Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

The proposed Center Initialization Based on Imperialist Competitive Algorithm (CIB-ICA)

Sanaz Asfia[1,*], Arash Ghorbannia Delavar[2]

[1] *Sanaz Asfia , Dept of Computer, Payame Noor University, Tehran, Iran*

[2] *Arash Ghorbannia Delavar, Assistant Professor, Dept of Computer, Payame Noor University, Tehran, Iran*

[*] (Corresponding Author) ***sanazasfia@yahoo.com*,**

a_ghorbannia@pnu.ac.ir

Article history:
Received March 2014
Accepted April 2014
Available online May 2014

Abstract

In this paper we will introduce an initial cluster centers method for k-means algorithm, which can achieve a significant impact on the convergence and will not fall local optimal solution trap .The proposed Center Initialization Based on Imperialist Competitive Algorithm (CIB-ICA) uses imperialist competitive algorithm and minimum spanning tree(MST) features to reduce clustering error percentage of K-means algorithm. The proposed method has been evaluated on some famous datasets and experimental results show that it is an efficient cluster center initialization method.

Keywords: data mining, clustering, Imperialist Competitive Algorithm, K-means, center initialization.

1. Introduction

One of the most usable techniques of data mining is clustering [1]. Clustering is a process for partitioning datasets into several clusters consisting of similar objects .This technique 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 most similar between them and are least similar to the objects belonging to other clusters.

One of the most used classes of data clustering algorithms is the center based clustering algorithm.

k-Means algorithm (Mac Queen, 1967) is the most well known and the fast method in the center based clustering algorithms. Because of the simplicity of k-means algorithm, this algorithm is used in various fields. k-Means algorithm is a partitioning clustering method that separates data into k mutually excessive groups.

Through such the iterative partitioning, k-means algorithm minimizes the sum of distance from each data to its clusters. k-Means does not guarantee unique

clustering because we get different results with randomly chosen initial clusters.

The final cluster centroids may not be the optimal ones as the algorithm can converge into local optimal solutions. An empty cluster can be obtained if no points are allocated to the cluster during the assignment step. Therefore, it is quite important for k-means to have good initial cluster centers.

Several methods proposed to solve the cluster initialization for k-means algorithm. A recursive method for initializing the means by running k clustering problems is discussed by Duda and Har (1973) A variation of this method consists of taking the entire data into account and then randomly perturbing it k times. For the initial cluster center, Jain and Dubes (1988) applied the k-means with several times by randomly selected initial values and selected the average of these final cluster centers.

Bradley and Fayyad (1998) proposed the refinement algorithm that builds a set of small random sub-samples of the data, then clusters data in each sub-samples by k-means. All centroids of all sub-samples are then clustered together by k means using the kcentroids of each sub-sample as initial centers. The centers of the final clusters that give minimum clustering error are to be used as the initial centers for clustering the original set of data using k-means algorithm. Likas et al. (2003) proposed the global k-means algorithm which is an incremental approach to clustering which dynamically adds one cluster center at a time through a deterministic global search procedure consisting of N (with N being the size of the dataset(executions of the k-means algorithm from suitable initial positions.

Khan and Ahmad (2004) proposed Cluster Center Initialization Algorithm (CCIA) to solve cluster initialization problem. CCIA is based on two observations, which some patterns are very similar to each other. It initiates with calculating mean and standard deviation for data attributes, and then separates the data with normal curve into certain partition. CCIA uses k-means and densitybased multi scale data condensation to observe the similarity of data patterns before finding out the final initial clusters. The experiment results of the CCIA performed the effectiveness and robustness this method to solve the several clustering problems.

Deelers and Auwatanamongkol (2007) proposed an algorithm to compute initial cluster centers for k-means algorithm. They partitioned the data set in a cell using a cutting plane that divides cell in two smaller cells. The plane is perpendicular to the data axis with the highest variance and is designed to reduce the sum squared errors of the two cells as much as possible, while at the

same time keep the two cells far apart as possible. Also they partitioned the cells one at a time until the number of cells equals to the predefined number of clusters, k. In their method the centers of the k cells become the initial cluster centers for k-means algorithm.
M. Erisoglu et al (2011) proposed an algorithm to compute initial cluster centers for k-means algorithm. In this algorithm, two principal variables are selected according to maximum coefficient of the variation and minimum absolute value of the correlation. The reduced dataset is partitioned one at a time until the number of cluster equals to the predefined number of clusters. Then, cluster membership of each points are determined according to candidate initial cluster centers and selected two axis. For p variables, the initial cluster centers are created using the determined cluster memberships.

Imperialist Competition Algorithm

ICA is a novel global search heuristic that uses imperialism and imperialistic competition process.
This algorithm starts with some initial countries.

Imperialist competitive algorithm (ICA) is a new evolutionary algorithm that is inspired by the human's socio-political evolution. Each individual of the population is called country. The population is divided into two groups, colonies and imperialist state. The competition among imperialists to take possession of the colonies of each other forms the core of this algorithm. In this competition the weak empires collapse gradually and finally there is only one imperialist that all other countries are its colony.

3.1 Initialization of the empires

This algorithm like other evolutionary ones starts with an initial random population called countries. For creating imperials, some of the best countries in the population selected are considered imperialists and the rest form the colonies of these imperialists. In an M dimensional optimization problem, a country is a 1 × M array.
This array is defined as :

$$\mathrm{Country} = [p_1; p_2; \ldots; p_M] \tag{1}$$

Where $p_1, p_2, \ldots, p_M$ are variables of the problem. The cost of ith country (c_i) is found by evaluating the cost function f as :

$$c_i = f(\mathrm{Country}_i) = f(p_{i1}; p_{i2}; \ldots; p_{iM}) \tag{2}$$

where $p_{i1}, p_{i2}, \ldots, p_{iM}$ are the M variables of ith country and ci is its cost. To start the optimization process, N countries are generated as the initial population. The N_{imp} of them (countries with minimum cost) chosen as the imperialists. The N_{col} remaining countries are colonies. To form initial empires, the colonies are divided among imperialists based on their power. For distribution of the colonies among imperialists proportionally, the cost of each imperialist should be normalized as follows:

$$C_n = c_n - \max_i\{c_i\} \tag{3}$$

Here, c_n is the cost of nth imperialist and C_n is its normalized cost. Each imperialist that has more cost value will have less normalized cost value. Having the normalized cost, the power of each imperialist is calculated as:

$$p_n = |c_n / \sum_{i=1}^{Nimp} C_i| \tag{4}$$

Finally, the initial number of colonies of an empire is calculated as:

$$NC_n = \text{round}\{ p_n . (N_{col})\} \tag{5}$$

where NC_n is initial number of colonies of nth empire. For dividing the colonies among imperialist, NC_n of the colonies are randomly chosen and assigned to each imperialist. More powerful empires have a higher chance to obtain greater number of colonies

3.2 Assimilation (moving the colonies of an empire toward the Imperialist)

Imperialist countries try to improve their colonies. This fact is modeled by moving all colonies toward the imperialists. Each colony moves toward the imperialist by x units. The direction vector of the movement is the vector from colony to imperialist. Also, a random amount of deviation to the direction of movement to search different points around the imperialist is added. Then, the new position of each colony is determined by following formula in which h and x are two random variables with uniform distribution:

$$x \sim U(0, \beta \times d) \tag{6}$$
$$\theta \sim U(-\gamma, \gamma) \tag{7}$$

Here, x is vector size, h is deviation angle from direction of the movement from colony to imperialist, b is a number greater than one, d is the distance between colony and imperialist, and c is a

parameter that adjusts the deviation from original direction. These parameters are presented in Fig. 1.

3.3. Imperialist updating (exchanging positions of the imperialist and a colony)

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.

3.4. Realization of an imperialist competition

In this algorithm, the imperialistic competition has an important role. This competition gradually decreases the power of weaker empires and increases the power of more powerful ones. This event can be modeled by picking some of the weakest colonies of the weakest empires and making a competition among all empires to possess these colonies. In this part, it is necessary to determine which imperialist can possess this colony. This colony will not be possessed by the most powerful empires, but we want to give chance to all empires base on their total power. The total power of each empire is modeled by

$$\mathrm{TC}_n = \mathrm{Cost}(\mathrm{Imperialist}_n) + \varepsilon \times \mathrm{mean}\}\mathrm{Cost})\mathrm{colonies\ of\ empires}_n)\} \tag{8}$$

Where TC_n is the total cost of nth empire and n is a positive number which is considered to be less than 1. n can determine the ratio of imperialist and colonies in the total power of empire. Also, the amounts closer to zero can increase the role of imperialist to determine the total power of empires and vice versa for the amounts close to 1. After calculating the cost of all empires, they should be normalized as:

$$\mathrm{NTC}_n = \mathrm{TC}_n - \max\{\mathrm{TC}_i\} \tag{9}$$

where TC_n and NTC_n are the total cost and normalized total cost of nth empire, respectively. Having the normalized total cost, the possession probability of each empire is given by

$$\mathrm{Pp}_n = |\ \mathrm{NTC}_n \ /\ \textstyle\sum_{i=1}^{N_{imp}} \mathrm{NTC}_n\ | \tag{10}$$

Therefore, a vector with all P can be expressed as:

$$\mathrm{P} = \{\mathrm{P}_{p_1} ; \mathrm{P}_{p_2}; \ldots; \mathrm{P}_{p_{n_{imp}}}\} \tag{11}$$

It is necessary to create a vector with the same size as p such as R in which the elements are uniformly distributed random numbers.

Therefore, the vector R can be expressed as:

$$\mathrm{R} = \{r_1; r_2; \ldots; r_{N_{imp}}\}; \tag{12}$$

in which

$$r_1; r_2; \ldots, r_{N_{imp}} \sim U(1,0) \tag{13}$$

Subtracting R from P, the vector D is represented as:

$$\mathrm{D} = \mathrm{P} - \mathrm{R} = [D_1; D_2; \ldots; D_{N_{imp}}] \tag{14}$$

$$= [\mathrm{P}_{p_1} - r_1; \mathrm{P}_{p_1} - r_2; \ldots; \mathrm{P}_{p_{n_{imp}}} - r_{N_{imp}}] \tag{15}$$

Referring to vector D, the mentioned colonies are assigned to an empire that has the maximum relevant index in D.

3.5.Eliminating the powerless empires

In this competition, powerless empires will gradually lose their colonies. An empire finally collapses when it loses all of its colonies.

3.6.Convergence

After a while, all empires except the most powerful one will collapse and all the colonies will be under the control of this empire.

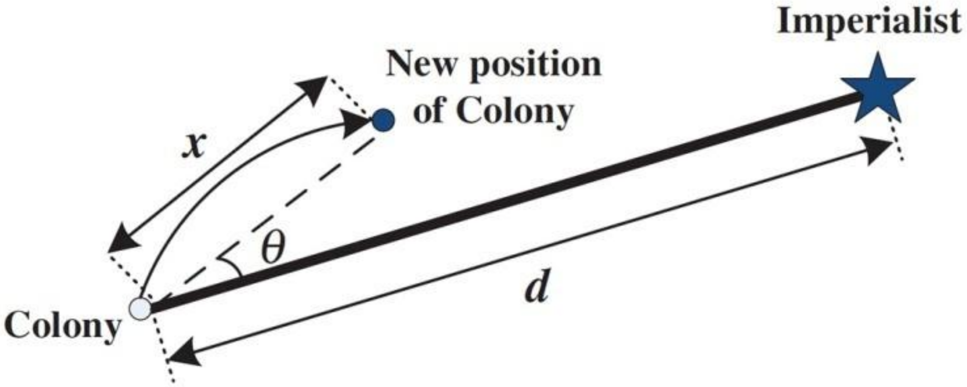

Figure 1 Movement of colonies toward their relevant imperialist.

Minimum spanning tree is made of weighted graphs .

Suppose the graph is a connected graph (ie, between two distinct vertices there is a path) to a spanning tree of the graph is a tree which contains all vertices in the graph, but only some of them include the edges. To the minimum spanning tree (a connected graph weighted) spanning tree among the trees of the graph, the total weight of its edges, the minimum amount possible.

2. Related work

To find optimized cluster centers for k-means algorithm different methods have been proposed. Celebi et al.[2] have compared computational efficiency of eight commonly used linear time complexity initialization methods. Their statistical analyses revealed that popular initialization methods such as **forgy**,**macqueen**, and **maximin** often perform poorly .

Liang Bai et al.[3] have applied local geometrical information to describe approximately the set of objects represented by a candidate cluster center then propose an acceleration mechanism for the production of new cluster centers for fast global k-means (FGKM) clustering algorithm.

Alaa H. Ahmed ,Wesam Ashour [4] have proposed an initialization method for k-means clustering based on reverse nearest neighbor (RNN) and Coupling Degree . For each instance RNN (x_i) is a set of instances whose nearest neighbor is x_i and the Coupling Degree measures the similarity between two instances. This algorithm first initialize a candidate set (CS) with all instances and a Representative set (RS) as an empty set then it computes the RNNs of each instance in CS, and ranks points according to the number of their RNNs in a descending order.

Then it selects the instance in the head of this ranked list as a center candidate, and deletes the selected point and its RNNs from the RNN list and (CS) and adds that instance to (RS). While list is not empty, the process of selection, deletion and recalculating of RNN of instances in (CS) will be continued. At the rest of the algorithm it computes number of neighbors in a specific radius and computes Coupling Degree of instances and adds mean of each neighborhood to centriod set.

Mohammad F. Eltibi, Wesam M. Ashour [5] proposed an algorithm which uses statistical information from dataset. It uses Central Limit Theory (CLT) which asserts that for large datasets, the distribution of dataset is approximately normal with mean μ and variance $\frac{1}{n}\sigma^2$ .This algorithm also uses Maximum Likelihood Estimator (MLE) and Gaussian distribution. The algorithm computes a radius and k-1 angular coordinates and chooses k instances with these coordinates and radius as initial cluster centers.

Coelho et al. [6] have introduced a modified ICA (MICA) approach based on concepts of attraction and repulsion between the colony and its imperialist is introduced during the search for better solutions. In their paper a brushless direct current wheel motor benchmark problem is used to investigate the performance of the classical ICA and the proposed MICA and results are shown to be competitive with those of other well-established optimization methods.

Ghanavati et al. [7] have developed imperialist competition algorithm to solve clustering problem.Their algorithm doesn't need to determine the number of clusters and their cost function is based on minimum distance of each instance to points of other clusters and average distance of each instance and cluster center

Niknam et al. [8] have presented a hybrid evolutionary optimization algorithm based on combining Imperialist Competitive Algorithm and K-means (K) for optimum clustering N objects into K clusters. The cost function of their algorithm is just based on distance to cluster center.

Fazel Zarandi et al. [9] have used Imperialist Competitive Algorithm in their new hybrid clustering algorithm for model structure identification. In their paper a new fuzzy functions model is presented and its main parameters are optimized with simulated annealing approach. For this purpose, a new hybrid clustering algorithm for model structure identification is proposed. This model is based on hybridization of extended version of possibilistic c-mean (PCM) clustering with mahalonobise distance measure and a noise rejection method.

Erisoglu et al. [10] have proposed a method which is based on the choosing the two of the p variables that best describes the change in the dataset according to two axes. After determining two axes, the mean of data points is calculated as the center of the dataset according to selected axis. Then the data point with the highest distance with other candidate cluster centers will be selected as initial cluster center.

R. Maghsoudi et al. [11] have proposed a model that named it GA-Clustering for improving K-Means method has been introduced. K-Means algorithm has been surveyed and tried to apply one of the optimization method named genetic algorithm improve in unsupervised classification procedure.

Kh. Jalali et al. [12] have explains the task of segmenting image by improved distance measure in K means algorithms and SOM. This method given that did not use the concept of grade of membership .Using the fuzzy degree of membership has come down to pixel noise in a classroom and thereby Their intervention comes down in classification. in the classification of pixels, they showed poor performance on noisy images.

3. Proposed method

In this section in order to solve an optimization problem, we will introduce proposed cluster center initialization method which is based on Imperialist Competitive Algorithm (ICA).So the whole dataset will be like **universe** and each instance with *m* attributes will be a **country** ($C_i = [P_{i1}, P_{i2}, \dots, P_{im}]$) of universe.

Each cluster in data set is an **empire** and if an instance is cluster center it is an **imperialist** and otherwise it is a **colony**. So in a dataset with k clusters and N instances we have K empires, K imperialists and N-K colonies.

At the beginning of the method we should compute the distance of any country with all other countries.

$$\boldsymbol{D}(\mathrm{C}_{\boldsymbol{x}}, \mathrm{C}_{\boldsymbol{y}}) = \sqrt[2]{\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} (\mathrm{C}_{\boldsymbol{xi}} - \mathrm{C}_{\boldsymbol{yi}})^{\boldsymbol{2}}} \tag{16}$$

After we computed Euclidean Distance between all countries, we will have a complete connected, undirected graph *G*= *(V, E)*, where *V* is the set of nodes and in proposed method is equal to countries in universe (|*V*|= N), *E* is the set of edges between pairs of nodes (countries), and a weight *w (u, v)* specifying weight of the edge *(u, v)* for each edge *(u, v)* ∈ *E*.

As you know, the Minimum Spanning Tree MST=*(V`,E`)* of a weighted graph is a sub graph of the main graph MST ⊆G which forms a tree and has equal set of nodes with main graph *V*=*V`* and a set of edges which is sub set of edges of the main graph E`⊆ E .Number of edges in a minimum spanning tree is equal to the number of nodes of the main graph |*E`*|=|*V`*| and sum of weights of it is less than or equal to any other sub graph which forms a tree containing all nodes of the main graph.

There are some graphs that don’t have any spanning tree but any complete graph, like the graph that we form with computing distances between all countries, at least has one minimum spanning tree. Although minimum spanning tree is not unique, it doesn’t have any effect on our algorithm.

We will make adjacency matrix of universe and find minimum spanning tree of the universe. This minimum spanning tree is a **road map** which guides us in finding shortest path between each two countries and also we will have just one road between two nearby empires. Figure .2 shows a dataset which has three clusters .It also shows the minimum spanning tree of that dataset and three empires which have been shown with e_i and the imperialists with I_i.

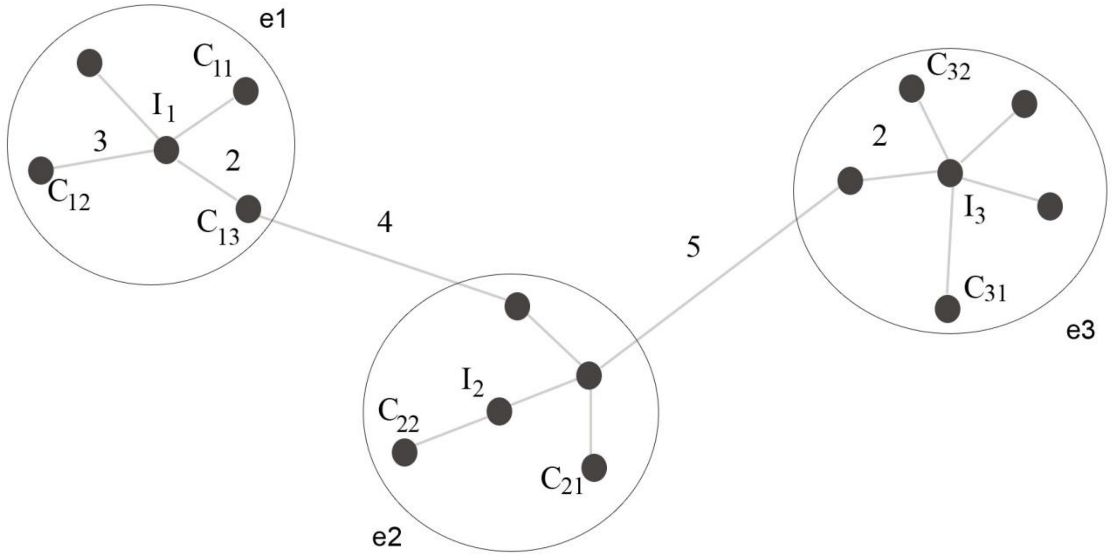

Figure 2 Road map of a data set with 3 clusters

In our minimum spanning tree if we remove K-1 edges we will have K connected components and each connected component is one of empires .To find initial empires of method we will randomly remove K-1 edges or roads between countries .The roads which ends to countries that doesn't have any other road to else ware will not be removed from the road map. For instance in Figure .2 if we remover the edge between I_1 and C_{12} we will have an empire with just one country, so proposed method will not remove this age.

After finding initial empires we need to introduce a country as imperialist of it. Initial imperialists also will be chosen randomly among countries that have more than one road. Now we have divided universe to K empires and have chosen K imperialist and proposed method should be repeated for predefined decades (D) and introduce final imperialists as initial cluster centers for K-means algorithm.

For any country (C_x) we should find the distance to the nearest neighbor $(NN(C_x))$ country in the empire (e_i) and compute a neighborhood radius $(Nr(e_i))$ for each empire as median of distances to the nearest neighbor.

$$NN(C_x) = \mathbf{min}_{C_x,C_y\in e_i}\left\{\sqrt[2]{\sum_{i=1}^{m}(\mathrm{C}_{xi}-\mathrm{C}_{yi})^{2}}\right\} \tag{17}$$

$$Nr(e_i) = median\left(\left(NN(C_x)\right)\right) \tag{18}$$

Now that we have computed a neighborhood radius ($Nr(e_i)$) we should find number of neighbors of each country in its empire in neighborhood radius we define $N_{Nr(e_i)}(C_x)$ as

$$N_{Nr(e_i)}(C_x) = \left|\left\{C_y \middle| \boldsymbol{D}(\mathrm{C}_{\boldsymbol{x}}, \mathrm{C}_{\boldsymbol{y}}) \le Nr(e_i)\right\}\right| \tag{19}$$

We will also compute the average distance from each country to other countries in empire ($AvgDis(C_x)$).

$$AvgDis(C_x) = Avg\left(\sqrt[2]{\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} (\mathrm{C}_{\boldsymbol{xi}} - \mathrm{C}_{\boldsymbol{yi}})^{\boldsymbol{2}}}\right), C_x, C_y \in e_i \tag{20}$$

For instance in Figure 2 $Avg(C_{13})_{e_i} = 2$ .The roads which connect two empires together will not considered in computing $Avg(C_x)_{e_i}$.

We will also compute the number of edges leading up to any given country on imperialist MST as bellow

$$NL(C_x)_{e_i} = \left|\left\{C_y \middle| \text{edge } (C_x, C_y) \in MST, C_x, C_y \in e_i\right\}\right| \tag{21}$$

$NN(C_x)$, $Nr(e_i)$, $N_{Nr(e_i)}(C_x)$,$AvgDis(C_x)$ and $NL(C_x)_{e_i}$ will be computed again for recreated empires and their countries.

Now we have collected enough information about countries and we will compute cost of each country. The **Cost Function** of proposed method is defined as bellow

$$\mathrm{C}(\mathrm{C_x}) = \left(\frac{\left(\frac{AvgDis\ (C_x)}{\max\left(AvgDis\ (C_x)\right)}\right)}{\frac{\max\ (N_{Nr\,(e_i)}(C_x))}{N_{Nr\,(e_i)}(C_x)} * \frac{\max\ (NL\,(C_x)_{e_i})}{NL\,(C_x)_{e_i}}}\right) \tag{22}$$

Power of the imperialists will be computed as below

$$P(I_i) = \mathrm{C}(I_i) + \varepsilon \times median\{C(C_x)\} \tag{23}$$

In each empire we will choose a colony with the lowest cost as a candidate for imperialists. And we will use new candidate as initial cluster centers of k-means algorithm and cluster the whole dataset and introduce new empires and their colonies with candidate imperialists. Then we will compute the power of candidate imperialists and for each empire we will select the new imperialist between previous imperialist and candidate according to their power.

The above steps will be repeated for a predefined number of decades. Then we will introduce final imperialists as initial cluster centers of K-means algorithm.

Table 1.Pseudo code of proposed algorithm

```
Input: dataset, number of clusters (k), number of decades (D), ε
Output: initial cluster centers for k-means algorithm
• For each colony set $\boldsymbol{D}(\mathrm{C}_{\boldsymbol{x}}, \mathrm{C}_{\boldsymbol{y}}) = \sqrt[2]{\sum_{i=1}^{m} (\mathrm{C}_{xi} - \mathrm{C}_{yi})^{2}}$
• Create adjacency matrix of universe.
• Create MST of a universe.
• Remove k-1 edges connected to non-leaf nodes in MST.
• Compute connected components.
• Introduce connected components as initial empire.
• Select initial imperialist randomly from each empire of non-leaf nodes and form empires.
• For  decade = 1 to D  do
    • Remove k-1 edges based on size of edges and the more road cost (Re-create imperialists)
    • Compute connected components.
    • Introduce connected components as initial imperialist.
    • For each empire compute
        ○ $Nr(e_i) = median\left(\left(NN(C_x)\right)\right)$.
    • For each colony compute
        ○ $NN(C_x) = \mathbf{min}_{C_x, C_y \in e_i} \left\{ \sqrt[2]{\sum_{i=1}^{m} (\mathrm{C}_{xi} - \mathrm{C}_{yi})^{2}} \right\}$.
        ○ $N_{Nr(e_i)}(C_x) = \left| \left\{ C_y \middle| \boldsymbol{D}(\mathrm{C}_{\boldsymbol{x}}, \mathrm{C}_{\boldsymbol{y}}) \leq Nr(e_i) \right\} \right|$.
        ○ $AvgDis(C_x) = Avg\left( \sqrt[2]{\sum_{i=1}^{m} (\mathrm{C}_{xi} - \mathrm{C}_{yi})^{2}} \right), C_x, C_y \in e_i$.
        ○ $NL(C_x)_{e_i} = \left| \left\{ C_y \middle| \text{edge } (C_x, C_y) \in MST, C_x, C_y \in e_i \right\} \right|$.
        ○ $C(\mathrm{C_x}) = \left( \frac{\left( \frac{AvgDis\ (C_x)}{\max\ (AvgDis\ (C_x))} \right)}{\frac{\max\ \left(N_{Nr\,(e_i)}(C_x)\right)}{N_{Nr\,(e_i)}(C_x)} * \frac{\max\ \left(NL\,(C_x)_{e_i}\right)}{NL\,(C_x)_{e_i}}} \right)$.
    • For each empire compute
        ○ $P(I_x) = P(I_i) = \mathrm{C}(I_i) + \varepsilon \times\ median\{C(C_x)\}$
        ○ Introduce a colony of the lowest cost as a candidate to become empire.
    • Use k-means clustering based on the data of the candidates for the empire.
    • Compute Power of candidates.
    • Select the new empire of the previous imperialist and candidate imperialists by more
      powerful.
• Introduce last empires as optimal cluster centers for k-means algorithm.
• Finish.
```

4. Experimental result

We have implemented proposed algorithms with java language in Netbeens IDE 7.1 environment and used weka.jar library of weka 3.6.6 software which contains necessary classes for implementing a new clusterer.

After we implemented and tested the algorithm, we used code of simple k-means of weka 3.6.6 and changed it to use initial cluster centers of our algorithm instead of random initialization of cluster centers for it. So except choosing initial cluster centers all other steps will be done according to standard k-means algorithm. Table 2 shows execution parameters.

Table 2 Execution parameters

Number Of Decades	10
Executions	100
ε	0.7

We have tested and compared standard k-means clustering error percentage on some famous data sets [13] with initial cluster centers of proposed algorithm and random initialization. Table 3 and Figure 3 show average error percentage of 100 times execution result.

Table 3 Clustering error percentage of k-means algorithm with different cluster center initializations

Data set	Instances	clusters	attributes	Random initialization		Proposed method	
				Error	Iterations	Error	Iterations
Iris	150	3	4	16.67%	6	14.00%	5
liver-disorders	345	2	6	51.30%	7	46.96%	5
spectf_train	80	2	43	42.50%	3	35.00%	3
Glass	214	6	9	58.41%	10	51.40%	8
Hepatitis	155	2	19	43.22%	5	38.71%	4
Vehicle	846	4	18	63.36%	8	59.10%	6

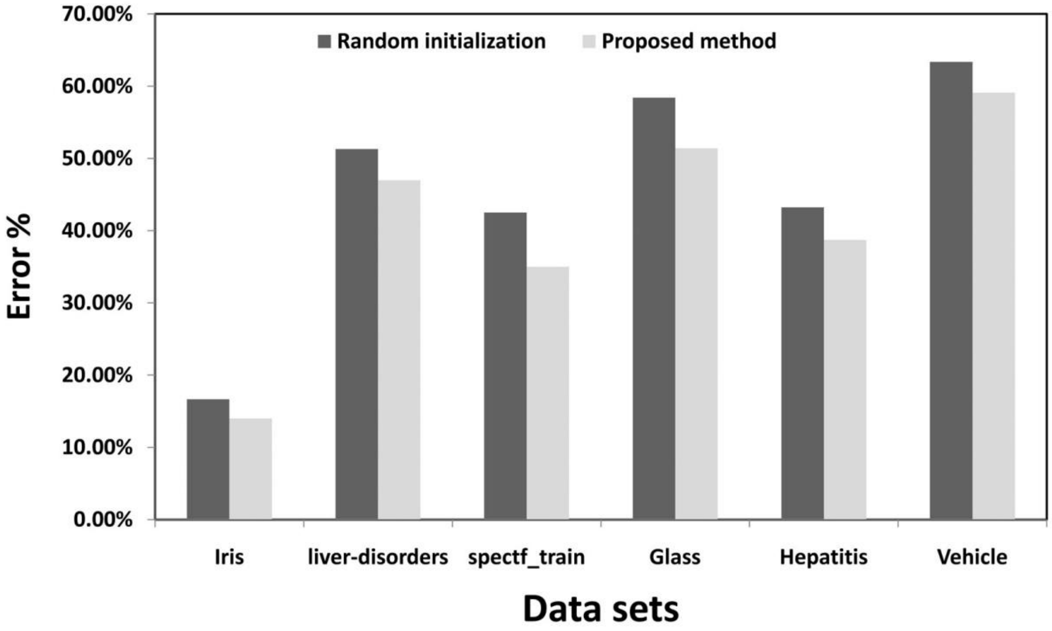

Figure 3 Clustering error percentage of k-means algorithm with different cluster center initializations

5. Conclusion

The imperialist competitive algorithm is a new method, which has great abilities to cope with different types of optimization problems. In this paper, has been proposed a new cluster center initialization method for k-means algorithm called Center Initialization Based on Imperialist Competitive Algorithm (CIB-ICA) which is based on Imperialist Competitive Algorithm (ICA) and uses advantages of minimum spanning tree to find the best cluster centers. Experimental results show that proposed method can improve efficiency of k-means algorithm and decreases clustering error percentage. Proposed method is able to be used as a clustering algorithm as future work and will be able to detect arbitrary shape clusters.

Refrences

[1] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: CEC IEEE Congress on Evolutionary Computation, 2007.

[2] Celebi, M. E., Kingravi, H. A., and Vela, P. A. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications 40, 1 (2013), 200 – 210.

[3] Liang Bai , Jiye Liang , Chao Sui , Chuangyin Dang ,Fast global k-means clustering based on local geometrical information . Information Sciences 245,(2013), 168 – 180.

[4] Ahmed, A. H., and Ashour, W. An initialization method for the k-means algorithm using rnn and coupling degree. International Journal of Computer Applications 25, 1 (July 2011), 1–6. Published by Foundation of Computer Science, New York, USA.

[5] Eltibi, M. F., and Ashour, W. M. Initializing k-means clustering algorithm using statistical information. International Journal of Computer Applications 29, 7 (September 2011), 51–55. Published by Foundation of Computer Science,New York, USA.

[6] Leandro Dos S. Coelho, Leonardo D. Afonso, Piergiorgio Alotto, A Modified Imperialist Competitive Algorithm for Optimization in Electromagnetics, IEEE TRANSACTIONS ON MAGNETICS, VOL. 48, NO. 2, (FEBRUARY 2012) 579-582.

[7] Mojgan Ghanavati, Mohamad reza Gholamian, Behrouz Minaei, Mehran Davoudi, An efficient cost function for imperialist competitive algorithm to find best clusters, Journal of Theoretical and Applied Information Technology, vol .29 No.1 (July- 2011) 22-31.

[8] Taher Niknam, Elahe Taherian Fard, Narges Pourjafarian, Alireza Rousta, An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering, Engineering Applications of Artificial Intelligence 24 (2011) 306–317.

[9] M.H. Fazel Zarandi , M. Zarinbal , N. Ghanbari , I.B. Turksen, A new fuzzy functions model tuned by hybridizing imperialist competitive algorithm and simulated annealing Application: Stock price prediction, Information Sciences 222 (2013) 213–228.

[10] Erisoglu, M., Calis, N., and Sakallioglu, S. A new algorithm for initial cluster centers in k-means algorithm. Pattern Recogn. Lett. 32, 14 (oct, 2011), 1701–1705.

[11] Rouhollah Maghsoudi, Arash Ghorbannia Delavar, Somayye Hoseyny, Rahmatollah Asgari, Yaghub Heidari, Representing the New Model for Improving K-Means Clustering Algorithm based on Genetic Algorithm, The Journal of Mathematics and Computer Science, vol .2 No.2 (2011) 329-336.

[12] Khosro Jalali, Mostafa Heydari, Asma Tanavar, Image Segmentation with Improved Distance Measure in SOM and K Means Algorithms, The Journal of Mathematics and Computer Science , vol.8 (2014) 367-376

[13] http://repository.seasr.org/Datasets/UCI/arff/