



AN EFFICIENT COST FUNCTION FOR IMPERIALIST COMPETITIVE ALGORITHM TO FIND BEST CLUSTERS

¹MOJGAN GHANAVATI, ²MOHAMAD REZA GHOLAMIAN, ³BEHROUZ MINAEI,
⁴MEHRAN DAVOUDI

² Professor, Iran University of Science and Technology, Department of Industrial engineering, Tehran, IRAN

³ Professor, Iran University of Science and Technology, Department of Computer engineering, Tehran, IRAN

^{1,4}MS Student, Iran University of Science and Technology, Department of Industrial engineering, Tehran, IRAN

E-mail: ¹Mojgan_ghanavaty@yahoo.com, ²gholamian@iust.ac.ir

ABSTRACT

Cluster analysis is one of the attractive data mining techniques that have been used in many fields. One of the popular types of clustering algorithms is the center based clustering algorithm. K-means used as a popular clustering method due to its simplicity and high speed in clustering large datasets. However, K-means has two shortcomings. K-means is dependent on the initial state and convergence to local optima in some of the large problems. In order to these shortcomings, in an unsupervised clustering the number of clusters needs to be fixed by a human analyst too. In order to overcome local optima problem and for determining the number of clusters, lots of studies done in clustering. In this paper we use a new search heuristic called "Imperialist Competitive Algorithm"¹ to find the best clusters with best numbers of clustering. In this algorithm we assume each clustering solution with special clusters number as a country and use a new cost function to calculate the clustering cost in each step. We compared proposed algorithm with other heuristics algorithm in clustering, such as traditional K-means, IGKA, CSO and GA-PSO by implementing them on several well-known datasets. Our findings show that the proposed algorithm works better than the others according to cost function and standard deviation.

Keywords: *Clustering; Meta-heuristic; K-means; Imperialist Competitive Algorithm*

1. INTRODUCTION

According to the large amount of data in the world, we need new data analysis and extracting information techniques. Therefore, new optimization algorithms are being presented every day. One of the most usable techniques of data analysis is clustering. Clustering can be considered the most important unsupervised learning problem. So, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering that is the subject of active research in several fields such as data mining, applied in a large variety of applications, like image segmentation, market segmentation, etc. Clustering algorithms can be classified as hierarchical clustering, partition-based

clustering, exclusive clustering, and overlapping clustering.

One of the most used classes of data clustering algorithms is the center based clustering algorithm. K-means is one of the simplest unsupervised learning algorithms that follow a simple and easy way to classify a given data set through a certain number of clusters fixed a priori [1]. However, K-means has two shortcomings: dependency on the initial state and convergence to local optima [2,3,4,5] and global solutions of large problems cannot found with reasonable amount of computation effort. In order to overcome local optima problem and for finding best number of clusters lots of studies done in clustering.

Kuo, et al., [6] proposed a novel clustering method, ant K-means² algorithm. AK algorithm modifies the



K-means as locating the objects in a cluster with the probability, which updated by the pheromone, while the rule of updating pheromone is according to total within cluster variance³.

Chu., S.C., et al., [7] used a new optimization algorithm that simulated the behavior of cats and called Cat Swarm Optimization⁴ to solve the clustering problem and showed that it has a higher performance than K-means, PSO and weighted-PSO algorithms.

Santosa, et al., [8] changed the CSO formula and tested on four variant datasets and compared with PSO and K-means. The results showed that CSO-based method might improve the performance of traditional CSO algorithm and generally is accurate enough.

Nguyen, C.D., Cios, K.J., [9] proposed a novel hybrid clustering algorithm called GAKREM⁵ for clustering analysis. It uses the best properties of K-means and EM⁶ algorithms and omits their shortcomings such as complicated computations, convergence to local optima and necessity to determine the number of clusters.

Kuo, et al [10] used adaptive resonance theory⁷ neural network to determine the number of clusters and an initial solution and then used the genetic K-means algorithm⁸ to find last solution to analysis of web browsing paths in electronic commerce. In order to verify the proposed method, data from a Monte Carlo Simulation are used. The simulation results show that the ART2 +GKA is significantly better than the ART2 + K-means, both for mean within cluster variations and misclassification rate. These results also show that, based on the mean within-cluster variations, ART2 +GKA is much more effective.

Due, et al. [11] integrate the K-means and particle-pair optimizer⁹ that is a variation on the traditional particle swarm optimization¹⁰ algorithm and is stochastic particle-pair based optimization technique and showed that PK-means is generally more accurate than K-means and Fuzzy K-means and outperformed these methods with fast convergence rate and low computation load. PK-means also is less sensitive to the initial randomly selected cluster centroids.

Niknam, T., Amiri, B., [12] presented a new hybrid evolutionary algorithm to solve nonlinear partitional clustering problem. The proposed hybrid evolutionary algorithm is the combination of fuzzy adaptive particle swarm optimization¹¹, ant colony optimization¹² and k-means algorithms, called FAPSO-ACO-K, which can find better cluster partition. The performance of the proposed algorithm is evaluated through several benchmark

data sets. The simulation results showed that the performance of the proposed algorithm is better than other algorithms such as PSO, ACO, simulated annealing¹³, combination of PSO and SA¹⁴ and k-means for partitional clustering problem.

Fathian, et al., [13] proposed application of honeybee mating optimization in clustering¹⁵ and compared it with other heuristics algorithm in clustering, such as GA, SA, TS, and ACO, by implementing them on several well-known datasets.

Finding showed that the proposed algorithm works better than the others and Amiri and Fathian, [14] proposed a two-stage method, which first uses Self Organizing Feature Maps¹⁶ neural network to determine the number of clusters and cluster centroids, then uses honey bee mating optimization algorithm based on K-means algorithm to find the final solution. The results of simulated data via a Monte Carlo study show that the proposed method outperforms two other methods, SOM followed by K-means and SOM followed by GAK, based on both within-cluster variations¹⁷ and the number of misclassification.

Most of the current evolutionary algorithms, such as genetic algorithm and ant colony are computer simulation of natural processes such as natural evolution and behavior of animals. In this paper we want to use imperialist competitive algorithm that uses imperialism and imperialistic competition, socio-political evolution processes, as source of inspiration developed by Atashpaz-Gargari, E., Lucas, C., [15].

Atashpaz-Gargary, et al., [16] applied imperialist competitive algorithm to tune the parameters of a multivariable PID controller for a typical distillation column process. Simulation results showed that the proposed controller tuning approach can be easily and successfully applied to the problem of designing MIMO controller for control processes. As a result not only was the controlled process able to significantly reduce the coupling effect, but also the response speed was significantly increased. Also the results showed that ICA had a higher convergence rate than GA, reaching to a better solution.

Biabangard-Oskouyi, et al., [17] employed ICA for material properties evaluation from indentation test curve. Results obtained from applying the proposed method to a variety of sharp indentation test responses, indicate the good ability of proposed method for interpreting the indentation test responses for material properties determination.

Nazari-Shirkouhi, S., et al., [18] applied imperialist competitive algorithm to solve the integrated product mix-outsourcing optimization problem in



manufacturing enterprise. Also, the results obtained from ICA are compared with the results of TOC and Standard Accounting approaches that generally are used to optimize such problems and are inefficient especially in large problems.

Rajabioun, et al., [19] applied imperialist competitive algorithm to find Nash Equilibrium points of nonlinear non-cooperative games and suggest that the proposed method can also be used as an alternative approach to solve multi-objective optimization problems. The effectiveness of the proposed method, in comparison to Genetic Algorithm, is proven through several static and dynamic example games and also multi-objective problems.

Niknam et al. [20] presented an efficient hybrid evolutionary optimization algorithm based on combining Modify Imperialist Competitive Algorithm¹⁸ and K-means, which is called K-MICA, for optimum clustering. The new Hybrid K-ICA algorithm is tested on several data sets and its performance is compared with ACO, PSO, Simulated Annealing, Genetic Algorithm, Tabu Search, Honey Bee Mating Optimization and K-means. The simulation results show that the proposed evolutionary optimization algorithm is robust and suitable for handling data clustering.

This paper uses traditional imperialist competitive algorithm with new objective function to find the optimum clustering. The paper organized as follow: in Section 2 we discussed cluster analysis problems. Section 3 introduces imperialist competitive nature and application of imperialist competitive algorithm in clustering, and then in Section 4 experimental result of proposed clustering algorithm in comparison with other heuristics clustering algorithms showed.

2. CLUSTERING

Clustering is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Cluster analysis can be used to discover structures in data without providing an explanation [21]. The standard clustering process consists of the following steps:

- (1) Data preparation and attribute selection,
- (2) Similarity measure selection,
- (3) Algorithm and parameter selection,
- (4) Cluster analysis and
- (5) Validation.

In this study, we will use Euclidian metric as a distance metric (Esq. (1)) and will use an efficient approach for computing silhouette coefficient as a validation and for finding the best number of clusters in a data set ([1]).

$$(1) \quad d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$$

3. APPLICATION OF IMPERIALIST COMPETITIVE ALGORITHM IN CLUSTERING WITH NEW COST FUNCTION.

3.1. IMPERIALIST COMPETITIVE ALGORITHM

ICA is a novel global search heuristic that uses imperialism and imperialistic competition process. This algorithm starts with some initial countries. Some of the best countries are selected to be the imperialist and all the other countries form the colonies of these imperialists. The colonies are divided among the mentioned imperialists based on their power.

After dividing all colonies among imperialists and creating the initial empires, these colonies start moving toward their relevant imperialist. This movement is a simple model of assimilation policy that was pursued by some imperialists.

Figure 2 shows the movement of a colony towards the imperialist. In this movement, θ and x are random numbers with uniform distribution as illustrated in equation (2) and d is the distance between colony and the imperialist.

$$(2) \quad x \sim U(0, \beta \times d), \theta \sim U(-\gamma, \gamma),$$

Where β and γ are arbitrary numbers that modify the area that colonies randomly search around the imperialist. In our implementation β and γ are 2 and 0.5 (rad), respectively.

The total power of an empire is defined by the power of imperialist plus a percentage of the mean power of its colonies. In imperialistic competition, all empires try to take possession of the colonies of other empires and control them.

This competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. This competition is modeled by just picking some of the weakest colonies of the weakest empires and making a competition among all empires to possess these colonies. Figure 2 shows a picture of the modeled imperialistic competition. Based on their total power, in this competition, each of empires will

have a likelihood of taking possession of the mentioned colonies. The more powerful an empire is, the more likely it will possess these colonies. In other words these colonies will not be certainly possessed by the most powerful empires, but these

empires will be more likely to possess them. Any empire that is not able to succeed in imperialist competition and can't increase its power or at least prevent decreasing its power will be eliminated.

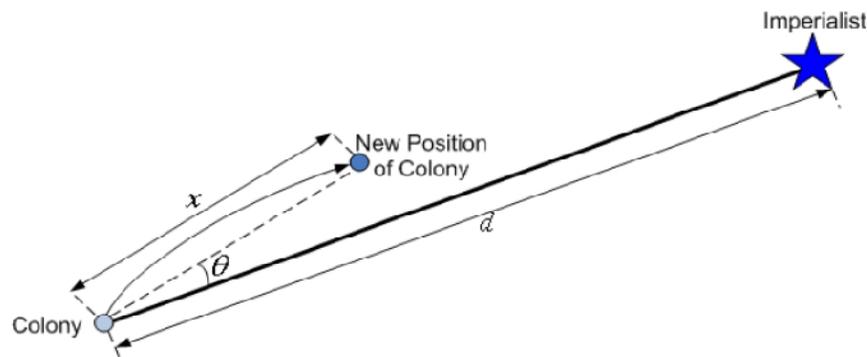


Figure 1. Motion of colonies toward their relevant Imperialist

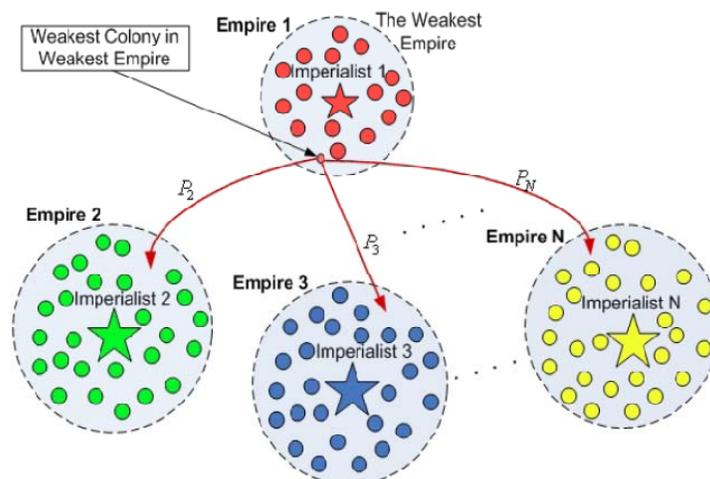


Figure 2. Imperialistic competition: The more powerful an empire is, the more likely it will possess the weakest colony of weakest empire

The imperialistic competition will gradually result in an increase in the power of great empires and a decrease in the power of weaker ones. Weak empires will lose their power gradually and ultimately they will collapse. The movement of colonies toward their relevant imperialists along with competition among empires and also collapse mechanism will hopefully cause all the countries to converge to a state in which there exist just one empire in the world and all the other countries are its colonies. In this ideal new world colonies have the same position and power as the imperialist.

The pseudo code of Imperialist competitive algorithm is as follows:

1) Select some random points on the function and initialize the empires.

2) Move the colonies toward their relevant imperialist (Assimilation).

3) Randomly change the position of some colonies (Revolution).

4) If there is a colony in an empire which has lower cost than the imperialist, exchange the positions of that colony and the imperialist.

5) Unite the similar empires.

6) Compute the total cost of all empires.

7) Pick the weakest colony (colonies) from the weakest empires and give it (them) to one of the empires (Imperialistic competition).

8) Eliminate the powerless empires.

9) If stop conditions satisfied, stop, if not go to 2.



3.2. APPLICATION OF ICA TO FIND BEST CLUSTERS WITH NEW COST FUNCTION

In this session we use ICA to find the best clusters. We describe the steps of the proposed algorithm in details.

Step 1: Creation of countries

We form an array of variable values to be optimized. In the GA, this array is called "chromosome", but in ICA the term "country" is used for this array. In this paper we form an $1*(N*K+1)$ array as a country where N is the feature numbers of each data point and K is the max number of clusters. First item of array is filled by number of cluster and rests of them are filled by random numbers between 0 and 1 that each N number of them shows a cluster center. This array is defined as following:

$$\text{Country} = [dp_1, dp_1, \dots, dp_{N*K}]$$

Actually each country defines one clustering solution.

Step 2: Creation of initial empires

To create initial imperialist, we select most powerful countries. So we calculate cost of each country with a cost function. In this paper we use k-means to create each clustering solution and use a new cost function to calculate cost of each solution. We give first item of country as clusters number and $N*\text{Country}(1)$ items of that as initial cluster centers to K-means, where content of country(1) shows clusters number of that clustering solution. After creating clusters, we calculate cost of solution for each data point according equation (3),

$$(3) \quad \text{Cost} = (k/E_1) * (1 - D_k/E_k)$$

where k is the number of clusters and E_k is sum of internal compactness of each cluster and defines based on equation (4). To calculate value of D_k at first we calculate d_i that is minimum distance of each data point to points of other clusters; Minimum of d_i s of all of data points will be D_k according equation (5). To implement this equation, we used method that introduced in [1] because of its speed. E_1 is a constant value that is calculated for each data set and shows the dispersal of each dataset.

$$(4) \quad E_k = \sum_{i=1}^k \sum_{j \in cl_i} ||x_j - c_i||^2$$

Where cl_i is the i th cluster set that c_i is the center of that and x_j s are data points that belong to cluster i .

$$(5) \quad D_k = \min_{i=1}^n (d_i)$$

E_k decrease by increasing k, so by increasing k, this factor decrease cost function. To prevent to select big k, we use k in cost function and use E_1 to handle the value of E_k .

Countries with a lower cost functions are chosen to create initial imperialists. Remains of initial countries will be the colonies which belong to empires. To form the initial empires, the colonies are divided among imperialists based on their power. The initial number of colonies of an empire should be directly proportionate to its power. Therefore to define initial number of colonies of an empire, we use equation (6)

$$(6) \quad NC_n = \text{round}(1/(\text{Cost}(\text{empire}(n))) * N_{\text{col}})$$

Where NC_n is the number of colonies which belong to n th empire and N_{col} is the total numbers of colonies. To divide the colonies between empires, NC_n of the colonies are randomly chosen and given to the n th imperialist.

Step 3: Assimilation; Movement of colonies toward the imperialist

In assimilation, each colony moves on the line that connects the colony and its imperialist by x units, which x is a random variable with uniform distribution as illustrated in equation (2).

Step 4: Revolution

After assimilating all of the colonies by imperialists in each empire, revolution takes place in some of the countries. This revolution includes changing in number of clusters and position of data points.

Step 5: New cost evaluation

After assimilation and revolution, the power of each colony is calculated in its new position. Some of the colonies in each empire might have reached to better positions than the imperialist itself. The total power of empires is calculated as in equation (7)

$$(7) \quad \text{Power}(\text{empire}(i)) = 1 / (\text{Cost}(\text{empire}(i)) + \sum 1 / (\text{Cost}(\text{empire}(i). \text{Colonies})))$$



Step 6: Imperialistic Competition

In this step imperialistic competition starts and a colony of poor empires are possessed by another one. The more powerful empires have the more probability to get colonies. The continuation of these processes converge the algorithm to reach the best clustering solution with best fitness function.

699 instances with 10 numeric attributes in data set. Each instance has one of benign or malignant classes. There are 16 missing attribute values in data set.

To evaluate the performance of the application of ICA algorithm in clustering, we have compared it with other stochastic clustering algorithms including the K-means, Cat Swarm Optimization, PSO-GA and Improved GKA algorithms [23] with new cost function. The performance of stochastic algorithms is dependent on the generation of initial solutions. Therefore, we ran all datasets 10 times with three mentioned algorithms.

The comparison of results for each dataset based on the solution found in 10 distinct runs of each algorithm, the average number of evaluations required and the convergence processing time taken to attain the best solution. The solution quality is also given in terms of the average and worst values of the clustering metric after 10 different runs for each of the five algorithms. Tables 1 show these results on Iris dataset. Results show that ICA works better than other clustering algorithms according to the value of their cost function, but its speed is lower than other except GA-PSO. Tables 2 show the results of these algorithms on Wine dataset and Tables 3 show their result on Winston Breast Cancer dataset. All the results show that we can get a optimum solution by ICA algorithm. You can see the parameters of each algorithm in Tables 4 and can see the min and mean cost of ICA in one run of that on Iris, Wine and Wisconsin Breast Cancer in Figures 1, 2 and 3.

4. EXPERIMENTAL RESULTS

In this section, we present a set of experiments that shows the power of our algorithm. We have coded our algorithm with Matlab 7.6 and run it on three different datasets. The datasets are iris, wine and breast cancer datasets taken from UCI repository [22]. We normalized datasets to 0 and 1, to use in our work.

DataSet1 is the Iris data set, which is most well-known data set to evaluate clustering algorithm. This data set contains three classes with 50 instances in each class, where each class refers to a type of iris plant. There are 150 instances with four numeric attributes in iris data set and there isn't any missing value. The attributes of the iris data set are sepal length in cm, sepal width in cm, petal length in cm and petal width in cm.

DataSet2 is the wine data set, which also taken from UCI repository. These data are the results of a chemical analysis of wines grown in the same region in Italy. This data set contains 178 instances with 13 numeric attributes that all of them are continuous. There is no missing attribute value.

DataSet3 is the Wisconsin Breast Cancer Database which also taken from UCI repository. There are

Table 1 Result obtained by the five algorithms for 10 different runs on Iris dataset

Method	Cost Function value			CPU Time	Standard deviation	SSE (Average)
	Min	Average	Max			
K-means	0.3414	1.3673	1.4758	0.10	0.6259	2.8275
IGKA	0.6167	0.9515	1.6799	2.13	0.5436	2.9156
CSO	29.6329	32.5334	34.7322	1.3	2.5576	25.2013
GA-PSO	0.3113	0.3315	0.3726	208.9477	0.0312	6.1179
ICA	0.1196	0.1196	0.1196	34.81	0	2.8253

Table 2 Result obtained by the five algorithms for 10 different runs on Wine dataset

Method	Cost Function value			CPU Time	Standard deviation	SSE (Average)
	Min	Average	Max			
K-means	3.7975	5.8879	7.9897	0.26	2.0961	27.3374
IGKA	0.9518	1.0683	1.5992	5.88	0.3450	27.0872
CSO	76.9144	84.7782	93.1911	1.5	8.1398	110.6672
GA-PSO	0.7876	0.8199	0.8912	220	0.0530	35.2913
ICA	0.2062	0.2361	0.2448	130.01	0.0202	27.3203



Table 3 Result obtained by the five algorithms for 10 different runs on Wisconsin Breast Cancer dataset

Method	Cost Function value			CPU Time	Standard deviation	SSE (Average)
	Min	Average	Max			
K-means	1.2537	2.2577	3.2337	2.5	0.9900	115.4132
IGKA	0.4047	0.4708	0.5034	1.8	0.0502	109.5587
CSO	323.9458 348.0002	335.4911		3.2	12.0304	334.7793
GA-PSO	0.4376	0.4997	0.5132	~3000	0.0403	272.5208
ICA	0.0956	0.1399	0.1658	~1000	0.0355	91.9343

Table 4 Values of parameters of each of five algorithms.

IGKA		CSO		GA-PSO		ICA	
Parameter	value	Parameter	value	Parameter	value	Parameter	value
PopSize	40	Copy	50	Popsize	30	#countries	40
MAXgen	50	SRD	0.2	KeepPercent	0.5	#Imperialists	5
MutationRate	0.005	Const1	2	CrossoverRate	0.7	Revolution Rate	0.3
CrossoverRate	0.7	R1	[0,1]	SelectionMode	1	γ	0.5
#iteration	40	Velmax	0.9	#iteration	40	β	3
		#iteration	40			#iterations	40

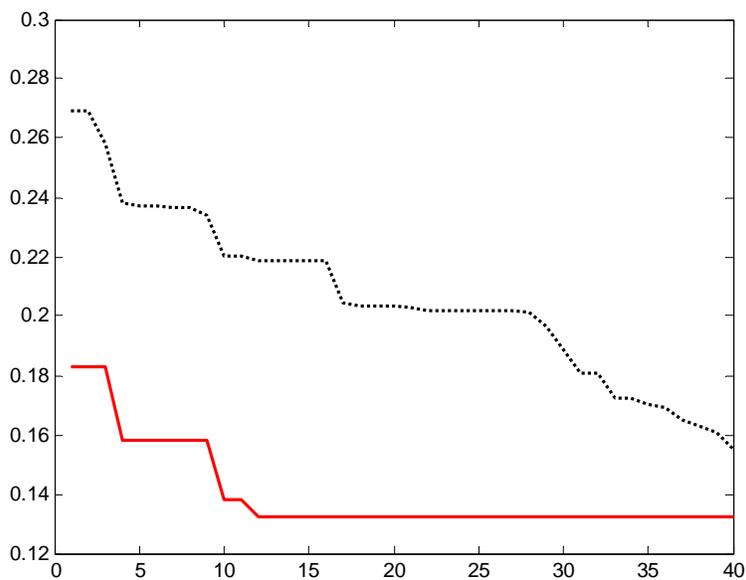


Figure 1. Min and Means Cost of ICA on Iris dataset

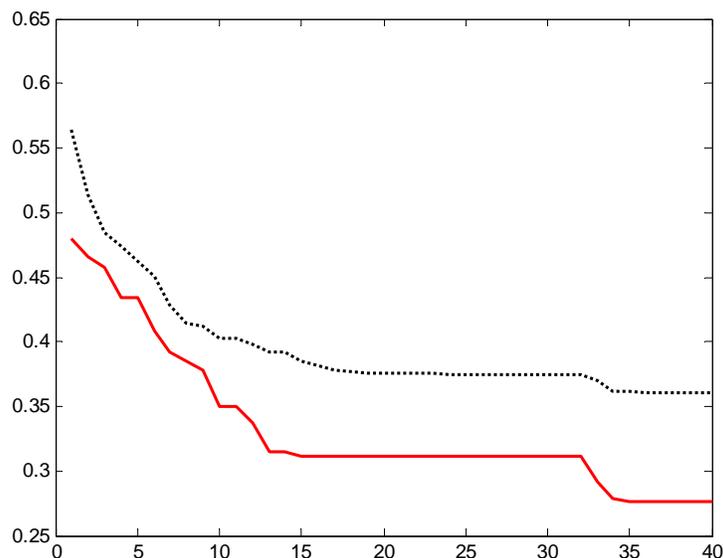


Figure 2. Min and Means Cost of ICA on Wine dataset

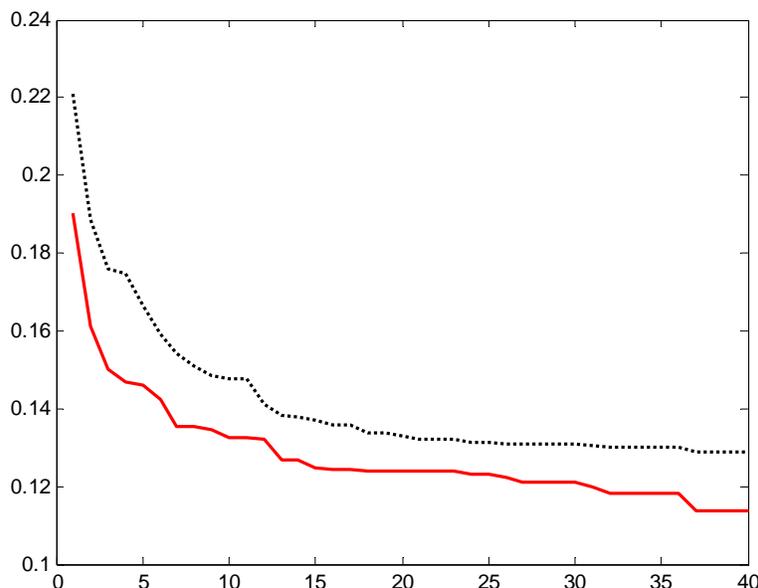


Figure 3. Min and means cost of ICA on Wisconsin Breast Cancer dataset

5. CONCLUSION

In summary, we developed imperialist competition algorithm to solve clustering problems with an efficient cost function in this paper.

To apply ICA for clustering we don't need to determine the number of clusters. ICA finds the

best K itself. To evaluate the performance of the ICA, it compared with other stochastic algorithms such as K-means, IGKM, CSO and GA-PSO. The algorithm implemented and tested on several real datasets.



REFERENCES

- [1] T., Niknam, et al., An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering, *Engineering Applications of Artificial Intelligence* (2010).
- [2] M., Laszlo, S., Mukherjee, A genetic algorithm that exchanges neighboring centers for k-means clustering, *Pattern Recognition Letters*, 28 (2007) 2359–2366.
- [3] T., Niknam, et al., An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering, *Journal of Zhejiang University SCIENCE*, (2009) 512-519.
- [4] C.C., Hung, L., Wan, Hybridization of Particle Swarm Optimization with the K-Means Algorithm for Image Classification (2009).
- [5] Z.M., Nopiah, et al, A Weighted Genetic Algorithm Based Method for Clustering of Heteroscaled Datasets, *International Conference on Signal Processing Systems*(2009).
- [6] Y-T., Kuo, et al., A hybridized approach to data clustering, *Expert Systems with Applications*, 34 (2008) 1754–1762.
- [7] S.C., Chu., et al., *Cat Swarm Optimization*, Springer-Verlag Berlin Heidelberg, (2006) 854 – 858.
- [8] B., Santosa, M.K., Ningrum, Cat Swarm Optimization for Clustering, *International Conference of Soft Computing and Pattern Recognition* (2009).
- [9] C.D., Nguyen, K.J., Cios, GAKREM: A novel hybrid clustering algorithm, *Information Sciences*, 178 (2008) 4205–4227.
- [10] R.J., Kuo, et al., Integration of ART2 neural network and genetic K-means algorithm for analyzing Web browsing paths in electronic commerce, *Decision Support Systems*, 40 (2005) 355– 374.
- [11] Z., Du, et al., PK-means: A new algorithm for gene clustering, *Computational Biology and Chemistry*, 32 (2008) 243–247.
- [12] T., Niknam, B., Amiri, An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis, *Applied Soft Computing*, (2010) 183–197.
- [13] M., Fathian, et al., Application of honey-bee mating optimization algorithm on clustering, *Applied Mathematics and Computation*, 190 (2007) 1502–1513.
- [14] Amiri, B., Fathian, M., integration of self organizing feature maps and honey bee mating optimization algorithm for market segmentation, *journal of theoretical and applied information technology*, (2007).
- [15] E., Atashpaz-Gargari, C., Lucas, Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistics, *IEEE Congress on Evolutionary Computation*, (2007), 4661 – 4667.
- [16] E., Atashpaz Gargary, et al., Colonial competitive algorithm A novel approach for PID controller design in MIMO distillation column process, *International Journal of Intelligent Computing and Cybernetics*, 1 (2008) 337-355.
- [17] E., Biabangard-Oskouyi, et al., Application of Imperialist Competitive Algorithm for Materials Property Characterization from Sharp Indentation Test, to be appeared in *International Journal of Engineering Simulation* (2009).
- [18] S., Nazari-Shirkouhi, et al., Solving the integrated product mix-outsourcing problem using the Imperialist Competitive Algorithm, *Expert Systems with Applications*, (2010) 7615–7626.
- [19] R., Rajabioun, E., Atashpas-Gargari, C., Lucas, Colonial Competitive Algorithm as a Tool for Nash Equilibrium Point Achievement, *Lecture Notes In Computer Science*, Vol. 5073, Proc. of the Intl. conf. on Computational Science and Its Applications, Part II, 680-695.
- [20] Niknam, T., et al., An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering, *Engineering Applications of Artificial Intelligence*, (2010).
- [21] Kao, Y-T., Zahara, E., Kao, I-W., “A hybridized approach to data clustering”, *Expert Systems with Applications*, 34 (2008) 1754–1762.
- [22] C.L., Blake, C.J., Merz, UCI repository of machine learning databases. Available from: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.
- [23] Guo, H.X., et al.,” An Improved Genetic k-means Algorithm for Optimal Clustering”, *Sixth IEEE International Conference on Data Mining*, (2006).



-
- ¹ ICA
 - ² AK
 - ³ TWCV
 - ⁴ CSO
 - ⁵ Genetic Algorithm K-means Logarithmic Regression Expectation Maximization
 - ⁶ Expectation Maximization
 - ⁷ ART2
 - ⁸ GKA
 - ⁹ PPO
 - ¹⁰ PSO
 - ¹¹ FAPSO
 - ¹² ACO
 - ¹³ SA
 - ¹⁴ PSO-SA
 - ¹⁵ HBK-means
 - ¹⁶ SOM
 - ¹⁷ SSW
 - ¹⁸ MICA