSO, K-PSO, and K-NM-PSO. Again, ACPSO is

superior to the other five methods with respect to the intra-cluster distance. However, it does not compare favorably with the other methods for Vowel, Iris, Crude Oil, and CMC data sets in terms of the best error rate, as there is no absolute correlation between the intra-cluster distance and the error rate [5].

The population size on PSO and K-PSO was 5N, NM-PSO and K-NM-PSO was 3N+1, ACPSO was 3N. The population size of ACPSO was smaller than the population of the other algorithms. This results in the lower computational cost of ACPSO. K-NM-PSO is a hybrid technique that combines the K-means algorithm, Nelder-Mead simplex search [11], and PSO. In a direct comparison the performance of ACPSO proved to be better than the performance of K-NM-PSO.

4. Conclusions

In this paper, we employed the ACPSO algorithm to clustering data vectors for six data sets. ACPSO uses minimum intra-cluster distances as a metric, searches the robust data cluster centers in an N-dimensional Euclidean space. Under the same metric, PSO, NM-PSO, K-PSO, and K-NM-PSO need more iterations to achieve a global optimum. The experimental results indicate that ACPSO reached the minimal error rate faster than the other methods, and thus reduces computational cost. In the future, we will employ ACPSO to other clustering problem in bioinformatics. We intend to develop a hybrid technique based on other clustering algorithms to enhance the performance of ACPSO.

5. References

[1] J. Han, M. Kamber, and A. K. H. Tung. Spatial clustering methods in data mining: A survey. In H. J. Miller & J. Han (Eds.). Geographic data mining and knowledge discover, *London: Taylor & Francis*, 2001.

[2] M. R. Anderberg. Cluster Analysis for Application. *Academic Press*, New York, 1973.

[3] S. Paterlini, and T. Krink. Differential evolution and particle swarm optimization in partitional clustering. *Computational Statistics and Data Analysis*. 2006, 50: 1220-1247.

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

[5] Y. T. Kao, E. Zahara, and I. W. Kao. A hybridized approach to data clustering. *Expert Systems with Applications.* 2008, 34: 1754-1762.

[6] B. Alatas, E. Akin, and A. B. Ozer. Chaos embedded particle swarm optimization algorithms. *Chaos,Solitons & Fractals.* 2008.

[7] L. Wang, D. Z. Zheng, and Q. S. Lin. Survey on chaotic optimization methods. Comput Technol Automat. 2001.

[8] R. M. May. Simple mathematical models with very complicated dynamics. *Nature.* 1967, 261: 459-467.

[9] R. C. Eberhart, and Y. Shi. Tracking and optimizing dynamic systems with particle swarms. *In Proceedings of the Congress on Evolutionary Computation.* 2001, pp. 94-97.

[10] X. Hu, and R. C. Eberhart. Tracking dynamic systems with PSO: where's the cheese?. *In Proceedings of the Workshop on Particle Swarm Optimization*, 2001.

[11] J. A. Nelder, and R. Mead. A simplex method for function minimization. *Computer Journal.* 1965, 7: 308-313.