A Novel Document Clustering Algorithm Using Squared Distance Optimization Through Genetic Algorithms

Harish Verma
Department of InformationTechnology
Indian Institute of Information Technology & Management
Gwalior

Eatesh Kandpal
Department of InformationTechnology
Indian Institute of Information Technology & Management
Gwalior

Bipul Pandey
Department of InformationTechnology
Indian Institute of Information Technology & Management
Gwalior

Joydip Dhar
Department of InformationTechnology
Indian Institute of Information Technology & Management
Gwalior

***Abstract*—K-Means Algorithm is most widely used algorithms in document clustering. However, it still suffer some shortcomings like random initialization, solution converges to local minima, and empty cluster formation. Genetic algorithm is often used for document clustering because of its global search and optimization ability over heuristic problems. In this paper, search ability of genetic algorithm has exploited with a modification from the general genetic algorithm by not using the random initial population.A new algorithm for population initialization is given in this paper and results are compared with k-means algorithm.** *(Abstract)*

Keywords: Genetic algorithm; optimization; Document clustering; k-means; mutation; crossover.

I. INTRODUCTION

With the explosion of data, clustering has gained its importance in almost all the fields like NeuroFuzzy Systems, Data Analysis, Linear Vector Quantization, Bio-informatics etc. Clustering is the process of grouping data into clusters, where objects within each cluster have high similarity, but are dissimilar to the objects in other clusters. Thus, clustering provides a better way to represent a huge amount of data into smaller groups called clusters sharing common attributes further we can use cluster centroid, medians or any other cluster representation for analytical processing of data according to our need. Developing some appropriate and efficient technique is required so that it can cope up with the huge amount of data to be dealt in real life problems.

Different algorithms have been developed over a period of time[2, 3, 4, 5]. These algorithms can be broadly classified into two agglomerative[6, 7, 8] and partitioning [9] methods based on the methodology used or into hierarchical or non-hierarchical solutions based on the structure of solution obtained. There has always been trade off between quality and complexity of clustering algorithm. Various approaches have been adopted to enhance the speed of the algorithms by better modeling [11,12, 13]. Genetic Algorithms also have been applied to an extent for the problem of clustering [1, 5, 15, 17]. Most widely used partitioning algorithms use greedy approach. A widely used example of greedy approach is k-means[5] algorithm. However k-means might converge to a local optimum and its result depends on the initialization process, which randomly generates the initial clustering solution. In other words, different runs of KM on the same input data might produce different results. Probability of getting good quality clusters depend on the initial clustering solution.

Unlike other common approaches we have proposed a modified genetic algorithm in which the generation of initial population is not random. We create the initial population by measuring the similarity between the documents on the basis of their sum of squared distances from the previously selected documents.

We found that in general use, the performance of the algorithm was Better than that of the k-means algorithm. This paper is organized as follows. In section II our proposed genetic algorithm is explained. Implementation details of our work are listed in section III. Section IV contains result analysis. Finally, we conclude in Section V.

II. THE ALGORITHM

In this section we will discuss the algorithm in details with elaboration of each step. We have used concept of genetic algorithm which works iteratively and refines solution in every iteration. The steps below show the pseudo code of GA [16].

```
t = 0;
Initialize P(t);
Evaluate P(t);
While not (termination condition)
begin
t=t+1;
Select P(t) from P(t - 1);
Recombine pairs in P(t);
Mutate P(t);
Evaluate P(t);
End
```

Figure 1. Goldberg's Pseudo-code of Genetic Algorithms

In our proposed algorithm the step of population initialization has been changed. Various steps of the proposed algorithm with the detailed description are given below.

A. String Representation

Each chromosome is encoded as a sequence of real numbers representing the K cluster centroids. For an N-dimensional space, the length of a chromosome is given by N*K words, where the first N positions (or, genes) represent the N dimensions of the first cluster centroid vector, the next N positions represent those of the second cluster centroid vector, and so on.

B. Population Initialization

In the proposed algorithm we have employed a novel algorithm for the purpose of population initialization which renders us many advantages over previous algorithms. In this approach instead of random population initialization we are using the following steps**:**

1 Choose a document, add it to a list and add that list to store_ seeds.

2 Find the document at largest distance from the first selected document, add it to a list and add that list tothe store_seeds.

3 Store the documents having sum of distances larger than a cutoff distance from selected seeds.(we set cutoff distance keeping in mind that the total number of qualifying documents is not smaller than our regulating parameter R).

4 Store the squared sum of distances, from the selected seeds, of the qualified documents in step 3.

5 Define a variable 'Z' called regulating variable as according to the type of data we are processing.

6 Calculate the value of the expression E = Z*(linear sum of distances) - (1/Z)*(squared sum of distances) for each document.

7 Choose top 'R' documents according to their corresponding values of 'E' and add the selected documents in a list.

8 Add list to vector store_seeds.

9 Repeat steps 3-7 unless number of lists in vector store_seeds = K.

10 Apply backtracking for all the lists in store_seeds making sure that at one time we select only one element from a single list and thus find K-documents in each full run of backtracking algorithm.

11 The K-seeds thus obtained form a solution, and at the end of complete procedure total number of solutions will be power(R,k-2). Thus according to need of number of solutions for initial population we can set the value of variable 'R'.

12 Consider the solutions, thus obtained, as initial population for the application of genetic algorithm on them.

This algorithm generates the many sets of seeds, each set called a solution, which are more likely to be in different clusters. This way we get the seeds which can represent their clusters better than any random seed does because if we select random seeds probability of two seeds being around each other and thus representing same cluster is more. As we could generate a number of solutions in the algorithm we can use this more optimized population as initial population for the application of genetic algorithm so that results get optimized further.

C. Fitness calculation

The proposed algorithm works on any given fitness function. The fitness function may be changed according to user needs. Although, it is expected that fitness function should use document vector notation. In this paper we have used two different fitness functions for our test purposes. One fitness function seeks to create cluster having documents within it more and more similar to centroid of the cluster while the other tries to separate the given cluster from the entire collection. The algorithm tries to optimize the fitness function, thereby giving us the desired results.

D. Selection

The solutions (individuals) are selected from the mating pool in the process according to their fitness values, those directed the principle by Charles Darwin of the fittest concept of natural genetic systems. In this paper, the proportional selection strategy is adopted; a chromosome is assigned a number of copies, which is proportional to its fitness in the

population that goes into the mating pool for further genetic operations. Roulette wheel selection is one common example in which type of selection is stochastic. Apart from this one may also use Deterministic Sampling, Stochastic Tournament Selection or Remainder Stochastic Sampling.

E. Cross over

Crossover or recombination is a probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. Crossover is done with the hope that offspring will contain good parts of old chromosomes. The cross over operation is described below.

CrossOver(p,q)

A ← Centroid vectors in parent1

B ← Centroid vectors in parent2

With a probability P_c and for a chromosome of length l, a random crossover point, is generated between the range [0, l]. The portions of the parent chromosomes lying to the right of the crossover point are exchanged to produce two off spring. The value of P_c is taken as 0.5.

F. Mutation

Mutation is also a probabilistic process applied to each off spring individually after the crossover which modifies each gene with a low probability, typically value in the range 0.001 and 0.01.

In this algorithm we perform mutation by moving all the centroid vector of the selected chromosomes randomly in some direction by a random amount. This amount can be a maximum of 0.1 of the size of the input space. If the centroid vector lies outside the input space for any dimension after movement, then we undo that movement for that dimension. Mutation is given by

Mutation (p)

With probability P_m, generate a random number x between [0-1] with uniform distribution.

For each centroid vector

Modify (add) the value at each gene position by value x.

III. IMPLEMENTATION DETAILS

To test and compare the algorithm we have coded it in java. The rest of this section describes about the input dataset, comparison with k-means algorithm applying two different fitness functions mentioned in this paper**.**

A. Data Sets

TABLE1. REUTERS-21578 DATASET

Data	Source	No of documents	No of classes
Re0	Reuters-21578	1504	13
Re1	Reuters-21578	1657	25

B. Entropy

Entropy measure is used to determine cluster quality, it employs class label of a document assigned to a cluster for this purpose. Entropy provides us the information about how the documents from various classes are distributed among distinct clusters.

The entropy will be zero if the solution contains clusters having documents from same class and such solution is considered as an ideal solution. Thus smaller the value of entropy better is the solution.

Given a particular cluster S_r of size N_r, the entropy [10] of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log q}\sum_{i=1}^{q}\frac{N_r^i}{N_r}\log\left(\frac{N_r^i}{N_r}\right)$$

where q is the total number of classes available in the dataset, and N_r^i is the number of documents assigned to the r_{th} cluster belonging to the i_{th} class. The total entropy will be given by the following equation

$$Entropy = \sum_{r=1}^{k}\frac{N_r}{N}E(S_r)$$

IV. RESULTS

We have compared k-means algorithm with proposed genetic algorithm for document clustering. Comparison is done using two different fitness functions. Fitness function F_1 is used to maximize the similarity of a document within a cluster with its cluster centroid while fitness function F_2 tries to separate the document of each cluster from centroid vector of the entire collection. Mathematically these fitness functions can be represented as:

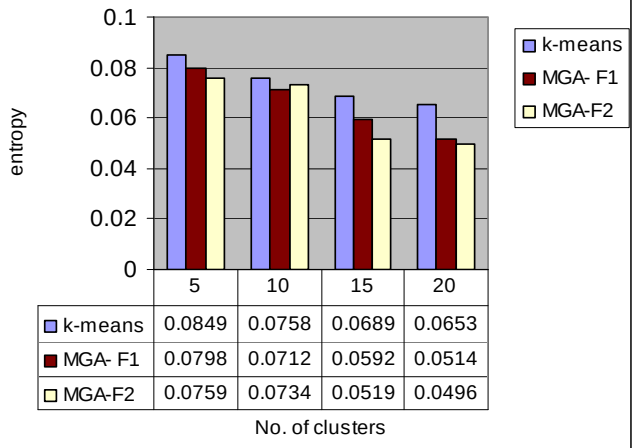
$$Maximize\ F1 = \sum_{r=1}\sum_{d\ \epsilon\ S} Cos\ (d_i, C_r)$$

Minimize F2

= Error! Bookmark not defined. Error! Bookmark no

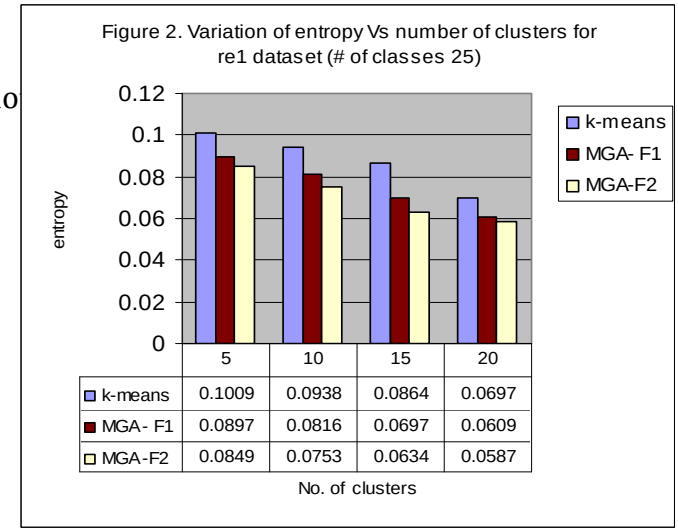
Results have been prepared by rigorous execution of programs over different data sets. Results are represented in form of the bar-graphs. Each graph consists of 3 bars of different colors blue, brown and yellow where blue color bar shows entropy value of the clustering solution obtained using k-means algorithm. Brown and yellow color bars show entropy value of the clustering solution obtained using proposed algorithm with fitness function F_1 and F_2 respectively. Experimental results are shown in the form of graphs [see Figure 1-2].The first is obtained using dataset re0 [16] and the second one is obtained using dataset re1 [16]. The results prove that the entropy values obtained using proposed algorithm is smaller, hence better then k-means algorithm.

Figure 1. Variation of entropy Vs number of clusters for re0 dataset (# of classes 13)



	5	10	15	20
k-means	0.0849	0.0758	0.0689	0.0653
MGA- F1	0.0798	0.0712	0.0592	0.0514
MGA-F2	0.0759	0.0734	0.0519	0.0496

Figure 2. Variation of entropy Vs number of clusters for re1 dataset (# of classes 25)



	5	10	15	20
k-means	0.1009	0.0938	0.0864	0.0697
MGA- F1	0.0897	0.0816	0.0697	0.0609
MGA-F2	0.0849	0.0753	0.0634	0.0587

V. CONCLUSION

In our proposed novel approach we found that algorithm is working better than k-means algorithm as it has less probability to be trapped in local optimal solutions. Our newly proposed algorithm for population initialization step in genetic algorithm has substantially decreased calculation complexity and it also increases the effectiveness of result as initial population was not random. Furthermore less number of iterations is required during execution to converge to a global optimal solution. With the increase in the number of clusters the performance of algorithm gets improved.

REFERENCES

[1] R. Kala,A. Shukla and R. Tiwari, "A novel approach toclustering using generic algorithm," IJERIA, 2010.

[2] X.Cui,T.E. Potok and P. Palathingal, "Document clustering using particle swarm optimization," Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE , vol., no., pp. 185-191, 8-10 June 2005.

[3] T. Kanungo, D. M. Mount and N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 881-892, July 2002.

[4] M. Mahdavi and H. Abolhassani, "Harmony k -means algorithm for document clustering," Data Mining and Knowledge Discovery 2009.

[5] A.K. Jain and R. C. Dubes, " Algorithms for Clustering Data," Prentice Hall, 1988.

[6] S. Guha, R. Rastogi, and K. Shim, "Rock: A robust clustering algorithm for categorical attributes," Information Systems, vol. 25, no. 5, pp. 345-366, 2000.

[7] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," SIGMOD Rec., vol. 27, no. 2, pp. 73-84, 1998.

[8] G. Karypis, Eui, and V. K. News, "Chameleon: Hierarchical clustering using dynamic modeling," Computer, vol. 32, no. 8, pp. 68-75, 1999

[9] E. H. Han, G. Karypis, V. Kumar, and B. Mobasher, "Hypergraph based clustering in high-dimensional data sets: A summary of results," Data Engineering Bulletin, vol. 21, no. 1, pp. 15-22, 1998.

[10] Y. Zhao and G. Karypis, "Criterion functions for document clustering: Experiments and analysis," Technical Report #01-40, University of Minnesota, 2001

[11] K. Alsabti,S. Ranka and V. Singh, "An Efficient K-Means Clustering Algorithm", Proceedings of IPPS/SPDP Workshop on High Performance Data Mining, 1998.

[12] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y.Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEETrans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 881–892, July 2002.

[13] K. Means, M. Data, B. Zhang, B. Zhang, M. Hsu, M. Hsu, U. Dayal, and U. Dayal, "K-harmonic means - a data clustering algorithm, 1999.

[14] Q. Chen, J. Han, Y. Lai, W. He, andK. Mao,"Clustering Problem Using Adaptive Genetic Algorithm,"ICNC 2005, Springer Lecture Notes in Computer Science, Vol. 3612, 2005, pp. 782 – 786.

[15] Y. Lu, S. Lu, F. Fotouhi, Y. Deng, and S. Brown, "FGKA: A Fast Genetic K-Means Clustering Algorithm," ACM Symposium on Applied Computing, 2004.

[16] D. Goldberg, "Computer-Aided Gas Pipeline Operation Using Genetic Algorithms And Rule Learning," Ph.D. thesis, University of Michigan, Ann Arbor, 1983.

[17] U. Maulik and S. Bandyopadhyay, "Genetic algorithm-based clustering technique," Pattern Recognition, Volume 33, Issue 9, September 2000.

AUTHORS PROFILE

Harish Verma
Department of Information Technology
Indian Institute of Information Technology and Management Gwalior,
Gwalior, Madhya Pradesh, India

EateshKandpal
Department of InformationTechnology
Indian Institute of Information Technology
& Management
Gwalior

BipulPandey
Department of InformationTechnology
Indian Institute of Information Technology
& Management
Gwalior

Joydip Dhar
Department of InformationTechnology
Indian Institute of Information Technology & Management
Gwalior