HYBRID CLUSTERING ALGORITHM BASED ON PSO WITH THE MULTIDIMENSIONAL ASYNCHRONISM AND STOCHASTIC DISTURBANCE METHOD

JUNYAN CHEN

School of Economics and Management, Tianjin Institute of Urban Construction, Tianjin, 300384, Tianjin, China

ABSTRACT

It is well known that solutions of k-means algorithm depend on the initialization of cluster centers and the final solution converges to local minima. In this paper, we introduce a clustering approach that combines ideas from modified particle swarm optimization (PSO) and k-means. The potential benefits of this technique are investigated by incorporating the multidimensional asynchronism and stochastic disturbance method to the velocity in the particle swarm optimizer to create new modifications of the clustering for k-means algorithmic model, which could keep populations diversity and ability of search global optimum as well as solve the problem of the curse of dimensionality. The simulation results of web log dataset show that the proposed algorithm, compared with the five previous developed PSO techniques, provides enhanced performance and maintains more diversity in the swarm.

Keywords: *Multidimensional Asynchronism, Stochastic Disturbance, Particle Swarm Optimization, Clustering, K-means algorithm*

1. INTRODUCTION

Clustering has been effectively applied on a variety of engineering and scientific disciplines such as psychology, biology, medicine, computer science, communications, and remote sensing [1-4]. In recent years, it has been recognized that the partitional clustering technique, such as K-means, is well suited for clustering large datasets, such as document clustering. The K-means algorithm has two main advantages. 1) It is very easy to implement, and 2) the time complexity is almost linear and only $O(n)$, which makes it suitable for large datasets. At the same time, it suffers from several drawbacks due to its choice of initializations. However, if favorable initial clustering centroids can be obtained using any of the other techniques, K-means would work well in refining the clustering centroids to find the optimal clustering centers. Since evolutionary computing is capable of avoiding convergence on a local solution, these approaches could be used to find a desired or even globally optimal solution. Clustering can be considered as an optimization problem (but its convergence on global optima cannot be proven); therefore, the desired solution with smaller fitness values than the other methods would be located by evolutionary computing.

Recently, many clustering algorithms based on evolutionary computing have been proposed, including Genetic Algorithms (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO), Ant Colony, Optimization (ACO) and Differential Evolution (DE) [1-3, 5, 6]. Paterlini [5] suggested that PSO should receive primary attention in partitional clustering algorithms provided the scholars with the direction to study the integration of partitional clustering and evolutionary computing. There have been many attempts to optimize clustering algorithm with PSO. Omran[6] first introduced PSO-based clustering algorithm, which was subsequently applied for pattern recognition and image processing. Liu [7] proposed an algorithm combined K-means clustering and PSO, called PSO-K, the results shows that its good convergence capability and optimizes capability better than both K-means and K-means based on GA. On the other hand, many scholars have considered noise injection to improve PSO performance using the probability theory. This could be viewed as two aspects of improvements:

1) to change parameters, such as acceleration coefficients, random variable and inertia weights, and 2) generation of a new location of a particle by the random mutation operator. Krohling [8] presented two random variable numbers generated from the Gaussian distribution. Higashi [9] introduced an approach using the mutation operator to change the location of the particles according to the Gaussian distribution. Magoulas [10] attempted to use a fixed constriction coefficient through a probability distribution characterized by the q entropic index of the nonextensive entropy. These methods tried to improve the performance of the standard PSO, but there have been few comparisons of the effectiveness of different mutation operators on particle velocity.

Although our simulation results already shows that PSO-K clustering is an effective and efficient algorithm for large datasets, it could not avoid the major disadvantage of premature convergence, since the particles' search ability is heavily dependent on the initial status of the location, velocity vector of particles and their interaction with each other when K-means only acts as a local search role in PSO-K model. Therefore, the key factor influencing the performance of PSO in this problem is the velocity vector of the particle. The expected situation is that the velocity vector has both global search ability in the global space and local search ability in the lesser local space.

In this paper, we incorporate the multi-dimensional asynchronism and stochastic disturbance method to the velocity in the particle swarm optimizer for the new k-means clustering algorithm (MSPSO-K). The potential benefits of this technique could keep populations diversity and ability of search global optimum as well as solve the problem of the curse of dimensionality. To demonstrate the effectiveness of MSPSO-K, we have applied the algorithms on weblog sets and got very good results compared to other five algorithms. The evaluation of the experimental results shows considerable improvements.

The rest of this paper is organized as follows: Section 2 gives a brief introduction of the K-means clustering. Section 3 gives a brief overview of the PSO and PSO clustering. The proposed method MSPSO-K is described in Section 4, and experimental results are given in Section 5. Finally, Section.6 concludes the paper.

2. K-MEANS CLUSTERING

K-means algorithm is one of the most commonly used and most efficient clustering algorithms. The procedure of K-means algorithm starts with K cluster centroids which are initially randomly selected and then assigned to the closest cluster center. The centroids are updated by the mean of that group.

The K-means algorithm is composed of the following steps:

Step1. Randomly initialize the K cluster centroid vectors.

Step2. Assign each object to the cluster that has the closest centroid, where the distance to the centroid is determined using

$$d\left(x_i, g_j\right) = \left\|x_i - g_j\right\| \tag{1}$$

Where x_i denotes the data vectors that belongs to cluster G_j, which is the subset of data vectors that form the cluster; g_j stands for the centroid vector.

Step3. Recalculate the cluster centroid vector with Eq. (2)

$$g_j = \frac{1}{n_j} \sum_{\forall x_i \in G_j} x_i \tag{2}$$

Where n_j is the number of data vectors that belong to cluster G_j.

Step4. Repeat steps 2 and 3 until convergence is achieved.

3. PARTICLE SWARM OPTIMIZATION

PSO mainly learns from a scenario in which a group of birds randomly searches for food in a given area and uses it to solve optimization problems. In PSO, each solution is like a "bird" in the search space, which is called a "particle." All particles have fitness values that are evaluated by the fitness function to be optimized, and have velocities vector that direct the flying of the particles.

3.1 Structure of PSO[11]

Particle swarm $S = \left\{Z_1, Z_2, ..., Z_i ..., Z_o\right\}$, where Z_i (the i^{th} particle current location) represents the candidate problem solution in D, the dimensional search space for which the i^{th} particle comprises the group.

At each generation, each particle moves in the search space with a velocity vector according to its own previous best location and its group's previous best location. The previous best location of the i^{th} particle $P_i = \left(p_{i1}, p_{i2}, \ldots, p_{id,} \ldots, p_{iD}\right)$ records a relatively good point in the search space, and all of the particle's previous best location results in $P_g = \left(p_{g1}, p_{g2}, \ldots, p_{gd} \cdots p_{gD}\right)$ (the previous global version of the best value). Every particle also has a velocity vector $V_i = \left(v_{i1}, v_{i2}, \ldots, v_{id,} \ldots, v_{iD}\right)$, which is adjusted and moves the particle to a new location. V' refer to the velocity vector of the next update. The solution depends on the fitness function. Each particle updates its location with the following two equations.

$$V_i' = \omega V_i + c_1 rand1\left(P_i - Z_i\right) + c_2 rand2\left(P_g - Z_i\right) \quad (3)$$

$$Z'_i = Z_i + V'_i \quad (4)$$

where inertia weight ω balances the search ability between the global and local particles [12]. The cognitive and social learning rates are c_1 and c_2, respectively; while $rand1$ and $rand2$ are random numbers distributed in the $(0,1)$.

3.2 PSO clustering

A clustering algorithm based on PSO was proposed by Omran [6], where the position Z_i=(z_{i1}, z_{i2}, z_{ij},...,z_{im}) of the particle represents the aggregate of the cluster centroid vector. z_{ij} refers to the j^{th} cluster centroid vector of the i^{th} particle. Dataset $X = \{x_n, n = 1, 2, ..., N\}$ is a given in the D-dimensional space. The partitioning of X into m clusters belong to datasets $G_i = \{G_{i1}, G_{i2}, ...G_{ij}, ...G_{im}\}$ and G_{ij} is composed of x_n of k_{ij}.

Usually, it evaluates the fitness of particles with distance measures, as in

$$F(Z_i) = \sum_{j=1}^{m} \sum_{\forall x_n \in G_{ij}} d\left(x_n, z_{ij}\right) \quad (5)$$

The $d\left(x_n, z_{ij}\right)$ could refer to many kinds of functions such as Euclidean distance, defined as (6). The function depicts the sum of all the intra-cluster distance, in which lower is better.

$$d(x_n, z_{ij}) = \left\| x_n - z_{ij} \right\| \quad (6)$$

Using PSO, data vectors can be clustered as follows:

```
Begin
Input $X$, the number of clusters $m$, the number of particles $n$, and the max generation number $t_{max\_gener}$
Initialize each particle to contain $m$ randomly selected cluster centroids
For $t = 1$ to $t_{max\_gener}$
Assign $x_n$ to the closest cluster $G_{ij}$ such that
    $d\left(x_{\mathrm{n}}, z_{ij}\right) = \min_{\forall j=1,\ldots,m} \left\{ d\left(x_n, z_{ij}\right) \right\}$    (7)
Calculate the fitness $F(Z_i)$ with (5)
    If the current value is better than the fitness $F(P_i)$
        Reset $P_i$
    End
    If the current value is better than the fitness $F(P_g)$,
        Reset $P_g$
    End
Update particles' location and velocity vector according to (3) and (4)
End
  Output the best result
End
```

4. MSPSO-K ALGORITHM

4.1 Analysis of Theories

In the PSO-K clustering algorithm [7], the fast convergence of the K-means and the ability of global searching of the PSO are combined. At every iteration, PSO results from the particle swarm's search are the initial seed for the K-means algorithm, which can then quickly locate new clustering centers with one repetition.

In addition, the K-means only acts in a local search role to gain the local optima near the initial solution from the PSO in the PSO-K, which could make the PSO-K transfer quickly from a global search to a local search. Therefore, the PSO-K would have higher probability of being trapped near the local optima. However, the character of PSO-K demands the particle swarm to exploit wider search space for finding better solutions to escape from the worse local optima.

As section 1 presented, some mutation methods could extend the PSO's search region, thus, we introduce some approaches that similar to the noise injection method to modify the algorithm.

4.2 The strategy of MSPSO-K

To solve the aforementioned problems and improve the performance of PSO-K, this paper proposes a hybrid clustering algorithm with the multidimensional asynchronyism and stochastic disturbance method that is MSPSO-K, with some variations, which improves the particle's search ability of optimization process and avoids the disadvantage of the K-means. The main optimization models are outlined as follows.

The n^{th} pattern $x_n = (x_{n1}, x_{n2}, ..., x_{nd}, ..., x_{nD})$, $n = (1, 2, ..., N)$ is a vector of D dimension. The given $x_\alpha = (x_{\alpha 1}, ..., x_{\alpha d}, ..., x_{\alpha D})$ and $x_\beta =$ $(x_{\beta 1}, ..., x_{\beta d}, ..., x_{\beta D})$ are the two arbitrary vectors in the x_n during the clustering process. $V^j_{i\_random}$ is a random velocity vector in the j^{th} cluster of the i^{th} particle.

Definition 1 $V^j_{i\_max}$ is defined as the maximal velocity vector corresponding to z_{ij}

$$V^j_{i\_max} = \left[\left(v^j_{i\_max}\right)_1, ..., \left(v^j_{i\_max}\right)_d, ..., \left(v^j_{i\_max}\right)_D\right] \quad (8)$$

where $\left|\left(v^j_{i\_max}\right)_d\right|$ as follows:

$$\left|\left(v^j_{i\_max}\right)_d\right| = \min_{1 \le \alpha < \beta \le N} \left|x_{\alpha d} - x_{\beta d}\right| \quad d = 1, 2, \cdots, D \quad (9)$$

Therefore

$$\left|V^j_{i\_max}\right| = \left[\left|v^j_{i\_max}\right|_1, ..., \left|v^j_{i\_max}\right|_d, ..., \left|v^j_{i\_max}\right|_D\right] \quad (10)$$

where

$$\left|v^j_{i\_max}\right|_d = \left|\left(v^j_{i\_max}\right)_d\right| \quad d = 1, 2, \cdots, D \quad (11)$$

Definition 2 A stochastic matrix is defined as follows:

$$R(r) = \begin{bmatrix} r_1 & & & & \\ & \cdots & & & \\ & & r_d & & \\ & & & \cdots & \\ & & & & r_D \end{bmatrix} \quad d = 1, 2, \cdots, D$$

(12)

where r_d are random numbers generated according to the absolute value of the uniform probability distribution, random number $r_d \in (-1, 1)$.

Definition 3 V_{i_random} is composed of $V^j_{i\_random}$, and the $V^j_{i\_random}$ is given by

$$V^j_{i\_random} = \left|V^j_{i\_max}\right| \cdot R(r) \quad (13)$$

$$V^j_{i\_random} = \left[\left(v^j_{i\_random}\right)_1, ..., \left(v^j_{i\_random}\right)_d, ..., \left(v^j_{i\_random}\right)_D\right]$$

(14)

Definition 4

$$V_i' = \omega V_{i\_random} + c_1 rand1(P_i - Z_i) + c_2 rand2(P_g - Z_i)$$

(15)

As for the probability distribution of random velocity vectors in Eq. (15), we use the uniform distribution. According to the uniform probability distribution, the probability in which V_{i_random} would be a value occurring is equal between two points. This stochastic disturbance to the velocity vectors method succeed to the advantages of searching continuity of standard PSO and improve the superior global searching ability.

Definition 5 λ is given by

$$\lambda = \frac{1}{2}\left(1 - \frac{t}{iter_{\max}}\right) \quad (16)$$

Definition 6 If $flag(t-1) > \mu_1$, $\varepsilon > \lambda$, then

$$\left(v^j_{i\_random}\right)_d = 0 \quad (17)$$

Where ε is random number distributed in the $(0,1)$, $flag(t\text{-}1)$ presents the continuous update times of $F(P_i)$ of i^{th} particle after the $(t\text{-}1)^{th}$ of generations, μ_1 is a parameter.

According to Eq. (16), λ is initialized to 0.5, then linearly decreases while the iteration number increases, which have a dynamic probability of occurrence of multidimensional asynchronism to the velocity vectors as in Eq. (17). The multi-dimensional asynchronism strategy could enhance the local search capabilities in different dimensions and reduce the difficulty of particles to search in multidimensional space.

4.3 The steps of the MSPSO-K algorithm

The steps of the MSPSO-K algorithm are listed as follows:

```
Begin
Input X, the number of clusters $m$, the number of particles $n$, the max generation number $t_{max\_gener}$, and parameter $\mu_1$
Initialization of particle swarm
Assign each object $x_n$ to the cluster $G_{ij}$ according to the Nearest Neighbor Optimal Algorithm, in which the distance to the centroid is determined using Eq. (7)
Calculate $F(Z_i)$ with Eq. (5) and Eq. (6)
Set
        $t = 0$        (18)
and
        $flag(t) = 1$        (19)
For $t = 1$ to $t_{max\_gener}$
    Update particles' location and velocity vector according to Eqs. (15), (17) and (4)
    Optimize particles with K-means: Assign each object $x_n$ to the cluster $G_{ij}$ according to the Nearest Neighbor Optimal Algorithm, in which the distance to the centroid is determined using Eq. (7)
    Update centroids with Eq. (20) as follows:
        $z_{ij} = \frac{1}{k_{ij}} \sum_{x_n \in C_{ij}} x_n$        (20)
    Calculate $F(Z_i)$ with Eqs. (5) and (6)
    If $F(Z_i) < F(P_i)$
    Reset $P_i$
        $flag(t) = flag(t) + 1$        (21)
    Else
        $flag(t) = 0$        (22)
        End If
        If $F(Z_i) < F(P_g)$
        Reset $P_g$
        End If
Next
Output the best result
End
```

5. SIMULATION EXPERIMENT.

An experimental subject obtained from a commercial website was chosen. After extracting from the web log file, 726 clients and 52pages were obtained. All client data, the value of which is 1 or 0, were separately clustered to 15 classes, 25 classes and 35 classes with PSO[6], PSO-K [7], GPSO-K1 [9], GPSO-K2 [8], QPSO-K [10], and MSPSO-K algorithms. The iterations number is 50 except PSO clustering set as 500.

In Table 1, the columns from left to right report the names of the algorithms, mean fitness values, best fitness values, worst fitness values. In Table 2, the columns are standard deviations and 95% confidence interval for the mean.

Table 1 : Performance Comparisons-1 With Different Methods

m	Algorithms	Mean	Best value	Worst value
15	*PSO (500)*	1,108.19	1,093.14	1,122.56
	PSO-K	1,022.12	1,017.30	1,028.34
	GPSO-K1	1,019.87	1,015.30	1,023.04
	GPSO-K2	1,019.03	1,014.30	1,023.22
	QPSO-K	1,018.36	1,015.45	1,022.13
	MSPSO-K	1,011.71	1,009.63	1,013.20
25	*PSO (500)*	1,083.29	1,058.44	1160.55
	PSO-K	981.10	975.53	988.32
	GPSO-K1	980.81	975.23	984.38
	GPSO-K2	981.10	978.21	983.95
	QPSO-K	977.84	973.38	981.67
	MSPSO-K	969.64	966.03	970.97
35	*PSO (500)*	1,114.07	1,095.52	1,125.31
	PSO-K	948.76	942.04	956.98
	GPSO-K1	948.94	945.67	952.88
	GPSO-K2	950.68	947.20	953.63
	QPSO-K	947.41	944.01	951.23
	MSPSO-K	936.79	934.65	938.90

Table 2 : Performance Comparisons-2 With Different Methods

m	Algorithms	Std	95% Conf int
15	PSO (500)	6.81	[1,105.60, 1,110.78]
	PSO-K	3.22	[1,020.89, 1,023.34]
	GPSO-K1	1.87	[1,015.30, 1,023.04]
	GPSO-K2	2.25	[1,018.20, 1,019.87]
	QPSO-K	1.46	[1,017.81, 1,018.91]
	MSPSO-K	1.02	[1,011.32, 1 ,012.10]
25	PSO (500)	23.88	[1,074.41, 1,092.17]
	PSO-K	3.87	[979.66, 982.54]
	GPSO-K1	2.39	[979.92, 981.70]
	GPSO-K2	1.71	[978.21, 983.95]
	QPSO-K	1.96	[977.11, 978.57]
	MSPSO-K	1.03	[969.25, 970.02]
35	PSO (500)	6.72	[1,111.51,1,116.62]
	PSO-K	4.12	[947.13, 950.40]
	GPSO-K1	1.86	[948.25, 949.64]
	GPSO-K2	1.91	[949.97, 951.39]
	QPSO-K	1.64	[946.80, 948.02]
	MSPSO-K	1.24	[936.27, 937.31]

It can be observed from Table 1, Table 2, compared with others, MSPSO-K always performed better in terms of best value, worst value, and standard deviation, and the mean. Moreover, the performance of PSO-K, GPSO-K1, GPSO-K2, and QPSO-K are very close, in which QPSO-K has the best performances. Table 3 shows the results of unpaired *t* -tests between the new algorithm (MSPSO-K) and the best algorithm (QPSO-K) of the other five in each dataset. It should be mentioned that the differences in the results between the MSPSO-K algorithm and the other algorithms are extremely significant in the three instances from the web log dataset. It could be seen the learning curves of the algorithms (PSO-K, GPSO-K1, GPSO-K2, and QPSO-K) from Fig.1.

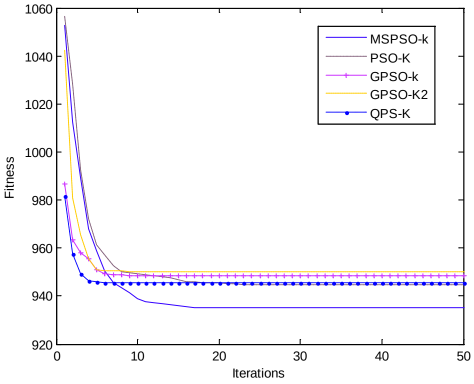

Figure 1: The Learning Curves Of The Algorithms (m = 35)

Table 3: Results Of Unpaired T-Tests On The Data Of Web Log Dataset

m	Std err	*t*	95% Conf int	Two-tailed *P*	Significance
15	0.336	19.8	[5.977, 7.324]	<0.0001	*Extremely significant*
25	0.404	20.3	[7.399, 9.015]	<0.0001	*Extremely significant*
35	0.381	26.3	[9.278, 10.806]	<0.0001	*Extremely significant*

6. CONCLUSION

This paper presented a hybrid and modified PSO for K-means strategy to cluster large-scale data. From the results of the experiments, the proposed MSPSO-K algorithm have been shown that it can improve the global searching ability of the PSO with a multidimensional asynchronism and stochastic disturbance model to preserve the population diversity of the PSO, ensure searching capability in high dimension, and take full advantage of the K-means in local search ability. MSPSO-K can be further used to more large-scale datasets (such as the web, image and video).

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education of Humanities and Social Science project (No. 12YJC630012) and Social Science Planning Fund of Tianjin (No. TJGL11-109).

REFERENCES:

[1] B. Bahmani Firouzi, M. Sha Sadeghi, and T. Niknam, "A new hybrid algorithm based on PSO, SA, and K-means for cluster analysis", *International Journal of Innovative Computing Information and Control*, Vol. 6, No.5, 2010, pp. 3177-3192.

[2] T. Niknam, B. Amiri, "An efficient hybrid approach based on PSO, ACO and kmeans for cluster analysis", *Applied Soft Computing*, Vol. 10, No. 1, 2010, pp. 183-197.

[3] Y. Liu, Xi. Wu, and Y. Shen, "Automatic clustering using genetic algorithms", *Applied Mathematics and Computation*, Vol. 218, No. 4, 2011, pp. 1267-1279.

[4] T. Niknam, E.T. Fard, N. Pourjafarian, and A. Rousta, "An efficient hybrid algorithm based on modified imperialist competitive algorithm and

K-means for data clustering", *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 2, 2011, pp.306-317.

[5] S. Paterlini, T. Krink, "Differential evolution and particle swarm optimization in partitional clustering", *Computational Statistics and Data Analysis*, Vol. 50, No. 5, 2006, pp. 1220-1247.

[6] M. Omran, A. Salman, and AP. Engelbrecht, "Image Classification using Particle Swarm Optimization", Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Nanyang Technological University, November 18-22, 2002, pp. 370-374.

[7] J. Liu, L. Han, and L. Hou, "Cluster Analysis Based on Particle Swarm Optimization Algorithm", *Systems Engineering-Theory & Practice*, Vol. 25, 2005, pp. 54-58.

[8] R. A. Krohling, F. Hoffmann and L. dos Santos Coelho Co-evolutionary particle swarm optimization for min-max problems using Gaussian distribution, Proceedings of the IEEE Congress of Evolutionary Computation, IEEE Conference Publishing Services, June 19-23, 2004, pp. 959-964.

[9] N. Higashi and H. Iba, Particle swarm optimization with Gaussian mutation, Proceedings of the IEEE Swarm Intelligence Symp., IEEE Conference Publishing Services, October 5-8, 2003, pp.72-79.

[10] G. D. Magoulas, A. D. Anastasiadis, "Approaches to Adaptive Stochastic Search Based on the Nonextensive q-Distribution", *International Journal of Bifurcation and Chaos*. Vol.16, No.7, 2006, pp. 2081-2091.

[11] J. Kennedy, W. M. Spears, "Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and Some Genetic Algorithms on the Multsi-model Problem Generator", *Proceedings of the IEEE Intel Conference on Evolutionary Computation*, IEEE Conference Publishing Services, May 4-9, 1998, pp. 78-83.

[12] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization". *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, March 25-27, 1998, pp 591-601.