Journal of AI and Data Mining
Vol 3, No 1, 2015, 39-46.
doi:10.5829/idosi.JAIDM.2015.03.01.05

Fuzzy clustering of time series data: A particle swarm optimization approach

Z. Izakian* and M. S. Mesgari

Department of Geodesy & Geomatics & Geoinformation Technology Center of Excellence, K. N. Toosi University of Technology, Tehran, Iran.

Received 14 June 2014; Accepted 9 May 2015
**Corresponding author: zahedeh_izakian@yahoo.com (Z. Izakian).*

Abstract
Developing information gathering technologies and getting access to a large amount of data, we always require methods for data analyzing and extract useful information from large raw dataset. Thus, data mining is an important method for solving this problem. Clustering analysis as the most commonly used function of data mining, has attracted many researchers in computer science. Because of different applications, the problem of clustering the time series data has become highly popular and many algorithms have been proposed in this field. Recently Swarm Intelligence (SI) as a family of nature inspired algorithms has gained huge popularity in the field of pattern recognition and clustering. In this paper, a technique for clustering time series data using a particle swarm optimization (PSO) approach has been proposed, and Pearson Correlation Coefficient as one of the most commonly-used distance measures for time series is considered. The proposed technique is able to find (near) optimal cluster centers during the clustering process. To reduce the dimensionality of the search space and improve the performance of the proposed method, a singular value decomposition (SVD) representation of cluster centers is considered. Experimental results over three popular data sets indicate the superiority of the proposed technique compared with fuzzy C-means and fuzzy K-medoids clustering techniques.

Keywords: *Clustering, Time Series, Particle Swarm Optimization, Singular Value Decomposition, Pearson Correlation Coefficient.*

1. Introduction

A time series is a sequence of data points, measured on a successive time space through a uniform interval. Meteorologists use time series for displaying climate change and forecasting weather. Demographers use time series for Anticipating population changes within a specified period of time. In Economics time series are used for analyzing and predicting stock price. Further analysis of time series in science, such as bioinformatics, geology, marine science, medicine and engineering are frequently used. Because of time series applications in various sciences, the interest to analyze these data has been increased. Challenges related to the analysis of these data (time series), is because of its not only size and volume, but also the complexity of this type of data. Time series clustering provides a way for reducing the complexity by categorizing time series in few groups. Grouping should be done in such a way that patterns in the same group should likely be similar to each other while maximizing the dissimilarity of different clusters. In general, the purpose of clustering is representing large datasets by a fewer number of cluster centers. It brings simplicity for large datasets and thus is an important step in the process of knowledge discovery and data mining. So far, different algorithms have been proposed for solving the problem of clustering spatio-temporal data such as time series. Recently, a family of nature inspired algorithms, known as Swarm Intelligence (SI), has attracted the attention of researchers working on clustering field and Particle Swarm Optimization (PSO) is a popular optimization algorithm, which is based on swarm intelligence [1].

Fuzzy C-Means (FCM) [2] is a popular clustering technique and there is a membership degree in unit interval instead of assigning each object to

one cluster. This clustering algorithm has been employed successfully in many applications but there are some challenges in using this technique for clustering time series data. FCM is sensitive to initialization and may get trapped in a local optimum. Since time series are high-dimensional data, it is actually more probable that results fall into local optima. Moreover the most commonly used similarity measure in FCM is Euclidean distance but sometimes in time series data, using another similarity/dissimilarity measures is more appropriate. To deal with the above-mentioned challenges, in this paper, a particle swarm optimization approach for time series data clustering has been proposed. In this method, PSO is applied to find optimal cluster centers based on the selected objective function and the selected similarity measure. For this purpose, a singular value decomposition (SVD) representation of time series is considered and PSO estimates SVD coefficients of cluster centers.

This study is organized as follows: In section 2, a brief literature review is presented. In section 3 we describe the fuzzy C-means technique. Section 4 focuses on PSO algorithm. In section 5, we briefly explain time series representation methods. In section 6, we focus on similarity measures. In section 7, the proposed method for clustering time series is explained, and in section 8, experimental studies are reported. Finally, section 9 concludes this work.

2. Literature review

In this section, we present some PSO based clustering methods. Niknam et al. [3] introduced a new clustering method based on combination of the Ant Colony Optimization and the Particle Swarm Optimization called PSO-ACO. They used ACO algorithm for decision making process of particle movement. This combination makes the particles search the surrounding area better. Results show that the proposed PSO-ACO optimization algorithm has much potential in allocating N objects to k clusters. Ahmadyfard and Modares [4] proposed a hybrid clustering method based on combination of the particle swarm optimization (PSO) and the k-mean algorithm to cluster dataset into a user specified number of cluster. They used the property of PSO in fast convergence during the initial stages of a global search and the fast convergence around global optimum property of k-means algorithm in their method. The performance of their method was compared to K-means and PSO clustering algorithms using five datasets. Hwang and Huang [5] presented a clustering algorithm based on particle swarm optimization (PSO) and fuzzy theorem. Their proposed algorithm can compute appropriate number of clusters and find cluster centers in a dataset. The result of comparing their algorithm with PSO clustering and fuzzy c-means on three datasets, showed good performance of the method in determining the number of clusters and clustering of data. Rana et al. [6] offered a new Hybrid Sequential clustering approach based on PSO and K-Means that overcomes the drawback of both algorithms. They used PSO in sequence with K-Means algorithm for data clustering. Their method improved the slow convergence of PSO near optimal solution. The obtained results of comparing presented method with K-Means, PSO and Hybrid K-Means-Genetic algorithms showed better performance of the method.

Premalatha and Natarajan [7] proposed a new approach integrating Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) for document clustering called HPSO. For convergence improvement, they applied the PSO algorithm capability in fast convergence and the genetic algorithm ability in exploiting previous solution. In this method, the crossover operation of GA is used in order to transmit information between two particles and the mutation operation is used to increase the population diversity. The results illustrated the efficiency of HPSO. Esmin and Matwin [8] presented a new clustering method based on particle swarm optimization and called it hybrid particle swarm optimization with mutation (HPSOM). This method was used to find the centroids of a user specified number of clusters. They applied mutation process of GA to improve the results obtained from PSO. Their approach was compared with K-means clustering method and the standard PSO algorithm on five benchmark datasets.

The results illustrated more efficiency of the proposed clustering method. Kamel et al. [9] proposed a new approach based on K means, PSO and Sampling algorithms for data clustering. Their proposed method was evaluated on four datasets and was compared with K means, PSO, Sampling+K means, and PSO+K means. The results showed that their approach generates the most compact clusters. Kamel and Gaikwad [10] proposed a new hybrid sequential clustering approach used for PSO algorithm in sequence with Fuzzy k means technique in data clustering. An experiment was done on standard datasets available online. The experimental studies showed the efficiency of the proposed method in detecting clusters.

This study used evolutionary algorithms for clustering datasets including short sequences in all methods provided in this section.
Applying evolutionary algorithms for long time series is time consuming and moreover with increasing the number of unknown elements, the efficiency of algorithm will be reduced. Therefore, we used time series dimensionality reduction technique to solve the problem of using evolutionary algorithms for long time series clustering.

3. Fuzzy C-means algorithm

Fuzzy C-Means is one of the commonly used fuzzy clustering methods proposed by Bezdek (1981) [2]. It is based on the minimization of the following objective function:

$$J(U,V)=\sum_{i=1}^{C}\sum_{j=1}^{n}u_{ij}^{m}d^{2}(v_i,x_j)\quad 1\le m\le\infty \tag{1}$$

Where m is fuzziness parameter and determines the level of cluster fuzziness, C is the number of clusters, n is the number of objects in the data set, v_i is the prototype of the center of cluster i, u_{ij} is the degree of membership of x_j in the cluster i and $d^2(v_i,x_j)$ is the distance between object x_j and cluster center v_i. V is a matrix including C cluster centers and U is partition matrix. A solution of the object function can be obtained via a sequence of iterations, which is carried out as follows:

1. Set values for C (the number of clusters) and m (fuzziness parameter)
2. Initialize U (fuzzy partition matrix)
3. Calculate V (the cluster centers) by using U

$$v_i=\frac{\sum_{j=1}^{n}u_{ij}^{m}x_j}{\sum_{j=1}^{n}u_{ij}^{m}} \tag{2}$$

4. Calculate the new partition matrix by using V

$$u_{ij}=\frac{1}{\sum_{k=1}^{C}\left(\frac{d(x_j,v_i)}{d(x_j,v_k)}\right)^{2/(m-1)}} \tag{3}$$

5. Repeat step 3 and 4 until stopping condition is true

4. Particle swarm optimization

Particle swarm optimization [11,12] is a population-based optimization technique inspired by the social behavior of bird flocking or fish schooling.
In PSO algorithm, each particle represents a possible solution that moves randomly in the search space towards the optimal solution. Displacement of each particle in the search space is influenced by their own and their neighbors knowledge. PSO algorithm tries to combine the local search method (using their experience) and global search method (using the experiences of neighbors) to achieve good results.
Let is the position of particle i at time t .The particle at time t +1 will change its position according to (4).

$$X_i(t+1)=X_i(t)+V_i(t+1) \tag{4}$$

Where $V_i(t+1)$ is the particle's velocity vector at time t+1. From (4) we can find that the position of each particle changes based on its previous position and the velocity vector. $V_i(t+1)$ is calculated from (5).

$$V_i(t+1)=w.V_i(t)+c_1R_1(Pbest_i(t)-X_i(t))+c_2R_2(Gbest_i(t)-X_i(t)) \tag{5}$$

In (5), $pbest_i(t)$ is the best known position of particle i since the beginning of the algorithm, and $nbest_i(t)$ is the best position found through its neighborhood.
c_1 and c_2 called acceleration constants and w is called inertia weight and increases the convergence speed. r_1 and r_2 are random positive numbers uniformly distributed in the range [0,1].
The personal best (*pbest*) of particle i is updated by:

$$pbest_i(t+1)=\begin{cases}pbest_i(t) & \text{if } f(pbest_i(t))\ge f(X_i(t+1))\\ X_i(t+1) & \text{if } f(pbest_i(t))<f(X_i(t+1))\end{cases} \tag{6}$$

where, $f()$ is the fitness function. According to (6) if a particle's current position is better than its previous best position, it is to be updated. If $X_k(t+1)$ is the best particle in the neighborhood of particle i at time t+1, it needs to be updated *Gbest* using (7).

$$Gbest_i(t+1)=\begin{cases}Gbest_i(t) & \text{if } f(Gbest_i(t))\ge f(X_k(t+1))\\ X_k(t+1) & \text{if } f(Gbest_i(t))<f(X_k(t+1))\end{cases} \tag{7}$$

The pseudocode for PSO algorithm is shown as table 1.

Table 1. Continuous PSO pseudocode (Elbeltagi et al. [13]).

```
Generate random population of P solutions (particles).
Repeat
for each particle i=1,…,P do
compute the fitness of particle i (Xi)
        if fitness(Xi) > fitness(pbesti) then
                pbesti= Xi;
    end
        if fitness(pbesti) > fitness(Gbesti)then
                Gbesti = pbesti;
    end
end
for each particle i=1,…,P do
        update the velocity vector using Eq. (5)
        update the position vector using Eq. (4)
end
until stopping condition is true;
return Gbest and corresponding position
```

5. Time series representation methods

Time series representation methods are categorized into data adaptive, non data adaptive and sometimes statistical methods. Among the methods, Single Value Decomposition is one of the most efficient techniques that is explained in this section.

Assume that we have *n* time series and each has *m* points. According to [14], if we consider these time series in the form of a matrix $A_{n\times m}$ then, it can be expressed as (8).

$$A = U \times S \times V^T \tag{8}$$

Where U is a $n \times n$ unitary matrix, S is $n \times m$ diagonal matrix and V is $m \times m$ orthogonal matrix called SVD-transform matrix. After calculating the SVD matrix, each time series $\boldsymbol{x}$ can be represented in the new space as follows:

$$y = xV \tag{9}$$

In this equation, $\boldsymbol{y}$ is a vector with *m* point so that except for its first *k* ($k << m$) coefficients, the other coefficient are almost equal to zero. Therefore, having just the first *k* coefficients in $\boldsymbol{y}$ is enough to represent the time series in the new space. Also, reconstructing the original time series from its SVD transform, $\boldsymbol{y}$, can be done by

$$x = yV^{-1} \tag{10}$$

Notice that if $\boldsymbol{y}$ is a vector with length *k*, zero padding should be performed to convert $\boldsymbol{y}$ to a vector with length *m*. Usually the first few coefficients of SVD are enough to capture the most important features of time series and the original time series can be reconstructed using these first few coefficients with a little loss of information.

Because of facing real numbers in using singular value decomposition coefficients (SVD) as one of the time series representation methods, high speed, high efficiency and popularity among researchers, we have selected this method to define particles in our proposed method.

6. Similarity measures in working with time series data

Selecting a suitable similarity measure or distance function is one of the main steps in clustering. A distance function is the criterion used for determining the similarity in a dataset. Selecting the right distance function is application dependent. Some similarity measures used for time series are Euclidean distance [15], Pearson correlation coefficient [16], LCSS distance [17] and DTW distance [18]. The similarity measure used in our work is Pearson correlation coefficient.

Pearson product moment correlation is the most commonly used measure of correlation, which is called simply Pearson correlation. The Pearson correlation shows the degree of the linear relation between two varieties. For two time series $\boldsymbol{x}$ and $\boldsymbol{y}$ with mean $\bar{x}$ and $\bar{y}$ and length *m* the Pearson coefficient can be calculated as (11).

$$\rho_{x,y} = \frac{\mathrm{cov}(\boldsymbol{x},\boldsymbol{y})}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{m}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{m}(y_i-\bar{y})^2}} \tag{11}$$

Where $\rho_{\mathbf{x},\mathbf{y}}$ is a number in range [-1, 1]. $\rho_{\mathbf{x},\mathbf{y}} = 1$ means the two time series are in a perfect positive correlation. In other words, an increase seen in one time series will lead to a proportionate increase in the other time series. $\rho_{\mathbf{x},\mathbf{y}} = -1$ means they are in a perfect negative correlation or an increase seen in one time series results in a proportionate decrease in the other time series and $\rho_{\mathbf{x},\mathbf{y}} = 0$ means there is no correlation between them. One may use $D(\mathbf{x},\mathbf{y}) = 1 - \rho_{\mathbf{x},\mathbf{y}}$ as a distance function between two time series. One may use $D(\mathbf{x},\mathbf{y}) = 1 - \rho_{\mathbf{x},\mathbf{y}}$ as a distance function between two time series.

7. Clustering of time series data using a PSO technique

As mentioned earlier, using FCM for clustering high-dimensional data may result in some local optima. To deal with the above-mentioned problems, we considered using a particle swarm optimization for clustering time series data.

The PSO clustering algorithm that was explained in section 4, is appropriate for short time series clustering. In datasets including long time series, finding the elements of cluster centers by particles

is time consuming and the large number of unknown parameters related to the cluster centers will reduce the efficiency of PSO algorithm. In the proposed method, the PSO algorithm finds the most important SVD coefficients of cluster centers instead of finding all elements of each cluster center. Then, all the cluster centers are reconstructed using corresponding SVD coefficients and with considering Pearson correlation coefficient as similarity measure, the objective function is calculated for each particle. This method significantly reduces the number of unknown parameters but increases the efficiency of the PSO algorithm.

In our proposed method for each particle, we have *C* cluster centers and every cluster center has *k* features, these features are the first *k* SVD coefficients of that cluster center. In this method, figure 1 shows each particle.

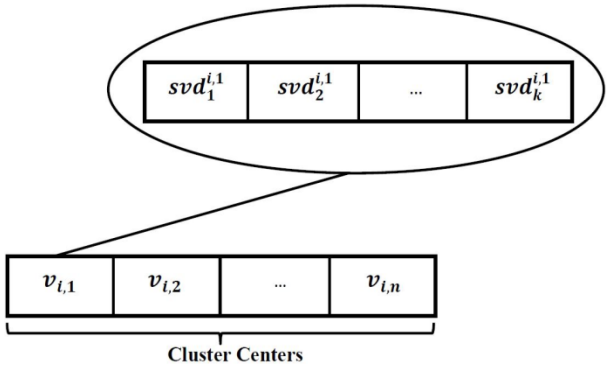

Figure 1. Particle scheme in the proposed method.

At the first step of the algorithm, a set of *p* particles including cluster center is generated randomly to comprise real numbers. The length of each cluster center is equal to the length of SVD representation of time series (the first k coefficients). Moreover, for each particle, a velocity vector is generated to include real numbers in a range. To evaluate a potential solution encoded by a particle, all the selected cluster centers are reconstructed to time series data using an inverse SVD. Next, (3) is used to estimate the membership degree of each trajectory to each cluster center and finally, the FCM objective function is considered as the fitness of each particle. FCM objective function is considered as the fitness of each particle.

Because of two reasons, we selected the objective function of FCM algorithm as our proposed method's objective function. First, this objective function takes into account compactness and separation of clusters well and second, we are going to compare our method with FCM and FKM clustering algorithms which try to minimize this objective function as their fitness function.

To update a particle in each step of the algorithm based on the calculated fitness values of particles, *pbest* of each particle and *gbest* of the whole population is updated and then (5) has been used for velocity vectors updating. As the result, (4) is used to update cluster center part. After updating particles, if in a particle there was a cluster center located outside of boundaries of SVD coefficients of the data set, it has to be generated randomly again. Table 2 shows the pseudo-code of the proposed method.

Table 2. Proposed method pseudo-code.

```
1: Create and initialize P particles with C cluster centers
2: FOR iteration_count = 1 to maximum_iterations DO
3:     FOR each particle i DO
4:         reconstruct cluster centers using their
           SVD coefficients.
5:         Calculate partition matrix (U) using
           E.q. (3).
6:         Calculate the fitness function f (Zi)
           using E.q. (1).
7:         Update pbest for each particle using
           E.q. (6).
8:     END
9:     Update Gbest using E.q. (7).
10:    FOR each particle i DO
11:        Update velocity for cluster centers.
           E.q. (5).
12:        Update cluster centers of each particle
           using PSO algorithm. E.q. (4).
13:    END
14:    END
15: Extract the cluster centers corresponding to
    Gbest.
```

8. Results and discussion

8.1. Dataset

We used three well-known [19, 20, 21] datasets including Cylinder Bell Funnel Data (CBF), Trace and Gun Point from the UCR Time Series Data Mining Archive [22]. The properties of each dataset were shown in table 3.

Table 3. The properties of datasets.

Data set	Dataset size	Time series length	Number of clusters
Cyliner Bell Funnel	900	128	3
Trace	200	275	4
Gun Point	200	150	2

8.2. Parameter setting

The parameters in the proposed method were set in table 4:

Table 4. The proposed method parameters.

Parameters	Value
m	2
p	30
itr	100
c_1	2
c_2	2
SVD	4

Where *m* is fuzziness parameter, *p* is the population size, *Itr* is number of iterations, c_1 and c_2 are inertia weight and *SVD* is the number of unknown SVD coefficients of clusters centers. A fine tuning has been performed to set the parameters of the proposed method for clustering time series data. In more data sets, the first four to five SVD coefficients are enough to capture the important features of time series; therefore, we use four coefficients for representing cluster centers in a particle.

8.3. Evaluation criteria

Precision, recall and f-measure are three most well-known validation techniques for clustering and classification. Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant, while recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Both precision and recall are therefore based on an understanding and the measure of relevance. The harmonic mean of precision and recall is another validation technique called F-measure. In this study, for comparing our method with other clustering algorithms, Objective function, Precision and F-Measure were considered as tools for evaluation. According to above explanations, the Precision is defined as (12).

$$P(i,j)=\frac{n_{ij}}{n_j} \tag{12}$$

Where n_{ij} is the number of members of class *i* in cluster *j*, n_j is the number of members of cluster *j* and n_i is the number of members of class *i*. the F-Measure is computed using 13.

$$F(i,j)=\frac{2\times P(i,j)\times R(i,j)}{P(i,j)+R(i,j)} \tag{13}$$

Where

$$R(i,j)=\frac{n_{ij}}{n_i} \tag{14}$$

Higher precision and F-Measure and lower objective function means that the performance of clustering algorithm was better and algorithm returned higher quality clusters.

8.4. Results

For evaluating the performance of the proposed method, we compared it with fuzzy C-means and fuzzy K-medoids clustering algorithms. By considering Pearson correlation coefficient as the similarity measurement, the following results were obtained. *J*, FM and PR respectively show the best value of fitness function, F measure and Precision achieved over 20 independent runs. Moreover, mean and standard deviation of run time over 20 runs are shown in table 5.

Table 5. Results of three clustering methods on three datasets.

Data set	method	J	FM	PR	Time (sec)
CBF	FCM	44.126	0.64	0.66	1.067±0.070
	FKM	67.138	0.59	0.60	6.827±0.052
	PSO	44.065	0.64	0.66	70.280±2.694
Trace	FCM	1.702	0.57	0.53	0.2450±0.130
	FKM	2.271	0.59	0.56	0.547±0.042
	PSO	1.664	0.59	0.57	69.365±0.607
Gun point	FCM	1.580	0.50	0.50	0.060±0.029
	FKM	1.963	0.50	0.50	0.333±0.018
	PSO	1.391	0.55	0.56	14.649±0.147

It is evident from table 5 that the proposed method has obtained the best fitness function value in all datasets. Also, in all cases, the F-Measure and Precision values obtained through the proposed method is one of the best. Comparing the run time of the methods, the evolutionary techniques in general are more time consuming than FCM and FKM. The reasons for the superiority of the proposed clustering algorithm are, first, the efficiency of PSO algorithms in data clustering and second, SVD dimension reduction technique ability in time series reconstruction with a limited number of coefficients. If we use PSO algorithm for long time series clustering, the algorithm faces

with many unknowns, thus the efficiency of that will be reduced and we cannot get the expected results.

9. Conclusion

In this study, a fuzzy clustering method based on PSO algorithm was proposed for clustering time series datasets. The main advantage of the proposed method is its ability to long time series data clustering. The main difference between our method and the existing time series clustering methods based on PSO algorithm is, how to define the particles. In the proposed method, the most important singular value decomposition coefficients (SVD) of the cluster centers that are limited number of real numbers, were recognized as the main unknown clustering problem and PSO algorithm tries to determine the unknowns by minimizing the selected objective function. This reduces the number of unknowns and increases the efficiency of the algorithm. By selecting fuzzy C-means objective function as the fitness function and with considering Pearson correlation coefficients as the similarity measure, we implemented our method and two other well-known algorithms on three datasets. By comparing with the obtained results of implementing three methods, we found out the best performance in our proposed method.

References

[1] Abraham, A., Das, S. & Roy, S. (2008). Swarm intelligence algorithms for data clustering. Soft Computing for Knowledge Discovery and Data Mining, Springer, pp. 279-313.

[2] Bezdek, J. C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press.

[3] Niknam, T., Nayeripour, M. & Firouzi, B. B. (2008). Application of a New Hybrid optimization Algorithm on Cluster Analysis Data clustering, World Academy of Science, Engineering and Technology, vol. 36, pp. 598-604.

[4] Ahmadyfard, A. & Modares, H. (2008). Combining PSO and K-means to enhance data clustering, In: International symposium on telecommunications, Tehran, Iran, 2008.

[5] Hwang, J. & Huang, C. (2010). Evolutionary dynamic particle swarm optimization for data clustering, in: International Conference on Machine Learning and Cybernetics (ICMLC), Qingdao, China, pp. 3240 – 3245.

[6] Rana, S., Jasola, S. & Kumar, R. (2010). hybrid sequential approach for data clustering using K-means and particle swarm optimization algorithm, International Journal of Engineering, Science and Technology, vol. 2, no. 6, pp. 167-176.

[7] Premalatha, K. & Natarajan, A. M. (2010). Hybrid PSO and GA Models for Document Clustering, International Journal of Advances in Soft Computing and Its Applications, vol. 2, no. 3, pp. 302-320.

[8] Esmin, A. A. A. & Matwin, S. (2012). Data clustering using hybrid particle swarm optimization, in: 13th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2012), Lecture Notes in Computer Science (Springer LNCS). Springer, Heidelberg, vol. 7435, pp. 159–166.

[9] Kamel, N., Ouchen, I. & Baali, K. (2014). A Sampling PSO-K-means Algorithm for Document Clustering, Advances in Intelligent Systems and Computing, Springer International Publishing Switzerland, pp. 45-54.

[10] Kamel, N. & Gaikwad, P. (2014). Hybrid Particle Swarm Optimization (HPSO) for Data Clustering, International Journal of Computer Applications, vol. 97, pp. 1-5.

[11] Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization. , Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ:IEEE Press, pp. 1942–1948.

[12] Kennedy, J. & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, pp. 4104-4108.

[13] Elbeltagi, E., Hegazy, T. & Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms, Advanced Engineering Informatics, vol. 19, pp. 43 -53.

[14] Korn, F., Jagadish, H. V. & Faloutsos, C. (1997). Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences, In Proceedings of the ACM SIGMOD Int'l. Conference on Management of Data, pp. 289–300.

[15] Zhang, Z., Huang, K. & Tan, T. (2006). Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes, Proceedings of the 18th International Conference on Pattern Recognition, pp.1135 -1138.

[16] Junejo, I. N., Javed, O. & Shah, M. (2004). Multi Feature Path Modeling for Video Surveillance, Proc. Int\'l Conf. Pattern Recognition, vol. 2, pp.716 -719.

[17] Vlachos, M., Gunopulos, D. & Kollios, G. (2002). Discovering similar multidimensional trajectories, in ICDE, San Jose, CA, pp. 673 – 684.

[18] Berndt, D. & Clifford, J. (1994). Using Dynamic Time Warping to Find Patterns in Time Series, Proc. AAAI-94 Workshop Knowledge Discovery in Databases, pp. 359-370.

[19] Zhang, H., Ho, T. B., Zhang, Y. & Lin, M. S. (2006). Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform, Informatica, pp. 305-319.

[20] Keogh, E. & Folias, T. (2002), The UCR Time Series Data Mining Archive, Available: http://www.cs.ucr.edu/»eamonn/TSDMA/index.html,2002

[21] Chis, M., Banerjee, S. & Hassanien, A. E. (2008). Clustering Time Series Data: An Evolutionary Approach, Foundations of Computational Intelligence, Springer , vol. 206, pp.193-207.

[22] Keogh, E. & Smyth, P. (1997). A probabilistic approach to fast pattern matching in time series databases, in: Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining, pp. 24–20.

نشریه هوش مصنوعی و داده کاوی

دانشگاه شاهرود

ارائه‌ی یک روش خوشه‌بندی فازی سری‌های زمانی با استفاده از الگوریتم انبوه ذرات

زاهده ایزکیان* و محمد سعدی مسگری

دانشکده‌ی مهندسی ژئودزی و ژئوماتیک، دانشگاه خواجه نصیر طوسی، تهران، ایران.

ارسال ۲۰۱۴/۰۶/۱۴؛ پذیرش ۲۰۱۵/۰۵/۰۹

چکیده:

با پیشرفت روزافزون تکنولوژی‌های جمع آوری اطلاعات و امکان دسترسی به حجم عظیمی از داده همواره نیازمند روش‌هایی برای تجزیه و تحلیل این حجم داده خام و استخراج اطلاعات مفید از آن می‌باشیم که داده کاوی یکی از مهمترین روش‌های حل این مسئله است. خوشه بندی داده نیز یکی از پرکاربردترین زمینه‌های داده‌کاوی محسوب شده و مورد توجه بسیاری از محققین در علوم مختلف قرار گرفته است. در سال‌های اخیر مسئله‌ی خوشه-بندی سری‌های زمانی بدلیل کاربردهای مختلف علاقه‌ی محققین را به خود جلب کرده و الگوریتم‌های بسیاری در این زمینه پیشنهاد داده شده است. امروزه از الگوریتم‌های هوش جمعی به عنوان خانواده‌ای از الگوریتم‌های الهام گرفته از طبیعت، در حل بعضی از مسائل خوشه‌بندی و تشخیص الگو استفاده می‌گردد. در این تحقیق از الگوریتم انبوه ذرات که در گروه الگوریتم‌های خوشه‌بندی مبتنی بر هوش جمعی می‌باشد، برای خوشه‌بندی سری-های زمانی استفاده شد و ضریب همبستگی پیرسون که یکی از پرکاربردترین توابع فاصله می‌باشد، به عنوان معیارشباهت میان داده‌ها در نظر گرفته شد. روش پیشنهادی توانایی یافتن مراکز خوشه‌های نزدیک به بهینه را در یک مجموعه داده دارا می‌باشد. در این روش برای کاهش ابعاد فضای جستجو و افزایش کارایی الگوریتم بهینه سازی استفاده شده، از یکی از روش‌های کاهش بعد داده به نام روش تجزیه‌ی مقدار تکین (Singular Value Decomposition) استفاده شد. روش پیشنهاد داده شده روی سه سری مجموعه داده‌ی شناخته شده پیاده‌سازی شده و با دو روش خوشه‌بندی fuzzy C-means و fuzzy K-medoids مقایسه گردید. نتایج حاصل کارایی روش پیشنهادی را نشان می‌دهد.

کلمات کلیدی: خوشه‌بندی، سری‌های زمانی، الگوریتم انبوه ذرات، روش تجزیه‌ی مقدار تکین، ضریب همبستگی پیرسون.