Text Documents Clustering using Genetic Algorithm and Discrete Differential Evolution

Yogesh Kumar Meena
Hindustan Institute of Technology and Management
Department of Information Technology, Agra, India

Shashank
Hindustan Institute of Technology and Management
Department of Computer Science, Agra, India

Vibhav Prakash Singh
Anand Engineering College,
Department of Information Technology,
Agra, India

ABSTRACT
Clustering in data mining is a discovery process that groups a set of documents such that documents within a cluster have high similarity while documents in different clusters have low similarity. Existing clustering method like K-means is a popular method but its results are based on choice of cluster centers so it easily results in local optimization. Genetic Algorithm (GA) is an optimization method which can be applied for finding out the best cluster centers easily. But sometimes it takes more iteration for finding best cluster centers. In this paper, we use features of GA with the features of Discrete Differential Evolution (DDE) to solve text documents clustering problem. To test the efficiency of our algorithm we have taken sample database of Reuters-21578. From the experimental results, it is clear that our algorithm performs better than GA and DDE.

Keywords
Genetic Algorithm, Discrete Differential Evolution, Document Clustering.

1. INTRODUCTION
Clustering analysis is an important field of artificial intelligence and data mining. The basic idea is to use words and characters from the documents for checking degree of similarities among documents and cluster those documents without prior knowledge. In clustering methods, the documents are converted into the vectors and similarities among them are calculated. According to the similarities the documents are grouped into a number of clusters such that intra-cluster documents are more similar than inter-cluster documents. These discovered clusters can be used to explain the characteristics of the underlying data distribution, and thus serve as the foundation for other data mining and analysis techniques. The applications of clustering include characterization of different customer groups based upon purchasing patterns, categorization of documents on the World Wide Web, grouping of spatial locations prone to earth quakes from seismological data, etc.

Wei Jian-Xiang[1] introduces that clustering algorithms can be broadly divided into two basic categories: hierarchical and non-hierarchical. K-means is a most widely used algorithm. It is the type of non-hierarchical, faster than the hierarchical algorithm. But the disadvantage is that the final result is based on the initial selection of cluster centers. Its objective function has local minimum, and it is greedy algorithm, so it easily results in local optimization. In this research there is a new dynamic method based on genetic algorithm (GA). Genetic algorithm is a global convergence algorithm, which can find the best cluster centers easily. Among the traditional document clustering methods, the document similar matrix is a sparse matrix. There are some improvement of formulas of the traditional method and some improvement on genetic algorithm. All individuals are encoded by floating-point number and the sum of mean square deviation of intra-class distance is adopted as the objective function. But some problems need to be solved, such as: For realistic instances there may be hundreds of unique keywords, so each individual is a vector of several hundred real numbers. And it is known that the size of an individual needed for GA to evolve satisfactory solutions grows exponentially with the length of the representation. So, there should be a way to reduce the dimension of clustering space so that the algorithm can be applied to large dataset.

A. Casillas [3] presents a genetic algorithm that deals with document clustering. This algorithm calculates an approximation of the optimum k value, and solves the best grouping of the documents into these k clusters. He has evaluated this algorithm with sets of documents that are the output of a query in a search engine.

K. Premalatha [5] presents the document clustering based on Genetic algorithm with Simultaneous mutation operator and Ranked mutation rate. The mutation operation is significant to the success of genetic algorithms since it expands the search directions and avoids convergence to local optima. In each stage of the genetic process in a problem, may involve aptly different mutation operators for best results. In simultaneous mutation the genetic algorithm concurrently uses several mutation operators in producing the next generation. The mutation ratio of each operator changes according to assessment from the respective offspring it produces. In ranked scheme, it adapts the mutation rate on the chromosome based on the fitness rank of the earlier population. Experiments results are examined with document corpus. It demonstrates that the proposed algorithm statistically outperforms the Simple GA and K-Means. Our work proposes that Genetic Algorithm is an optimization algorithm and can find best cluster centers easily. But sometimes GA takes more iteration for finding best cluster centers. Our objective is to reduce the iteration in GA. Therefore we applied Genetic Algorithm with another optimization technique Discrete Differential Evolution such that in the odd iteration GA is applied, in the even iteration DDE is applied. The combined form of these two algorithms gives better results in less iteration. The rest of the paper is organized as follows. Section 2 gives an overview of text documents clustering

preprocessing. Section 3 presents text documents clustering by combined form of GA and DDE. Section 4 presents the simulation of our algorithm. Finally section 5 presents the conclusion and future scope.

2. TEXT DOCUMENTS CLUSTERING PREPROCESSING

In this research, Reuter-21578 database is used for implementation. We select some documents apply Keyword-Extraction Algorithm.

2.1 Keyword-Extraction-Algorithm

1. Select n text documents for clustering.
2. Merge these n documents into one document.
3. Apply Tokenize process(the process of splitting sentences into word tokens).
4. Remove Stopwords (a, an, the, is, am, are, that, these,....)
5. Apply stemming process like Portal Stemming Algorithm and find m distinct keywords.

2.2 Document vectors

After extracting the keywords we convert the documents in vector form. When we have a number of keywords from given documents, we calculate the frequencies of each word in each documents. These frequencies are used for documents vectorization. Let there are n documents and in which there are m distinct keywords. So each document has m words frequencies. These m frequencies can be considered m co-ordinates of one document and n documents can be drawn into m dimensional space. After drawing the n documents into m dimensional space, the similarities between documents are calculated. The distances among documents are considered as similarities among documents. For maximum similarity between two documents, the distance between them should be minimum. So two matrices can be made i.e. frequency matrix and distance matrix.

i. m × n frequency matrix: This matrix tells the frequencies of m keywords into n documents.
ii. n × n distance matrix: This matrix tells the distances among n documents in m dimensional space If there are two documents d1, d2 in which frequencies of m words are (x1, x2, …, xm) and (y1, y2, …, ym), then the distance between d1 and d2 is
D= [(x1-y1) +(x2-y2) +(x3-y3) +…+ (xm - ym)]1/2

3. TEXT DOCUMENTS CLUSTERING BY COMBINED FORM OF GA AND DDE

3.1 Making Initial Population

For clustering documents, k documents are randomly selected from n documents. These k documents are k centers for k clusters. After this, for each of the remaining documents, a document is assigned to the cluster to which it is most similar, based on the distance between the document and the cluster center. These k centers are considered as one chromosome of population. The above process is continued for making all chromosomes of population.
E.g. if n=15, k=5 then chromosome is (2, 5, 8, 10, 1).

3.2 Fitness Function

Fitness of the chromosome can be calculated by following formula.

$$F = \sum_{i=1}^{k} \sum_{x_t \in c_i} (x_t - x_i^*)^2 / n_i$$

Where x_i^* is the center of cluster ci, x_t is the document in the cluster ci, ni is the no. of documents in cluster ci. It is clear that the smaller the value of F is, the better the clustering effect is.

3.3. Crossover Operator of GA and DDE

In our algorithm the chromosome has k centers and these k centers are distinct. During crossover there may be possibility of common centers in one new chromosome. For solving this problem first find out common centers in two chromosomes and then apply crossover operator on uncommon centers.
E.g. if two chromosomes of 5 centers are (1, 4, 6, 7, 9) and (5, 11, 10, 8, 1). First find out common center i.e. 1. Now apply crossover operator on (4, 6, 7, 9) and (5, 11, 10, 8).

3.4 Mutation Operator of GA and DDE

During mutation there may be possibility of common centers in new chromosome. For solving this problem change the value of gene other than given centers in the chromosome.
E.g. if one chromosome of 5 centers is (1, 4, 6, 7, 9) and we want to update second gene's value, so we will replace 4 by value v ϵ {1, 2, … , n}-{1, 4, 6, 7, 9}.

3.5 Proposed Algorithm

In our algorithm, first we initialize the population. Then we call GA function and DDE function one by one for making new population. In GA function, Crossover and Mutation operators are applied. After applying the operators, if new chromosomes are better, then these chromosomes are added to the new population otherwise old solutions are added into the new population. When DDE is called, first we find out global best chromosome from the old population, then apply Mutation operator on that best solution. After then, we apply Crossover between old population and global best chromosome and make temporary population. Then we compare old population chromosomes with temporary population chromosomes and make new population with better chromosomes. The steps of algorithms are given below. In the algorithm (pc)GA is the Crossover probability for Genetic Algorithm, (pc)DDE is the Crossover probability for Discrete Differential Algorithm, (pm)GA is the Mutation probability for Genetic Algorithm, (pm)DDE is the Mutation probability for Discrete Differential Algorithm, iter is the no. of iterations, and np is the population size.

3.5.1 Algorithm

1. for i ←1 to np (population initialization) do
a) randomly select k documents from n documents and consider them k centers of k clusters. These k centers are one chromosome of initial population.
b) for every (n-k) documents do
   i. find distances from k centers by n×n matrix.

ii. cluster document with any center according to minimum distance.

c) Find out the fitness value of this chromosome by fitness formula.

2. consider initial population as the oldpop.
3. for i ←1 to iter do
   a) if i is odd then
      i. call GA_function
   b) else
      i. call DDE_function

4. Find out best chromosome from the last population.

This chromosome has best clusters of documents.

3.5.2 GA_function

1. for j=1 to np/2 (Make new population by Crossover) do
   a) select two chromosomes from oldop randomly.
   b) if crossover probability (pc)GA is satisfied then
      i. apply Crossover between chromosomes and find out two offspring.
      ii. calculate fitness values of two offspring by fitness formula.
      iii. select best two among old chromosomes and two offspring.
      iv. add them into newpop.
   c) else
      i. add the chromosomes into newpop.
2. for every chromosome in newpop do
   a. if mutation probability (pm)GA is satisfied then
      i) apply Mutation upon chromosome and find out offspring.
      ii) calculate fitness value of offspring by fitness formula.
      iii) select best one between old chromosome and offspring.
      iv) update newpop by best one.
3. update oldpop by newpop and empty newpop.

3.5.3 DDE_function

1. select best chromosome as a gbest from oldpop.
2. if mutation probability (pm)DDE is satisfied then
   a. apply Mutation upon gbest and generate offspring.
   b. update gbest by offspring.
3. for every chromosome in the oldpop do
   a. if crossover probability (pc)DDE is satisfied then
      i. apply Crossover between particular chromosome and gbest and generate two offspring.
      ii. calculate fitness values of two offspring by fitness formula.
      iii. select best of among old chromosome and two offspring.
      iv. add that one into newpop.
   b. else
      i. select best between particular chromosome and gbest.
      ii. add that one into newpop.
4. update oldpop by newpop and empty newpop.

4. SIMULATION & RESULT OF CLUSTERING ALGORITHM

We took Reuter21578 Documents for experiment. We took 500 documents and cluster them into k clusters. We took population size 50, Crossover probability for GA 0.8, Mutation probability for GA 0.1, Crossover probability for DDE 0.3, and Mutation probability for DDE 0.8. We run our algorithm and compare results with GA and DDE. In the table we can see that combination of GA and DDE gives better results in less iteration.

Table 1. Text Documents Clustering of 100 Documents in 3 Clusters

Algorithm	No. of iterations from which to 1000 iterations fitness value is same	Best Chromosome	Fitness Value
GA	145	78 84 3	165.990005
DDE	414	78 84 3	165.990005
GA & DDE	66	78 84 3	165.990005

Table 2. Text Documents Clustering of 250 Documents in 5 Clusters

Algorithm	No. of iterations from which to 1000 iterations fitness value is same	Best Chromosome	Fitness Value
GA	153	78 104 3 195 84	221.175993
DDE	894	84 182 109 78 15	223.076004
GA & DDE	117	78 104 3 195 84	221.175993

Table 3. Text Documents Clustering of 500 Documents in 7 Clusters

Algorithm	No. of iterations from which to 1000 iterations fitness value is same	Best Chromosome	Fitness Value
GA	496	3 294 99 78 109 372 195	235.429993
DDE	392	3 294 99 78 109 372 195	235.429993
GA & DDE	382	195 99 15 104 78 361 372	235.257996

5. CONCLUSIONS AND FUTURE WORK

k-means is a popular method for text documents clustering but its results are based on choice of cluster centers so it easily results in local optimization. So for optimized results we can use Genetic Algorithm and Discrete Differential Evolution. But both algorithms require more iteration for finding better clusters. So we applied combination of both algorithms, which can give better clusters in less iteration. We can see the results in the above tables in which better fitness values are found out by applying the combination of GA & DDE in less iteration.

In the future, we can make more efficient algorithm so that it can be used for large datasets efficiently. For making efficient algorithm, we will try to reduce the dimensions of text documents by using semantic of the words.

6. REFERENCES

[1] Wei Jian-Xiang, Liu Huai, Sun Yue-hong, Su Xin-Ning. "Application of Genetic Algorithm in Document Clustering", International Conference on Information Technology and Computer Science, Vol. 01, pp. 145-148, 2009.

[2] George Karypis, Eui-Hong(Sam)Han, Vipin Kumar. "CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling", IEEE Computer Society, Vol. 32, Issue. 8, pp. 68-75, 1999.

[3] A. Casillas, M. T. Gonz´alez de Lena, and R. Mart. "Document Clustering into an unknown number of clusters using a Genetic Algorithm", Lecture Notes in Computer Science, Vol. 2807/2003, pp. 43-49, 2003.

[4] Calinski & Harabasz. "A Dendrite Method for Cluster Analysis", Communications in Statistics, Vol. 3(1), pp. 1-27, 1974.

[5] K. Premalatha, A.M. Natarajan. "Genetic Algorithm for Documents Clustering with Simultaneous and Ranked Mutation", Modern Applied Science, Vol. 3, No. 2, 2009.

[6] Sheng ZHONG, Zhiwei LIN, Beihai ZHANG, Chengcheng YU. "Genetic Algorithm on Documents Clustering", Journal of Computational Information Systems, Vol. 3, pp. 1063-1068, 2008.

[7] Quan-Ke Pan, M. Fatih Tasgetiren, Yun-Chia Liang. "A Discrete Differential Evolution Algorithm for the permutation flowshop scheduling problem", GECCO, pp. 126-133, 2007.

[8] Chong Su, Qingcai Chen, Xiaolong Wang, Xianjun Meng. "Text Clustering Approach Based on Maximal Frequent Term Sets", IEEE International Conference, pp. 1551-1556, 2009.

[9] Jeffrey L. Solka, "Text Data Mining: Theory and Methods", Statistics Surveys, Vol. 2, (2008), pp. 94-112 (electronic), 2007.

[10] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", 2nd Edition, Elsevier, 2008.

[11] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", 1st edition, Pearson, 2008.

[12] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[13] ftp://ftp.cs.cornell.edu/pub/smart/ (accessed on April 13, 2006)

[14] C. Xiaohui, T.E. Potok, P. Palathingal Document Clustering using Particle Swarm Optimization, IEEE Swarm Intelligence Symposium,The Westin Pasadena, Pasadena, California, 2005.

[15] S. Das, A. Konar, U. K. Chakraborty, Two Improved Differential Evolution Schemes for Faster Global Search in ACM-SIGEVO Proceedings of Genetic and Evolutionary Computation Conference (GECCO-2005), Washington DC, June, 2005.

[16] K.Deb, A. Anand and D. Joshi (2002). A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization, Evolutionary computation, 10(4), pp. 371 – 395.

[17] A. Ratnaweera, K, S. Halgamuge,: Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. IEEE Trans. on Evolutionary Computation (2004) 8(3): 240-254.

[18] TREC. 1999. Text Retrieval Conference. http://trec.nist.gov ((accessed on April 13, 2006).