Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

A Novel Document Clustering Algorithm Based on Ant Colony Optimization Algorithm

Kayvan Azaryuon[1] , Babak Fakhar[2]

[1]Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
k.azaryoun@yahoo.com

[2]Department of Computer Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
fakharbabak@yahoo.com

Article history:
Received March 2013
Accepted May 2013
Available online May 2013

Abstract

Document clustering based on ant colony optimization algorithm has lately attracted the attention of many scholars throughout the globe. The aim of document clustering is to place similar content in one group, and non-similar contents in separate groups. In this article, by changing the behavior model of ant movement, we attempt to upgrade the standard ant's clustering algorithm. Ants' movement is completely random in the standard clustering algorithm. On the one hand, we improve the algorithm's efficiency by making ant movements purposeful, and on the other hand, by changing the rules of ant movement, we provide conditions so that the carrier ant moves to a location with intensive similarity with the carried component, and the non-carrier ant moves to a location where a component is surrounded by dissimilar components. We tested our proposed algorithm on a set of documents extracted from the 21578 Reuters Information Bank. Results show that the proposed algorithm on presents a better average performance compared to the standard ants clustering algorithm, and the K-means algorithm.

Keywords: Ant Colony Optimization Algorithm, Ant Clustering, Document Clustering, Ant Movement Rules

1. Introduction

The Communication Era with it greatly influenced one the most important challenges of humanity that is Electronic Learning (eLearning). In the infinite internet world of information, rapid, clear, and comprehensive access to information is an important aspect in the process of teaching and learning.

Therefore, with the ever-growing expansion of eLearning, the need for tools capable of automatically creating training documents in certain fields is essential. Therefore, designing efficient methods for clustering documents is an important step in direction of designing and implementing such tools. Document clustering is an unsupervised process for classification of text data. The goal of document clustering is to place different text in different groups. This process is carried out in a way that documents in one group are always similar to each other compared to documents in other groups. Each group is called a cluster. We should note that document clustering in contrary to tent categorization that is a supervised process, does not include any presumptions regarding document grouping [13].

Many studies have been carried out around text mining and document clustering in the past decade. Some of these studies include:

- Hierarchical clustering (Watigan, 1963)
- K-means (Hartigan, Wong 1979)
- Ankersts Beruning Optics (1999)
- Clustering based on the Genetic Algorithm (GA) (Chiou and Lan, 2000)

Notwithstanding numerous research carried out in this arena, yet efficiency of present algorithms is not reasonable enough to be deployed in real world applications. Therefore, researches on creating optimized methods in this field are still in progress. Fundamentally, a text clustering method can be produced from a combination of a data clustering method with a document classification method based on similarity. Thus, we can conclude that research on document clustering can be divided into the two branches of data clustering and document classification.

Extensive research has been carried out in [10] regarding ant clustering by Beni and Wang (1989), Bonabeau, and Dorigo (1999). Ant Clustering is designed based on the idea that ants always carry their corpses and newborns, which are placed alone in a location, to places where there are a collection of corpses and newborns (corpses and newborns in separate groups). At first, they attempt to place their corpses and newborns in more than two locations. However, based on an evolutionary and completely random process, eventually they attempt to combine smaller clusters to create larger clusters. Increase in size of a cluster, results in increase of its absorption rate. Finally, corpses and newborns are clustered in two separate clusters. In 1994, Lumer upgraded Deneubourg’s 1991 algorithm and used it to cluster numerical data. Based on these studies, later, Handel presented an algorithm for document clustering based on ant colony.

In this article, by modifying the ant movement behavior model, we attempt to increase the efficiency of the standard clustering algorithm. Our research around current methods shows that ant movement is blindly in all these methods. This ant behavior results in waste of a great volume of the algorithm computing capacity, and extends algorithm execution time. In order to resolve these limitations, a mechanism for determining ant movement is presented in the proposed algorithm. Based on this mechanism, carrier ants move toward the location with the most similarity of elements to the carried element, and non-carrier ants move toward a location where a non-similar element is surrounded by similar elements. By carrying out these modifications, significant increase will be gained both in speed of the algorithm and the quality of produced clusters.

We tested our proposed algorithm on a collection of documents extracted from the 21578 Reuters database. Test results indicated that the proposed algorithm exhibited relatively better performance compared to the standard ant-clustering algorithm and the K-means algorithm.

2. Ant Colony Algorithm (ACS)

Models based on natural systems are successful samples of solving combined optimization problems (e.g. NP-Hard). The ant colony algorithm is one of these problems, which by presenting a model-oriented search method, has introduced new context in problem solving methods [2].

This model was first introduced by Dorigo, and was used to solve the travelling salesman problem[14]. The ACS algorithm showed desirable efficiency in solving the travelling salesman problem. The structure of the travelling salesman problem assumes a salesman that needs to travel to multiple cities in order to sell his products. Each city is connected to other cities via a road. In order to minimize travel time, the shortest route must be selected in a way that starts from the salesman's city, and passes each city only once, and in the end, returns to the salesman's city. Determining the shortest route for this problem is called the travelling salesman. If the problem space is embodied as a graph, we can say that the travelling salesman problem will result in a minimum Hamiltonian circuit. The ACS algorithm when solving this problem is as follows: assume n cities exist in the problem, which for each two cities of *i* and *j*, *d(i,j)* represents the distance between the two cities. At the beginning of the cycle, the ants are randomly distributed in the cities in *(m<=n)m* numbers. Each ant, moves based on equation (1) after selecting the next city. [7]

In nature, ants find the shortest distance from their hole to their food. Ants use a substance called Pheromone in order to represent their paths. The ant colony optimization algorithm presents a random search method. This method, with the aid of pheromone representation provided positive feedback, which by repetition of the algorithm execution, converges the ants in optimized paths, and finally, results in a response. The work process is in a way that each ant marks the distance between two nodes (cities) using pheromone, and in this way, increases desirability of a edge for the next selection. The amount of pheromone for this edge is displayed with the $\tau(i,j)$ parameter. In the beginning of each movement, equations (1) and (2) are utilized in order to determine the next step for *s*, while *k* shows the index for the ant residing in city *r*. Equation (1) is a greedy choice method based on the best possible combination from the distance and level of pheromone, and equation (2) balances this procedure by implementing a probabilistic selector.

$$S = \begin{cases} \arg\max_{u\in} J_k(r) \{ \tau(r,u) \lfloor d(r,u) \rfloor^{\beta} \; if q <= q0 \quad (Exploita \\ \\ Equation 2 \qquad otherwise \quad (Exploration) \end{cases} \tag{1}$$

$$P_k(r,s) = \begin{cases} \dfrac{\tau(r,s) \lfloor d(r,s) \rfloor^{\beta}}{\sum_{u\in J_k(r)} \tau(r,s) \lfloor d(r,s) \rfloor^{\beta}} & if \; s \in J_k(r) \\ \\ 0 & otherwise \end{cases} \tag{2}$$

Parameter q is a random number in the range of [0, 1], which is determined based on a uniform distribution of possibility. Parameter q_0 is a constant value, which is used to adjust queries. A list of visited nodes is kept in the back of each ant, therefore cities that are not visited by ant *k* with the current location of *r,* are kept in list $J_k(r)$. In addition, *β* is a negative parameter that represents effect of distance. The higher value of *T(r,s)* meaning pheromone of a edge, increase the absorption rate of other ants in travelling that edge. After an ant passes a edge, the pheromone of that edge increases based on equation (3) [9].

$$\tau(r,s) = (1-\rho).\tau(r,s) + \rho.\tau 0 \tag{3}$$

In order to prevent rapid convergence of ants in one course, and falling in the trap of local optimal response, the term pheromone evaporation is used. Parameter ρ is used for this job. After completion of each repetition cycle of the algorithm, all ants have produced a complete tour, and it is time for a universal update. The universal update must be conducted in a way that the shortest distance be encouraged based on equation (4) [8].

$$\tau(r,s)=(1-\gamma).\tau(r,s)+\gamma.\Delta\tau(r,s) \qquad (4)$$

$$\Delta\tau(r,s)=\begin{cases}\frac{Q}{L} & if\ (r,s)\in globally\ best\ t\\ 0 & otherwis\end{cases} \qquad (5)$$

$\Delta T(r,s)$ indicates the amount of pheromone rise in the best tour, which is obtained from equation (5). L is the length of the shortest tour until the end of the current repetition cycle, and Q is an evaporation constant parameter for the whole problem, and represents the quality of response. Y is the universal evaporation parameter of pheromone (0<Y<1).

3. Document Display

Prior to commencement of the clustering operation, existing data in the documents must be suitably displayed. The operation of displaying a document is in a way that firstly, textual data of the documents are mapped on the feature numerical vector. The Vector Space Model (SVM) is one of the conventional methods for this mapping. Two main stages I this method are extracting features and representing each document based on extracted features [1]. In the feature extraction stage, differentiating keywords of the document collection are obtained after extraction of the utilized keywords, and eliminating generic words (e.g. since, an, the), and finding linguistic roots of words (e.g. merging words like engineering, engineer, and engineers). Then, each document is represented in form an *m*-numbered vector in an *m* dimensional space, which *m* is the number of extracted features. The value of each element in the vector representing each document, introduces the importance of this feature (word) in that document, and is usually calculated based on one of the binary representation methods or word frequency display in the Term Frequency/Inverse Document Frequency (TF/ID). The value of each element in the binary representation of each document is one, provided the corresponding word with that element has appeared at least once. Otherwise, the value is zero. In the TF-IDF representation, each element is calculated based on equation (6). With this method, a larger weight is given to words with a larger frequency in related documents, but have seldom appeared in the document collection, and therefore, have a higher information value.

$$w_{ij}=tf_{ij}.\log\frac{N}{df_i} \qquad (6)$$

where:

wij is the weight of the j^{th} word in the i^{th} document, tfij is the j^{th} word frequency in the i^{th} document, N is the total number of documents in the documents collection, df_i is the number of documents in which the j^{th} word has appeared at least once.

When the document representing vectors are created using one of the mentioned methods, we can define the similarity criteria between documents based on Euclidean space, and or internal product of vectors in vector space.

4. Standard Ant Clustering Algorithm

Deneubourg initially presented the Ant Clustering Algorithm in 1991, and after that, Lumer upgraded the algorithm in 1994. Other existing algorithms in the field of ant clustering are based on the algorithms provided by Deneubourg and Lumer. In this article, the standard ant-clustering algorithm we discuss is the algorithm presented by Lumer[15].

The standard ant-clustering algorithm is in a way that firstly, a two-dimensional $m \times m$ space is considered. The size of this space must be chosen larger than the clustered elements. Therefore, the algorithm will have sufficient time to converge to the final clusters. The number of ants and the value m is determined according to the number of clustering elements. Supposing n is the number of clustered elements, then the number of ants will be $m=4n$, $n/3$. The ants within a repetitive process surround a stack of two or more elements. Then, by combining smaller stacks, attempt to create larger stacks [4, 5].

In the end, the obtained stacks are considered as clusters. The stacks are created based on the fact that ants collect similar elements in adjacent locations. If stack H includes N_H number of elements, for each stack containing element O_i, three parameters can be defined.

1. Center of the element present in each stack:

$$O_{center}(H) = \frac{1}{n_H} \sum_{Oi \in H} Oi$$

2. The longest distance in terms of similarity between two specimens in one stack can be defined as follows:

$$D_{man}(H) - man_{oi,oj \in H} D(O_i O_j)$$

In this relation, D can be a statistical distance (e.g. a Euclidean distance). Hence, the most different element in a stack can be the element with the most distance from the center of the stack. This element is shown as H.

The mean distance between of elements present in a stack with the center of the stack is shown as follows:

$$O_{center}(H) = \frac{1}{n_H} \sum_{Oi \in H} D(O_i, O_{center}(H))$$

After completing the initial value-setting phase, each ant completes the following stages in terms of a repetitive process.

IV.1 Picking Up

If an ant is not carrying any element, it will attempt to pick up the O_i element with a probability of P_p in its surrounding eight squares. The lesser the similarity of O_i with its neighbors, the higher P_p will be.

IV.2. Moving

When an ant completes it decision-making process about picking up an element, then, it changes status to one of the adjacent cells. In the ant-clustering algorithm presented by Deneubourg, ants are unable to move to cells previously occupied by other ants. However, this limitation has been eliminated in other versions of the algorithm. Each ant faces one of the following three situations within its surrounding cells [3]:

1. Only one element exists in each cell. In this case, the ant picks up the element with the probability of P_p.
2. Two elements exist inside a cell. In this case, with the probability of $P_{destroy}$, it breaks the two-element cluster into two separate elements, and randomly picks up one of them.
3. In each cell exists on stack with more than two elements. In this case, ants pick up the most different element in the stack based on the similarity criterion. The distance from the selected element's center must be more than T_{remove}.

$$\frac{D(o_{dissim}(H), o_{center}(H))}{D_{mean}(H)} > T_{remove}$$

IV.3. Dropping

Based on the probability P_d, when the carrier ant reaches a new cell, then it decides about dropping the carried element. In this case, three situations may occur:

1. The new cell is empty. In this case, with a fixed probability, the ant attempts to drop the carried element [3, 4].
2. The new cell contains two elements. In this state, the amount of similarity between the carried element sand the existing elements in the cell are measured. The more the similarity, the more will be added to the probability P_d.
3. The new cell contains a stack of elements. In this case, the amount of similarity of the carried element with the stack center element is measured, where more similarity between these elements will increase the probability P_d [7].

Repetition of the above stages results in the classification of elements in different groups. The algorithm either continues for a specified number of repetitions, and or persists until reaching stability. Two important factors in the ant-clustering algorithm are P_p and P_d probability values. Deneubourg in his article calculated the values of these two parameters based on the following functions:

$$P_p = (\frac{k1}{k_1 + f})^2 \text{ Probability of picking up}$$

$$P_d = (\frac{f}{k_2 + f})^2 \text{ Probability of dropping}$$

In these functions, f represents the similarity of element O_i with its neighbors. K_1 and K_2 are two threshold values. K_1 is the threshold of picking up, and K_2 is the threshold of dropping. By running the standard clustering algorithm on different data, we found the following weaknesses [8]:

- Ant movement in this algorithm is completely random, and ants blindly attempt to move to a new location in any stage. On this basis, the majority of ants will probably not do any useful work in different stages of the algorithm. These conditions will result in efficiency reduction of the algorithm.
- Ant activity in terms of picking up and dropping elements is only based on the K_1 and K_2 threshold values. Therefore, the correct configuration of these values is extremely effective in generating desirable results. In the standard ant-clustering algorithm, the value of these two parameters is considered constant throughout all stages of the algorithm.
- The range, in which similarity of documents is calculated compared to each other, is considered the view range of ants from other influencing parameters in efficiency of the ant-clustering algorithm. This is a constant value in the standard clustering algorithm, which is determined at the beginning of the algorithm. This constant value in some cases causes the algorithm to act abnormally while carrying out the actual clustering process [9, 10, and 11].

For example, if the value of this parameter is low, then in most cases the algorithm will be incapable of generating large clusters. On the other hand, if a large parameter value is selected, then the algorithm will not be able to extract larger clusters into smaller clusters, and this causes some clusters to have non-similar elements.

5. Change in Ant Movement Rules (The Proposed Algorithm)

As explained in the previous section, random movement of ants results in algorithm inefficiency. In order to create a purposeful ant movement regime, we can devise this heuristic that an ant always moves to a region where the carried element is most similar to elements of that region, and the non-carrier ant always moves to a region where an element is surrounded by non-similar elements. In order to execute this process, we assume that each ant has access to a universal map. In this map, the location of each document is specified in the two-dimensional environment. This map acts as a universal cache memory for determining document locations. Using this universal cache memory, a carrier ant is able to find a location with most similarity with its carried element, and move towards that location. Ants in order to select a correct path of movement, must calculate the amount of similarity between their carried element, and existing elements in different locations. Executing this process in conditions where the two-dimensional space (*M*) is large requires a great deal of computation. Therefore, having a universal cache memory in this situation does not seem useful. Thus, in this section, we need compromise between precision in determining locations with maximum similarity and computation volume. Our solution in this section is for the surface space (m*m) to be broken into smaller spaces, and then one of the subspaces be selected randomly, followed by calculation of the amount of similarity. The number of subspaces is dependent on the number of clusters and number of clustered elements. In addition, by calculating the amount of similarity of the ant's original location, we provide a condition in which the ant does not move to a location where its similarity to the carried element is less than the original location. In such conditions, the ants attempt to move after calculating the amount of similarity between the current location and neighbors. In algorithmic sense, movement of the carrier ant can be explained as follows:

1. The amount of similarity of the selected element for carrying to the external cell is calculated.
2. The $n*n$ gird is broken into $l*l$ subspaces and an empty cell is selected randomly in each of the resulting subspaces.
3. The amount of similarity for element *O* is calculated is calculated with each of the selected cells.
4. Among the randomly selected cells, the cell with the most similarity with element *O* is selected.
5. The carrier ant moves towards the cell.

In this solution, we have used the following relation in order the calculate the similarity of element O_i with its neighbors. This relation is known as Moore Neighborhood.

$$f(o) = j^{\sum sim(o,o_j)/N(o)}$$

In this relation, the concept of *f(o)* explains the amount of similarity of element *O* with existing elements in the Moore Neighborhood of element *O*. Element O_j is the neighbor. $Sim(O,O_j)$ explains the amount of similarity between elements *O* and O_j, and finally, *N(o)* specifies the number of cells in the Moore Neighborhood. This value is determined based on the scope determined as neighborhood. In contrast to carrier ants, which are continuously moving towards a region most similar to their carried element, non-carrier ants move towards a region where an element is surrounded by non-similar elements in that region.

On this basis, we can explain the manner of movement for a non-carrier ant according to the following algorithm:

1. The *n*n* grid is broken into *l*l* subspaces.
2. An occupied cell is randomly selected in each subspace.
3. The amount of similarity of the existing element inside the selected cell with existing elements in the Moore Neighborhood is calculated.
4. Find a cell with the least similarity with adjacent element.
5. Guide the ant towards the identified cell.

The whole ant clustering process can be divided into three sections:

Initial Step: In this section, a number of small clusters appear.

Middle Step: In this section, small clusters are combined with each other to create larger clusters.

Final Step: In this stage, while the procedure of combining smaller clusters has completed, larger clusters have reached stability, and the Pheromone secretion of smaller clusters increases.

6. Analyzing the Proposed Algorithm Efficiency

In this section, results of the conducted tests on the proposed algorithm are presented. Although these tests have been carried out on small-sized data, however, test results show the positive performance of the clustering algorithm. In this test, 50 documents extracted from the Reuters 21578 database, which is one of the largest benchmarks in the field of data mining, were tested. The collection of selected documents involves gold, money, and gas topics. Algorithm parameters at the time of the test were determined based on the Table (1).

TABLE (1):

CONSTANT PARAMETERS FOR THE ANT-CLUSTERING ALGORITHM

Parameter	Parameter Value
Grid size	14*14
Number of ants	15
P_{Load}	0.6
P_{Drop}	0.4

The *F-Measure* index was used to numerically evaluate the proposed algorithm. This index utilizes the two concepts of precision and recall, which are among data recovery concepts. This is a value between 0 and 1, which a value near 1 is desirable.

$$P(i,j)=n_{ij}/n_j \quad , \qquad R(i,j)=n_{ij}/n_j$$

where:

n_i: number of class *i* data

n_j: number of cluster *j* data (the number obtained from the clustering algorithm)

n_{ij}: number of class *i* data in cluster *j*

Hence, the *F* index for class *i* and cluster *j* is calculated as follows:

$$F(i,j)=2\frac{recall(i,j)*precision(i,j)}{recall)i,j)+precision(i,j)}$$

Finally, the value of *F* for all clusters with *n* number of data is obtained as follows:

$$F=\sum_i \frac{n_i}{n}\max{}_j(F(i,j))$$

The amount of precision and recall obtained from the clustering process using the proposed algorithm on the mentioned databases is compared with the recall and precision obtained from the K-means and standard ant-clustering algorithm and presented in Table (2). Considering the uncertain nature of the ant-clustering algorithm, a new result is obtained from each instance of running the algorithm. Therefore, the values given in Table (2) are average values of all tests carried out using the algorithm.

TABLE (2):
COMPARISON BETWEEN CLUSTERING ALGORITHMS

	Recall			Precision		
	Our algorithm	Standard clustering	k-means	Our algorithm	Standard clustering	k-means
Gas	0.714	0.766	0.420	0.750	0.714	0.672
Gold	0.900	0.933	0.320	0.821	0.900	0.638
money	1.000	0.716	0.620	0.452	1.000	0.800
AVG.	0.871	0.805	0.453	0.674	0.871	0.703

Our experience gained from running the proposed algorithm showed that our algorithm in terms of time complexity acts better than the standard ant clustering algorithm, because the proposed algorithm is capable of reaching stability after 5000 stages and create suitable clusters, whilst the standard clustering algorithm requires 20000 stages in order to obtain the same clusters.

7. Conclusion

In this article, a novel algorithm was proposed for clustering by changing and creating an intelligent form of movement for ants in the standard ant-clustering algorithm. The proposed algorithm is an improved version of the ant-clustering algorithm presented by Lumer and Faeita. Creating purposeful ant movement provides timesaving for the algorithm run-time, and increases the quality of the obtained clusters.

Our observations from the proposed algorithm showed that this algorithm is preferable in two fields (quality of produced clusters, and algorithm run time) compared to the standard ant clustering algorithm and the K-means algorithm that is in fact one of the most powerful algorithms in the field of clustering.

8. References

[1] Berry, M. (ed.). "Survey of Text Mining: Clustering, Classification, and Retrieval". Springer, New York (2003)

[2] Bonabeau, E., Dorigo, M. & Theraulaz, G. "Swarm Intelligence: From Natural to Artificial Systems" Oxford University Press, New York (1999)

[3] Chen, L., Xu, X. & Chen, Y. "An adaptive ant colony clustering algorithm" In: Proceedings of the Third International Conference on Machine Learning and Cybernetics (ICMLC04), pp. 1387-1392 (2004)

[4] Chiou, Y-C. & Lan, L.W. "Genetic clustering algorithms. European Journal of A Modified Ant-Based Text Clustering Algorithm with Semantic Similarity Measure" Journal Of Systems Science And Systems Engineering Operational Research, 413-427. (2000)

[5] Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C. & Chétien, L. "The dynamics of collective sorting: robot-like ants and ant-like robots", In: Proceedings of the 1st International Conference on Simulation of Adaptive Behaviour, pp. 356-363, MIT Press, Cambridge, (1991)

[6] Handl, J., Knowles, J. & Dorigo, M. "On the performance of ant-based clustering". In: Proceedings of the Third International Conference on Hybrid Intelligent Systems. pp. 204-213, IOS Press (2003)

[7] Handl, J. & Meyer, B. "Improved ant-based clustering and sorting in a document retrieval interface". In: Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII), pp. 913–923, Springer-Verlag, Berlin (2002)

[8] Labroché, N. Monmarché, N. & Venturini, G. "A new clustering algorithm based on the chemical recognition system of ants". In: Proceedings of the 2002 European Conference on Artificial Intelligence, pp. 345-349 (2002)

[9] Lewis, D. Reuters-21578 text categorization test collection. Available via: http://www.daviddlewis.com/resources/testcollections/reuters21578. Cited Nov. 10, 2006.

[10] Lumer, E. & Faieta, B. "Diversity and adaption in populations of clustering ants". In: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour, MIT Press, Cambridge, MA (1994)

[11] Megaputer Intelligence Inc. "Online introduction to TextAnalystTM". Available via: http://www.megaputer.com/products/, Cited Nov. 12, 2006

[12] Ramos, V. & Abraham, A. ANTIDS: "self-organized ant based clustering model for intrusion detection system". In: Proceedings of The Fourth IEEE International Workshop on Soft Computing as Trans disciplinary Science and Technology (WSTST'05), pp. 977-986, Springer-Verlag, Berlin (2005)

[13] Sebastiani, F. "Machine learning in automated text categorization" ACM Computing Surveys, 34: 1-47. (2002)

[14] R. Shakerian, S. H. Kamali, M. Hedayati, M. Alipour , "Comparative Study of Ant Colony Optimization and Particle Swarm Optimization for Grid Scheduling", Journal of mathematics and computer Science(JMCS) 2011 Issue: 3 Pages: 469-474

[15] 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", Journal of mathematics and computer Science(JMCS) 2011 Volume: 2 (2011) Issue: 2 ,Pages: 329 - 336