5th SASTech
Iran | Mashhad
May 12 - 17 | 2011
5th Symposium on Advances in Science & Technology

Hybrid Imperialist Competitive Algorithm and Dynamic Validity Index to find the best clusters

Mojgan Ghanavati[1]
mojgan.ghanavati@gmail.com

Mohammad Reza Gholamian[2]
gholamian@iust.ac.ir

Behrouz Minaei[3]
b_minaei@iust.ac.ir

Mehran Davoudi[4]
mehran.davoudi@gmail.com

Photograph of Presenter

[1,2,3,4] Iran University of science and technology
Paper Reference Number: 16
Name of the Presenter: Mehran Davoudi

Abstract

Cluster analysis is one of attractive data mining technique that use 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 combine a new search heuristic called "Imperialist Competitive Algorithm" with "Dynamic Validity Index (DVIndex)" to find the best clusters. In this algorithm, we assume each clustering solution as a country and use we use DVIndex as an efficient method to find number of clusters for calculating the clustering cost in each step. We compared proposed algorithm with other heuristics algorithm in clustering, such as traditional K-means, CSO, GKA and PSO-GA, by implementing them on several well-known datasets. Our findings show that the proposed algorithm works better than the others.

Key words: Clustering, Meta-heuristic, K-means, Imperialist Competitive, DVIndex

1. Introduction

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.

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 (Niknam, T., et al., 2010). However, K-means has two shortcomings: dependency on the initial state and convergence to local optima and global solutions of large problems cannot found with reasonable amount of computation effort. In order to overcome these drawbacks, lots of studies done in clustering.

Nguyen, C.D., Cios, K.J., 2008 proposed a novel hybrid clustering algorithm called GAKREM[1] for clustering analysis. It uses the best properties of K-means and EM[2] algorithms and omits their shortcomings such as complicated computations, convergence to local optima and necessity to determine the number of clusters.

Due, et al. 2008 integrate the K-means and particle-pair optimizer that is a variation on the traditional particle swarm optimization 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.

Niknam, T., Amiri, B., 2010 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 optimization, ant colony optimization\ and k-means algorithms, called FAPSO-ACO–K, which can find better cluster partition.

Kao, et al., 2008 proposed a hybrid technique based on combining the K-means algorithm, Nelder–Mead simplex search, and particle swarm optimization, called K–NM–PSO. The K–NM–PSO searches for cluster centers of an arbitrary data set as does the K-means algorithm, but it can effectively and efficiently find the global optima.

Zhang, et al., 2010 presented an artificial bee colony based clustering algorithm to optimally partition N objects into K clusters and has used Deb's rules to direct the search direction of each candidate. This algorithm has been tested on several well-known real datasets and compared with other popular heuristics algorithm in clustering, such as GA, SA, TS, ACO and K–NM–PSO algorithm. The computational simulations reveal very encouraging results in terms of the quality of solution and the processing time required.

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., 2007. This new algorithm has been used in many fields (Atashpaz Gargary, et al., 2008; Biabangard-Oskouyi, et al, 2009; Rajabioun, et al 2008; Mahmoudi, et al 2008).

Niknam et al. 2010 presented an efficient hybrid evolutionary optimization algorithm based on combining Modify Imperialist Competitive Algorithm 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

[1]Genetic Algorithm K-means Logarithmic Regression Expectation Maximization

[2] Expectation Maximization

results show that the proposed evolutionary optimization algorithm is robust and suitable for handling data clustering.
This paper integrates Dynamic Validity Index with traditional imperialist competitive algorithm and uses new objective function to find the optimum clustering.

2. Imperialist Competitive Algorithm

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.
Fig. 1 shows the movement of a colony towards the imperialist. In this movement, θ and x are random numbers with uniform distribution as illustrated in Eq. (2) and d is the distance between colony and the imperialist.

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

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.

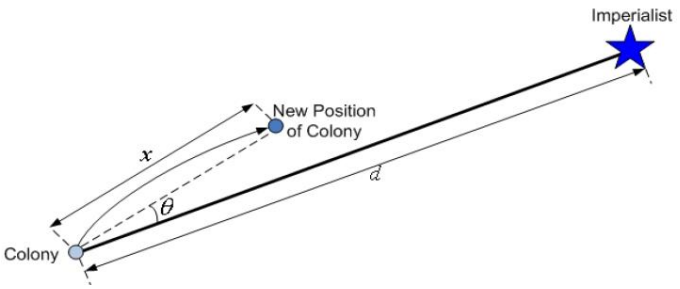

Fig 1. Motion of colonies toward their relevant Imperialist

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.

3. Dynamic validity index

In order to obtain the optimal number of clusters, we need a measure that usually known as the cluster validity index. A validity index shows the power of clustering algorithm in partitioning a given data set. Shen proposed a new dynamic validity index (DVIndex), that has omitted disadvantages of other validity indexes such as Dunn index (Ray S., Turi, R.H.,

(1999)) that are sensitive to the noise in data sets and are not so accurate in defining minimal and maximal inter distance.
DVIndex is described according Eq. 3, where IntraRatio used as the overall compactness of clusters and InterRatio used as the overall separateness of clusters.

$$\text{DVIndex} = min_k\{IntraRatio(k) + \gamma^* IntraRatio(k)\},\ k = 1, \dots, K \tag{3}$$

Where

$$\text{IntraRatio(k)} = \frac{Intra(k)}{MaxIntra}\ ;\ \text{InterRatio(k)} = \frac{Inter(k)}{MaxInter} \tag{4}$$

$$\text{Intra(k)} = \frac{1}{N}\sum_{i=1}^{k}\sum_{x\in C_i}\|x - z_i\|^2\ ;\ \text{MaxIntra} = max_i\big(Intra(i)\big),\ i=1, \dots, K \tag{5}$$

$$\text{Inter(k)}=\frac{Max_{i,j}(\|z_i-z_j\|^2)}{Min_{i\neq j}(\|z_i-z_j\|^2)}\sum_{i=1}^{k}\left(\frac{1}{\sum_{j=1}^{k}(\|z_i-z_j\|^2)}\right)\ ;\ \text{MaxInter} = max_i\big(Inter(i)\big),\ i=1, \dots, K \tag{6}$$

Where N is the number of data points, K is an upper bound that is predefined for number of clusters, z_i is the center of cluster C_i and γ is a parameter that is used to balance the importance between IntraRatio and InterRatio terms. If there isn't noise in raw data, this parameter is set to 1 ($\gamma = 1$). If there exists some noise in the data, we should decrease effect of such noise by adjusting the parameter γ less than 1 and in some cases that the cluster compactness is more important than cluster separateness, we should set parameter γ greater than 1.
In general, the relation of DVIndex and number of clusters (k) is described as DVIndex=F(k). It means that when the value of DVIndex becomes minimum, value of k would be optimum.

3. Hybrid DVI-ICA-K-Means

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+1 array as a country where N is the number of data points. First item of array is filled by number of cluster and rest of them are filled by random numbers between 0 and 1 that show the membership degree of each data point to its relative cluster. This array is defined as following:

$$\text{Country= } [K, dp_1, dp_1, \dots, dp_n] \tag{7}$$

Actually each country defines one clustering solution. We determine the bounds of clusters by an 1*K-1 array that is defined as following:

$$\text{Bounds}= [1/k, 2/k \ldots n/k] \quad (8)$$

We assign each data point to the clusters according its membership degree. For example, we assign data point i to the first cluster when it's membership degree is less than 1/k.
First item of country array determines the number of clusters. First we used dynamic validity index to determine a primary cluster number (P). Then we choose a bound for cluster numbers by decreasing and increasing 2 from P. First item of country will be a random number between P-2 and P+2.

Step 2: Creation of initial empires
To create initial imperialist, we should calculate cost of each country with a cost function. In this paper we use dynamic validity index that is described in section 3.2.

Step 3: Assimilation; Movement of colonies toward the imperialist
In this step colonies start moving toward their relevant imperialist state which is based on assimilation policy. Fig.1 shows the movement of a colony towards the imperialist. In this movement, θ and x are random numbers with uniform distribution as illustrated in Eq. (2) and d is the distance between colony and the imperialist.

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 cost of empires is calculated as in equation (9)

$$\text{Cost(empire(i))}=\text{DVI(empire(i))}+\sum \text{DVI(empire(i).Colonies)} \quad (9)$$

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 global minimum of the cost function.
Step 7: K-means
In final step we run K-means and use most powerful imperialist as a primary centers for it and then compare the result of K-means with this imperialist and choose best one as a result of algorithm.

4. Results and Analysis
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. We normalized datasets to

0 and 1, to use in our work. Before we insert the data into the algorithms we have normalized the data according to:

$$X=\frac{X-min(X)}{max(X)-min(X)} \quad (10)$$

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 (Guo, H.X., et al., 2006).
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 it's 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 Figs 2~ 4.

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 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	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 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	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	335.4911	348.0002	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 3 Result obtained by the five algorithms for 10 different runs on Wisconsin Breast Cancer dataset

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

Table 4 Values of parameters of each of five algorithms.

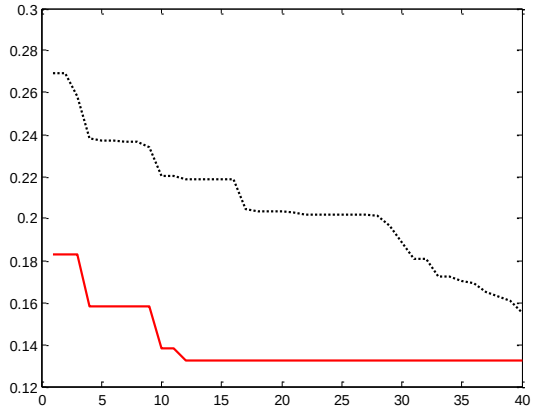

Fig 2.Min and Means Cost of ICA on Iris dataset

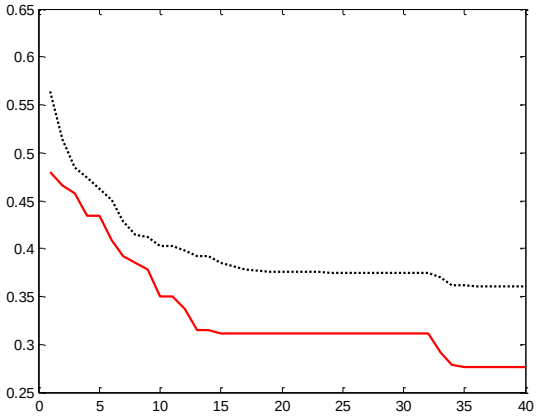

Fig 3. Min and Means Cost of ICA on Wine dataset

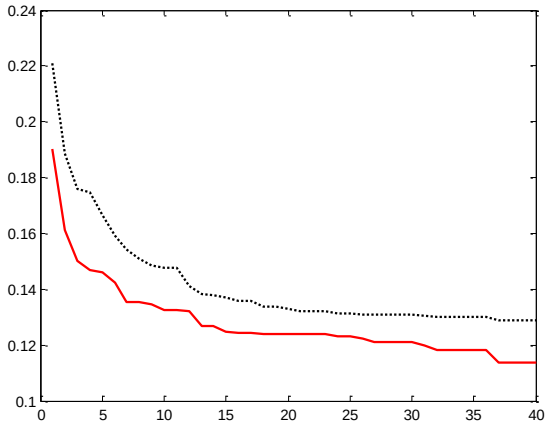

Fig 4. Min and means cost of ICA on Wisconsin Breast Cancer dataset

5. *Conclusion*

In summary, we developed a hybrid clustering algorithm by integrating dynamic validity index, K-means and imperialist competition algorithm to solve clustering problems.

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

References

Atashpaz Gargary, E., et al., 2008. Colonial competitive algorithm A novel approach for PID controller design in MIMO distillation column process, International Journal of Intelligent Computing and Cybernetics, 1, 337-355.

Atashpaz-Gargari, E., Lucas, Caro., 2007. Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic, IEEE Congress on Evolutionary Computation, 4661 – 4667.

Biabangard-Oskouyi, E., et al., 2009. Application of Imperialist Competitive Algorithm for Materials Property Characterization from Sharp Indentation Test, to be appeared in International Journal of Engineering Simulation.

Blake, C.L., Merz, C.J., UCI repository of machine learning databases. Available from: <http://www.ics.uci.edu/-mlearn/ MLRepository.html>.

Du, Z., et al., 2008. PK-means: A new algorithm for gene clustering, Computational Biology and Chemistry, 32, 243–247.

Hung, C.C., Wan, L., 2009. Hybridization of Particle Swarm Optimization with the K-Means Algorithm for Image Classification.

Kao, Y-T., et al., 2008. A hybridized approach to data clustering, Expert Systems with Applications, 34, 1754–1762.

Nazari-Shirkouhi, S., et al., 2010. Solving the integrated product mix-outsourcing problem using the Imperialist Competitive Algorithm, Expert Systems with Applications, 7615–7626.

Niknam, T., Amiri, B., 2010. An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis, Applied Soft Computing, 183–197.

Niknam, T., et al., 2009. An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering, Journal of Zhejiang University SCIENCE, 512-519.

Niknam, T., et al., 2010. An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering, Engineering Applications of Artificial Intelligence.

Nguyen, C.D., Cios, K.J., 2008. GAKREM: A novel hybrid clustering algorithm, Information Sciences, 178, 4205–4227.

Nopiah, Z.M., et al, 2009. A Weighted Genetic Algorithm Based Method for Clustering of Heteroscaled Datasets, International Conference on Signal Processing Systems.

Rajabioun, R., Atashpas-Gargari, E., Lucas, C., 2008, 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.

Tayefeh Mahmoudi, M., et al., 2009. Artificial Neural Network Weights Optimization based on Imperialist Competitive Algorithm.

Zhang, C., Ouyang, D., Ning, J., 2010, An artificial bee colony approach for clustering, Expert Systems with Applications, 37, 4761–4767