International Journal of Academic Research in Business and Social Sciences
May 2013, Vol. 3, No. 5
ISSN: 2222-6990

Solving Data Clustering with the Hybrid PSO

Li Qi
Geography Department, Dezhou University, Dezhou, Shandong 253023, China

Abstract

Data clustering is a popular approach for automatically finding classes, concepts, or groups of patterns. The term “clustering” is used in several research communities to describe methods for grouping of unlabeled data. These communities have different terminologies and assumptions for the components of the clustering process and the context in which clustering is used. This paper looks into the use of Particle Swarm Optimization (PSO) for cluster analysis. In standard PSO the non-oscillatory route can quickly cause a particle to stagnate and also it may prematurely converge on suboptimal solutions that are not even guaranteed to local optimal solution. In this paper a modification strategy is proposed for the particle swarm optimization (PSO) algorithm and applied in the data sets. This paper provides a method for particles to steer clear off from local stagnation and the local search is applied to improve the goodness of fitting. The effectiveness of this concept is demonstrated by cluster analysis. Results show that the model provides enhanced performance and maintains more diversity in the swarm and thereby allows the particles to be robust to trace the changing environment.

Keywords*:* Particle Swarm Optimization (PSO), Roulette-Wheel selection, K-Means, Local Search

1. Introduction

The notion of a "cluster" cannot be precisely defined (Estivill-Castro, V, 2002) which is one of the reasons why there are so many clustering algorithms. There of course is a common denominator: a group of data objects. However, different researchers employ different cluster models, and for each of these cluster models again different algorithms can be given. The notion of a cluster, as found by different algorithms, varies significantly in its properties.

Clustering algorithms can be categorized as either hierarchical or optimization. Hierarchical clustering techniques proceed by either a series of successive merges or a series of successive divisions. The result is the construction of a tree like structure or hierarchy of clustering’s which can be displayed as a diagram known as a dendogram. Agglomerative hierarchical methods begin with the each observation in a separate cluster. These clusters are then merged, according to their similarity (the most similar clusters are merged at each stage), until only one cluster remains. Divisive hierarchical methods work in the opposite way. An initial cluster containing all the objects are divided into sub-groups (based on dissimilarity) until each object has its own group. Agglomerative methods are more popular than divisive methods.

Unlike hierarchical techniques, which produce a series of related clustering’s, optimization techniques produce a single clustering which optimizes a predefined criterion or objective function. The number of clusters in this clustering is either specified a priori or is determined as part of the clustering method. Optimization methods start with an initial partition of objects into a specified number of groups. Objects are then reassigned to clusters according to the objective function until some terminating criterion is met. These methods differ with respect to the starting partitions, the objective functions, the reassignment processes, and the terminating criteria (Yun Peng, Hongxin Wan, 2013) (Meng-Dar Shieh, Fang-Chen Hsu, 2013). Unlike hierarchical clustering techniques, optimization methods do not store similarity matrices. Thus the size of the data is not limited by storage space. However, there are a number of disadvantages affecting optimization methods:

(1) Some methods require the number of clusters a priori, and will divide the data into this number of clusters regardless of the data structure;
(2) Certain clustering criterion are biased towards particular cluster shapes, and will impose these shapes on the data;
(3) The performance of optimization techniques is highly dependent on the initial partition.

In this study, a data clustering algorithm based on Simple PSO, Roulette Wheel Selection and K-Means algorithm.

2. Particle Swarm Optimization

In computer science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

PSO is originally attributed to Kennedy, Eberhart and Shi (Kennedy, J.; Eberhart, R, 1995) (Shi, Y, Eberhart, R.C, 1998) and was first intended for simulating social behaviour (Kennedy, J, 1977) as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. An extensive survey of PSO applications is made by Poli (Poli, R, 2008). PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Bird flocking optimizes a certain objective function. Each particle knows its best value so far (pbest) and its position.

This information is analogy of personal experiences of each particle. Moreover, each particle knows the best value so far in the group (gbest) among pbests. This information is analogy of knowledge of how the other particles around them have performed. Namely, each particle tries to modify its position using the following information:

(1) current positions;
(2) current velocities;
(3) distance between the current position and pbest;
(4) distance between the current position and gbest.

This modification can be represented by the concept of velocity. Velocity of each particle can be modified by the following equation:

$$v_{id} = w \times v_{id} + c_1 \times rand() \times (P_{id} - X_{id}) + c_2 \times rand() \times (P_{gd} - X_{id}) \quad (1)$$

Where v_{id} is velocity of particle;

x_{id} is current position of particle;

w weighting function;

c_1 and c_2 determine the relative influence of the social and cognitive components;

P_{id} is pbest of particle i;

P_{gd} is gbest of the group;

The following weighting function is usually utilized in

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{iter_{\max}} iter \quad (2)$$

where $w_{\max}$ is initial weight;

$w_{\min}$ is final weight;

$iter_{\max}$ is maximum iteration number;

$iter$ is current iteration number.

Using the above equation, a certain velocity, which gradually gets close to pbest and gbest can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$X_{id} = X_{id} + V_{id} \quad (3)$$

The general flow chart of PSO is shown in Figure 1.

The features of the searching procedure of PSO can be summarized as follows:

(a) As shown in equation (1)(2)(3), PSO can essentially handle continuous optimization problem.

(b) PSO utilizes several searching points like genetic algorithm (GA) and the searching points gradually get close to the optimal point using their pbests and the gbest.

(c) The first term of right-hand side (RHS) of (1) is corresponding to diversification in the search procedure. The second and third terms of that are corresponding to intensification in the search procedure. Namely, the method has a well-balanced mechanism to utilize diversification and intensification in the search procedure efficiently.
The above feature (c) can be explained as follows. The RHS of (2) consists of three terms. The first term is the previous velocity of the particle. The second and third terms are utilized to change the velocity of the particle. Without the second and third terms, the particle will keep on “flying” in the same direction until it hits the boundary. Namely, it tries to explore new areas and, therefore, the first term is corresponding to diversification in the search procedure. On the other hand, without the first term, the velocity of the “flying” particle is only determined by using its current position and its best positions in history. Namely, the particles will try to converge to the pbests and/or gbest and, therefore, the terms are corresponding to intensification in the search procedure.

3. Proposed PSO for Data clustering

The original PSO described in section 3 is basically developed for continuous optimization problems. However, lots of practical engineering problems are formulated as combinatorial optimization problems. Kennedy and Eberhart developed a discrete binary version of PSO for the problems (R. Eberhart and J. Kennedy, 1995). The proposed system employs Discrete Binary PSO with gobalized and localized search.

3.1 Problem Formulation

The fitness of panicles is easily measured as the quantization error. The fitness function of the data clustering problem is given as follows:

$$f = \frac{\sum_{i=1}^{N_C} \left\{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \right\}}{N_C}$$

The function f should be minimized.

where m_{ij} is j-th data vector belongs to cluster i;

o_i is centroid vector of the i-th cluster;

$d(o_i, m_{ij})$ is the distance between data vector m_{ij} and the cluster centroid o_i;

p_i stands for the number of data set, which belongs to cluster C_i;

N_c is number of clusters.

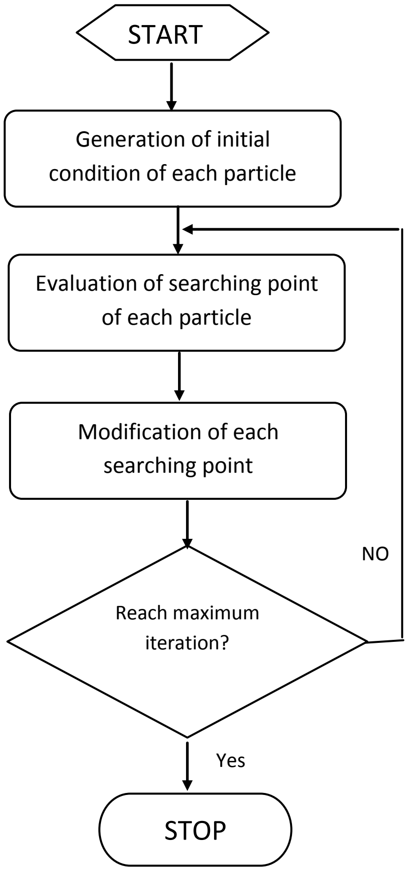
START
Generation of initial condition of each particle
Evaluation of searching point of each particle
Modification of each searching point
Reach maximum iteration?
NO
Yes
STOP

Figure 1. The general flow of PSO

3.2 Particle Representation

In the context of clustering, a single particle represents the cluster centroid vectors. That is, each particle X_{ij} **,** is constructed as follows:

$$X_{ij} = (m_{i1}, m_{i2}, ..., m_{im})$$

where m_{ij} refers to the j-th cluster centroid vector of the i-th particle in cluster C_{ij} . Therefore, a swarm represents a number of candidates clustering for the current data vectors.

3.3 Initial Population

One particle in the swarm represents one possible solution for clustering. Therefore, a swarm represents a number of candidate clustering solutions for the data set. At the initial stage, each particle randomly chooses k different data set from the collection as the initial cluster centroid vectors and the data sets are assigned to cluster based on one iteration of K-Means (MacQueen, J. B, 1967) (Lloyd, S. P, 1957).

3.4 Local search

After finding the solutions of N particles, a local search is performed to further improve fitness of these solutions. Local search helps to generate better solutions, if the heuristic information can not be discovered easily. Local search is applied on all generated solutions or on a few percent N. In this work, local search is performed on 20% of the total solutions. So in the test data set of N data, local search is applied on the 20% of solutions based on roulette-wheel selection. The requirement is that the fittest individuals have a greater chance of selection than weaker ones. In the local search procedure, the objective function values selected particles are computed again. These solutions can be accepted only if there is an improvement on the fitness, namely, if the newly computed objective function value is lower than the first computed value, newly generated solution replaces the old one.

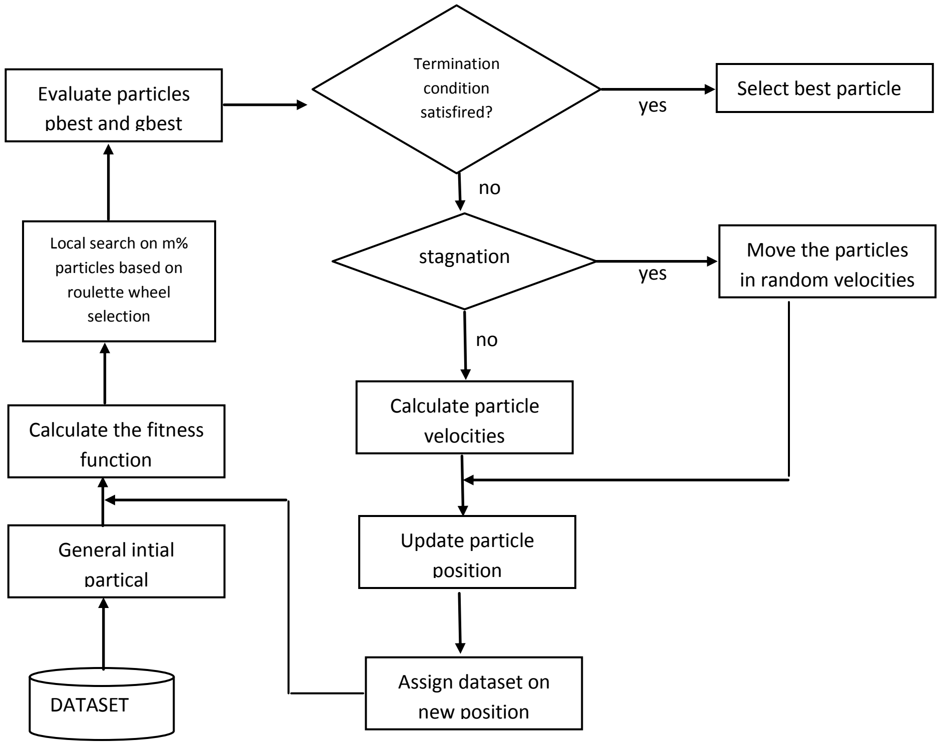
Evaluate particles pbest and gbest
Termination condition satisfired?
yes
Select best particle
no
Local search on m% particles based on roulette wheel selection
stagnation
yes
Move the particles in random velocities
no
Calculate the fitness function
Calculate particle velocities
General intial partical
Update particle position
DATASET
Assign dataset on new position

Figure 2. Hybrid PSO for Data Clustering

3.5 Personal best & Global best positions of particle

The personal best position of particle is calculated as follows

$$P_{id}(t+1) = \begin{cases} P_{id}(t) & if \quad f(X_{id}(t+1)) \geq f(P_{id}(t)) \\ X_{id}(t+1) & if \quad f(X_{id}(t+1)) < f(P_{id}(t)) \end{cases}$$

The particle to be drawn toward the best particle in the swarm is the global best position of each particle. At the start, an initial position of the particle is considered as the personal best and the global best can be identified with minimum fitness function value.

3.6 Finding new solutions

According to its own experience and those of its neighbors, the particle adjusts the centroid vector position in the vector space at each generation. The new velocity is calculated based on equation (1) and changing the position based on equation(3) Generally, in PSO algorithm, operations described above are iterated in main loop until a certain number of iterations are completed or all particles begin to generate the same result. This situation is named as stagnation behavior, because after a point, algorithm finishes to generate alternative solutions. The reason of this situation is, after a certain number of iterations, particles generate continuously the same solutions. Aiming minimizes the stagnation behavior of particles, the proposed technique follows the Quantization error of particles and if there is no change on the error after last 10 iterations, it moves the particles with the random velocities. In other words, to improve the solution, a feedback technique is applied on the algorithm. Figure 2 demonstrates the proposed Hybrid PSO for data clustering.

4. Experiment Results

In this section, results from the proposed PSO method and the K-Means on well-known test data sets are reported. The choice of the parameter values seems not to be critical for the success of the methods; it appears that faster convergence can be obtained by proper fine-tuning. The balance between the global and local exploration abilities of the proposed system is mainly controlled by the inertia weight, since the positions of the particles are updated according to the classical PSO strategy. A time decreasing inertia weight value, i.e., start from 0.9 and gradually decrease towards 0.4, proved to be superior to a constant value. The optimal solution (fitness) is determined with N=20, c1= 2.1 & c2 = 2.1. The test data sets are obtained from UCI's machine learning repository (UCI Repository, web). The Results obtained from test data sets by K-Means and the proposed system are shown in Table 1 & Table 2 respectively.

Iris plants database: This is a well-understood database with **4** inputs, **3** classes and 150 data vectors.

Wine: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Glass identification: From USA Forensic Science Service; 6 types of glass; defined in terms of their oxide content. For each data set with two different distance measures 50 runs have been performed using the proposed PSO and the performance is exhibited in terms of the Fitness value, Inter and Intra Cluster similarity. Results for all of the aforementioned datasets are reported with the conventional cluster algorithm K-Means. Table 1 illustrates the analysis of the results for K-Means and Table 2 shows for Proposed PSO system

5. Conclusion

The advantages of the PSO are very few parameters to deal with and the large number of processing elements, so called dimensions, which enable to fly around the solution space effectively. On the other hand, it converges to a solution very quickly which should be carefully dealt with when using it for combinatorial optimization problems. In this study, the proposed PSO algorithm developed for data-clustering problem is verified on the datasets. It is shown that it increases the performance of the clustering and the best results are derived from the proposed technique. Consequently, the proposed technique markedly increased the success of the data-clustering problem.

Data sets	**Distance Measure**	**K-Means Clustering**		
		FV	Intra	Inter
Iris	Euclidean	0.8013	0.0616	5.2805
	Chebychev	0.6873	0.1902	4.7052
Wine	Euclidean	126.14	11.4103	759.170
	Chebychev	124.68	11.0918	759.008
Glass	Euclidean	1.5968	0.49094	6.2713
	Chebychev	1.1856	0.2544	5.0068

Table 1: Analysis with K-Means

Data sets	**Distance Measure**	**Proposed PSO System**		
		FV	Intra	Inter
Iris	Euclidean	0.5439	0.0616	9.8228
	Chebychev	0.4209	0.0537	9.2193
Wine	Euclidean	83.826	5.4399	831.25
	Chebychev	83.416	3.9643	822.12
Glass	Euclidean	0.5991	0.4909	10.2561
	Chebychev	0.4209	0.1569	9.8352

Table 2: Analysis with Proposed PSO System

References

Estivill-Castro, V, “Why so many clustering algorithms”, ACM SIGKDD Explorations Newsletter, 4: 65, 2002.

Yun Peng, Hongxin Wan, "Web Text Clustering and Evaluation Algorithm Based on Fuzzy Set", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 7, No. 1, pp. 11-18, 2013.

Meng-Dar Shieh, Fang-Chen Hsu, "Using FCM Clustering for Consumer Segmentation in Kansei Engineering System", JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 7, No. 1, pp. 563 -571, 2013.

ZHANG Hong-qi, WANG Chun-guang, "Study on the Time Sequence Data Stream Clustering Algorithm Based on Property Information Contribution", IJACT: International Journal of Advancements in Computing Technology, Vol. 5, No. 2, pp. 148-154, 2013.

Kennedy, J.; Eberhart, R, "Particle Swarm Optimization", Proceedings of IEEE International Conference on Neural Networks. IV, pp. 1942-1948, 1995.

Shi, Y, Eberhart, R.C, "A modified particle swarm optimizer", Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73, 1998.

Kennedy, J, "The particle swarm: social adaptation of knowledge", Proceedings of IEEE International Conference on Evolutionary Computation, pp. 303–308, 1977.

Poli, R, "An analysis of publications on particle swarm optimisation applications", Technical Report CSM-469, Department of Computer Science, University of Essex, UK, 2007.

Poli, R, "Analysis of the publications on the applications of particle swarm optimisation", Journal of Artificial Evolution and Applications, 1, 10 2008.

R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp.39-43, 1995.

MacQueen, J. B, "Some Methods for classification and Analysis of Multivariate Observations", Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press. pp. 281-297, 1967.

Steinhaus, H, "Sur la division des corps matériels en parties" (in French), Bull. Acad. Polon. Sci. 4 (12): 801–804, 1957.

Lloyd, S. P, "Least square quantization in PCM", IEEE Transactions on Information Theory 28 (2): 129–137, 1957.

UCI Repository for Machine Learning Databases retrieved from the World Wide Web:http://www.ics.uci.edu/~mlearn/MLRepository.htm.