A Multi Criteria Document Clustering Approach Using Genetic Algorithm

D. Mustafi, G. Sahoo and A. Mustafi

Abstract In this work we present a multi criteria based clustering algorithm and demonstrate its usefulness in clustering documents. The algorithm proposes various metrices to judge the veracity of the clusters formed and then finds a near optimal solution that ensures good fitness scores for the all metrices. In view of the complexity of optimizing multiple clustering goals using classical optimization techniques, the paper proposes the use of an evolutionary strategy in the form of Genetic algorithm to quickly find a near optimal cluster set that satisfies all the cluster goodness criteria. The use of Genetic algorithm also inherently allows us to overcome the problem of converging to locally optimal solutions and find a global optima. The results obtained using the proposed algorithm have been compared with the outputs from standard classical algorithms and the performances have been compared.

Keywords Nearest neighbor · Crossover · Genetic algorithm · Chromosomes

1 Introduction

The unprecedented proliferation of digital documents into all aspects of our lives provides a great software challenge. Many large document repositories often require meaningful segregation of the documents into sets based on similarity metrices to facilitate retrieval while not sacrificing speed. While some amount of manual interference is acceptable, automated document clustering algorithms play a

D. Mustafi (✉) · G. Sahoo · A. Mustafi
Department of CSE, Birla Institute of Technology, Mesra 835215, India
e-mail: debjani.mustafi@bitmesra.ac.in

G. Sahoo
e-mail: gsahoo@bitmesra.ac.in

A. Mustafi
e-mail: abhijit@bitmesra.ac.in

H.S. Behera and D.P. Mohapatra (eds.), *Computational Intelligence in Data Mining—Volume 1*, Advances in Intelligent Systems and Computing 410, DOI 10.1007/978-81-322-2734-2_25

significant role in lubricating this process. Many different clustering paradigms have been proposed in literature, but the partition based K-Means [8] method and its variants have remained the principal clustering algorithms used in the vast majority of cases. With its excellent speed and adaptability the K-Means performs extremely well with proper tuning. However, as has been documented elsewhere [6] a badly tuned K-means algorithm is prone to local minima convergence and incorrect cluster recognition. Also, in the case of document clustering it is often required to use multiple similarity parameters to ensure that the clusters are well separated. This is a challenge as documents tend to share a lot of common words even when their topics are completely different. An N-gram approach may be a solution to this but comes at the cost of huge computational load. It is believed that evolutionary algorithms can provide a viable alternative while clustering large document corpuses because of their ability to work with multiple objectives simultaneously and their exploration capabilities which negate local convergence. In this paper we propose a multi criteria GA based algorithm [6] for clustering documents, which performs excellently in comparison to the standard K-means algorithm and does not require exceptionally high end hardware resources.

2 Mathematical Model

The clustering of text data documents requires the documents to be represented in the form of vectors traditionally known as the Vectors space model [2]. The model takes into consideration the frequency of a term in a document and assigns a suitable weight for the same which is referred to as term frequency (TF). The assessment of more relevant terms is also very important for correct prediction of results and so along with the term frequency the sparsity of terms among the documents is also considered. This is commonly known as the Inverse Document Frequency (IDF) factor. Thus, the significance of a term in a document can be estimated by TF-IDF value [1]. Cosine similarity [5] is a commonly used similarity measure which measures the similarity of two documents in the vector space. It is defined as $\cos\left(\overrightarrow{d_1}, \overrightarrow{d_2}\right) = \overrightarrow{d_1} \cdot \overrightarrow{d_2} / \left|\left|\overrightarrow{d_1}\right|\right| \cdot \left|\left|\overrightarrow{d_2}\right|\right|$ where $\overrightarrow{d_1}, \overrightarrow{d_2}$ are two document vectors, and "$\cdot$" denotes the dot product and $\|\cdot\|$ denotes the norm of a vector. Documents found to be in close proximity of each other are then grouped to be in the same cluster, where the number of clusters to be formed is usually a user driven input. Any crisp clustering algorithm is considered to be valid if and only if it satisfies the following conditions:

1. A document vector must be assigned to only a single cluster.
2. A cluster must not be empty.
3. All documents must be assigned a cluster.

3 Proposed Method

3.1 Basic Principles and Flow Chart of GA

GA is a popular evolutionary search optimization technique where parameters are encoded in the form of strings known as chromosomes. The collection of these chromosomes is referred to as population [6]. Each chromosomes is associated with a fitness value indicative of its goodness in the search space with regards to a fitness function. The chromosomes [4] with best fitness value are selected as the population in the next generation following the Darwin's theory of survival for the fittest. Crossover [3] and mutation operations are performed to generate off springs which improves the overall health of the chromosomes over generations. The basic flow chart [9] of GA is shown in the Fig. 1.

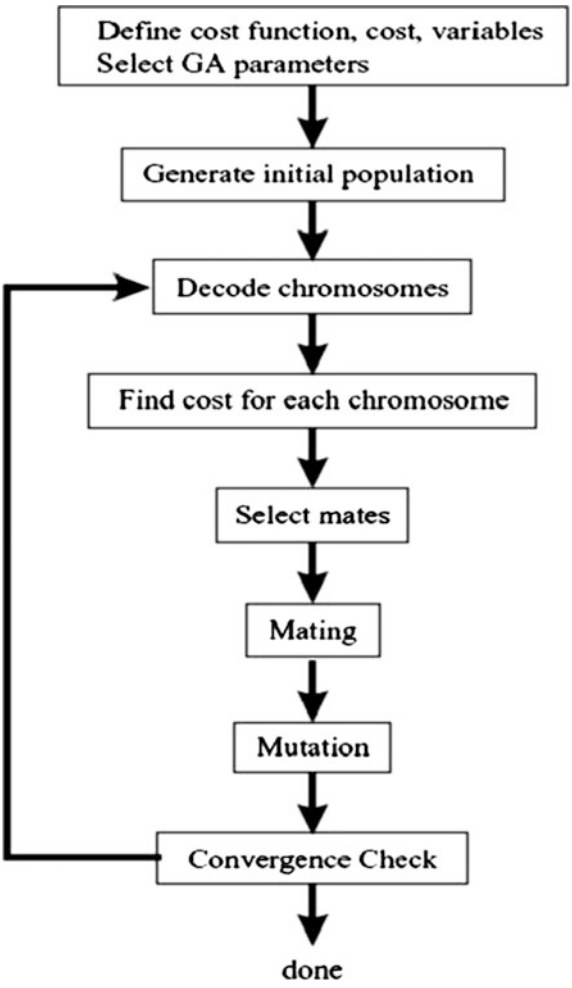

Fig. 1 Flowchart of simple GA

3.2 Chromosome Representation for Clustering

Every document d_i in the data set D has been represented in the form of TF-IDF vector which is then reduced to "p" Singular Value Decomposed (SVD) components [7]. Our chromosomes represent a collection of "k" centroids, each having "p" numbers of components, as shown in Fig. 2. Each chromosome thus represents an initial estimate of "k" centroids, against which the clustering is performed. In each iteration, the GA simply finds better and better centroids till the termination condition is reached.

3.3 The Choice of the Fitness Function

For the purpose of our investigation, we have considered purely internal cluster purity measures to evaluate the goodness of the clusters formed. Our motivation for the same is the unsupervised nature of the clustering problem, where apriori inputs may not be available in many cases. Such measures focus on reducing the intra cluster scatter while ensuring that inter cluster distances are maximized. The "nearest neighbour separation" measure has been cited in literature as a means of measuring the nearest separation between clusters. Maximizing such a measure would allow diffuse or intricately mixed cluster points to be separated in a more appropriate manner. However, it was also felt that this one metrics would not be sufficient in the creation of good clusters.

In this paper we propose a heuristic to reduce the time required in finding the distance of nearest separation between two clusters. The algorithm used for finding the separation is presented in Algorithm [1].

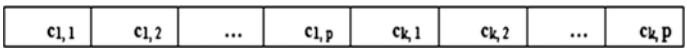

Fig. 2 Chromosome representation

Algorithm 1. Algorithm to find the nearest separation between two clusters

FIND NEAREST SEPARATION (C1, C2, T) // C1 and C2 are two clusters T is a scalar threshold between 0 and 1

1. D1 = {distance from centroid of C1 to all points in C1}
2. D2 = {distance from centroid of C2 to all points in C2}
3. PT1 = {all points in C1 with a distance greater than T*max(D1) from centroid of C1}
4. PT2 = {all points in C2 with a distance greater than T*max(D2) from centroid of C1}
5. NNS = find minimum distance between any pair of points (i, j) where i ∈ PT1 and j∈ PT2
6. return NNS

Based on this, we propose the use of four different metrices to measure the cluster validity of the clusters returned by the algorithm. These measures are presented in equations [1–4]

$$obj1 = \frac{1}{k}\sum_{i=1}^{k}\{\cos(c_i, c_j)\} \tag{1}$$

$$obj2 = \frac{1}{k}\sum_{i=1}^{k}\left[\frac{1}{n}\sum_{d_j \in C_i}\cos(d_j, c_i)\right] \tag{2}$$

$$obj3 = \min\left[\cos\left\{\sum_{d_i > T*\max(D_i)}\cos(d_i, c_i), \sum_{d_j > T*\max(D_j)}\cos(d_j, c_j)\right\}\right] \tag{3}$$

$$obj4 = k^{\wedge}[|\text{No. of unique clusters in assignments}|] \tag{4}$$

The final fitness function has been obtained by incorporating the objective functions defined in equations [1–4]. Thus the fitness function for the ith chromosome is given as

$$F_i = \alpha\frac{obj2}{obj1 * obj4} + \beta\frac{1}{obj3} \tag{5}$$

where α and β are two scalar quantities. It is to be observed that while the first component, controls the inter cluster distances and intra cluster scatter, the second component controls the nearest neighbour separation. By choosing different values of the scalar parameters α and β the user can control the amount of diffusivity between the various clusters.

4 Implementation and Results

To demonstrate the performance of the proposed method two different test beds were used. The first test bed contained a synthetically generated dataset having high diffusivity, while the second test bed contained a collection of documents from the BBC and BBC Sports corpus. The details about the datasets used for the experiments is shown in Table 1. The simulation of the synthetic datasets and the text processing used in the implementation haven been carried out in Python, while the actual clustering algorithms were implemented using Matlab. The results obtained were compared with a standard implementation of K-Means with the maximum number of iterations set to 300 and number of iterations being set to 10 (to prevent local convergence).

4.1 Results on the Synthetic Datasets

In the case of the synthetic dataset, the clusters can be seen to be quite diffuse (refer to Figs. 3, 4, 5 and 6). However, the GA based algorithm performs quite well in comparison to the K-Means algorithm and produces well defined clusters, even in this case. The parameters used by the GA algorithm in both the cases is shown in Table 2. To speed up the running of the GA, all functions were vectorized in MATLAB allowing for parallel computations on multiple chromosomes.

Table 1 Synthetic dataset 1

Dataset attribute	Value
No. of samples	600
No. of features	3
Data type of features	Floating point
No. of clusters	4
Standard dev. of clusters	0.5

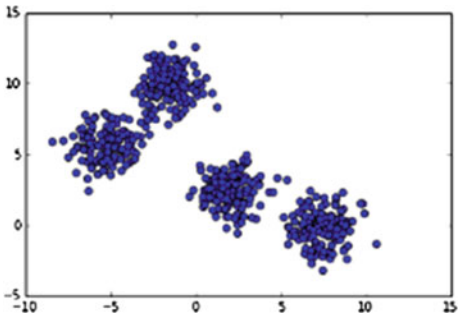

Fig. 3 Two dim. view of synthetic dataset 1

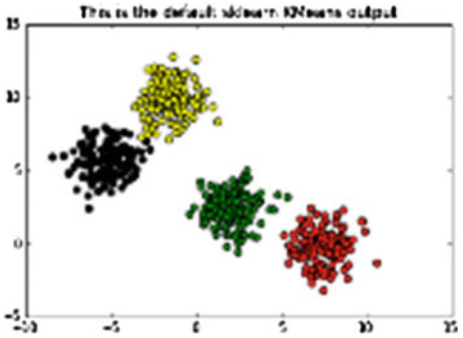

Fig. 4 Result of K-means clustering on synthetic dataset 1

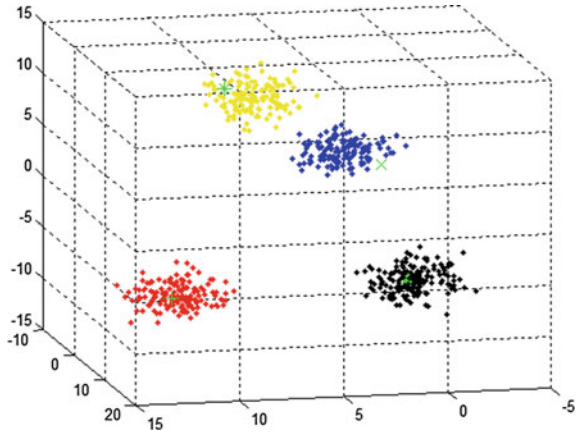

Fig. 5 Clustering of synthetic dataset using proposed algorithm

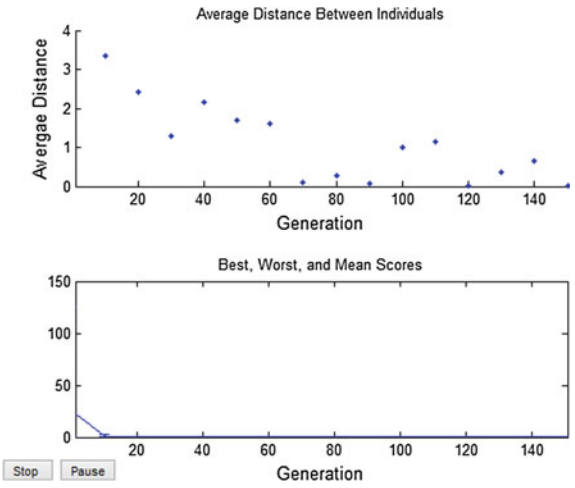

Fig. 6 Convergence of the proposed algorithm on synthetic dataset 1

Table 2 Parameters used by proposed GA

GA parameter	Value
Population size	10 * no. of features
Generations	250
Selection method	Tournament selection
Crossover	Two point crossover
Mutation	Adaptive mutation
Crossover function	0.8

4.2 Results on Text Data Sets

The proposed algorithm was used to classify the BBC Sports dataset which comprised of a subset of 389 documents taken from the original corpus. These documents belonged to two distinct sports topics i.e. cricket and football. As a part of preprocessing the data, the corpus was folded to lower case, stop words were removed and then the TF-IDF representation of the dataset was obtained. To further reduce the dimensionality of the data, we performed Singular Value Decomposition [7] of the TF-IDF representation and truncated the decomposition to contain only the first three dimensions. This allowed for a comprehensible visualization of the data, while retaining sufficient information about the document contents themselves. The representation of this truncated dataset is presented in Fig. 7. The truncated SVD vectors were further normalized to unit length to compensate for different document lengths. Cosine similarity has been used to measure the similarity between the vectors.

The results obtained by performing K-Means is seen in Fig. 8, and while the clusters seem to be quite compact, it is interesting to observe that documents near the borders of the two clusters seem to have been wrongly clustered. The results obtained on the same dataset using the proposed GA based algorithm is shown in Fig. 9, and is observed that while the most of the results replicate the results obtained using the K-Means algorithm, the proposed algorithm seems to have done a better job in classifying the border cases. This can be attributed to the fact that not only does the algorithm consider inter cluster distance and intra cluster scatter, it also takes into account the nearest neighbour separation. The convergence of the

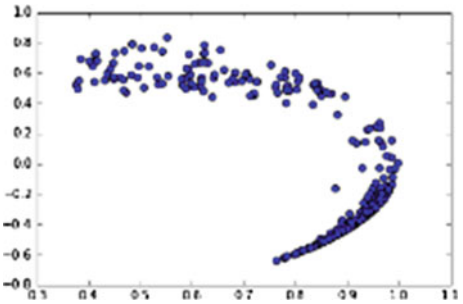

Fig. 7 Two dim. representation using first two SVD vectors

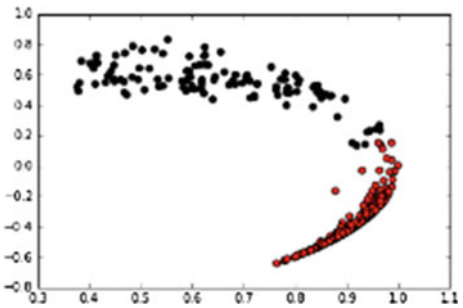

Fig. 8 Result of K-means clustering for two clusters

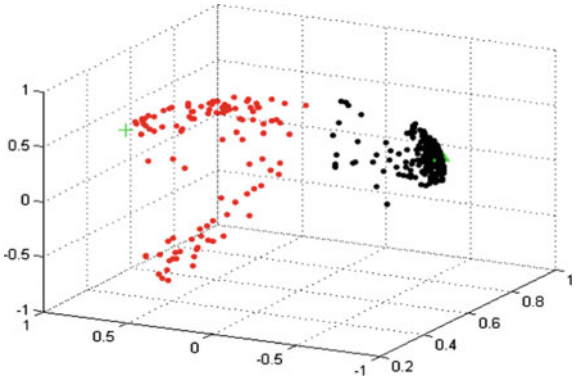

Fig. 9 Clustering using the proposed algorithm

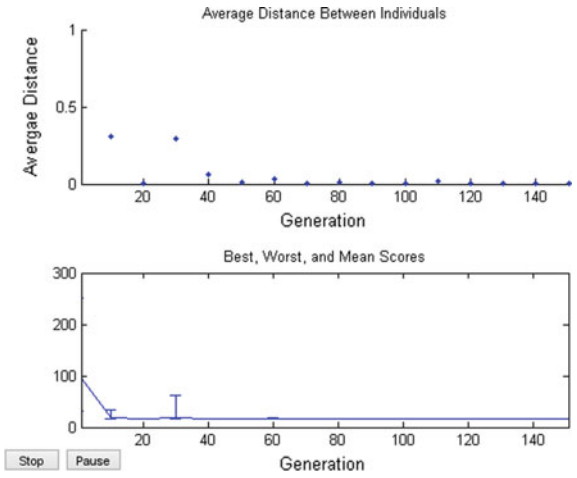

Fig. 10 Convergence of the proposed algorithm on the BBC sports dataset

algorithm is seen in Fig. 10 and it can be seen that again the algorithm converges in a small number of iterations.

To further test the algorithm the experiment was repeated on the BBC dataset, containing 3 different topics i.e. Sports, Politics and Technology. The corpus

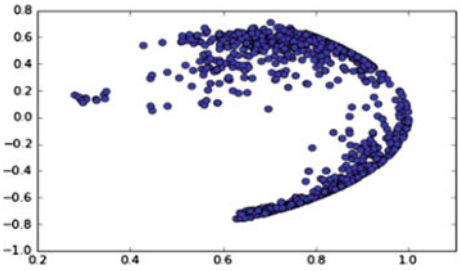

Fig. 11 Two dim. representation using first two SVD vectors

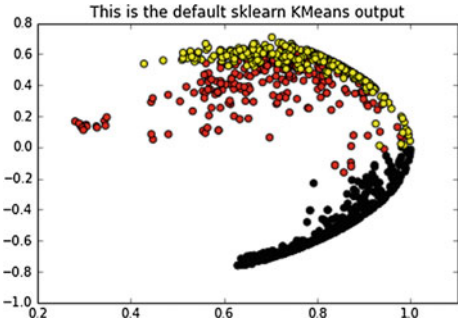

Fig. 12 Result of K-means clustering into three partition

contained 1012 documents, and the representation of the dataset is shown in Fig. 11.

The results obtained using K-Means with K = 3 is seen in Fig. 12, and the results obtained using the proposed algorithm is seen in Figs. 13 and 14. The results demonstrate the change in the clusters obtained while varying the values of α and β. While using large values of α tend to form more compact clusters, there is a possibility of overlap around the borders of the clusters. On the other hand, providing lager values of β, it is possible to pull the clusters away from each other. By choosing a suitable value of the tuple <α, β>, it is possible to fine tune the clusters as per requirement.

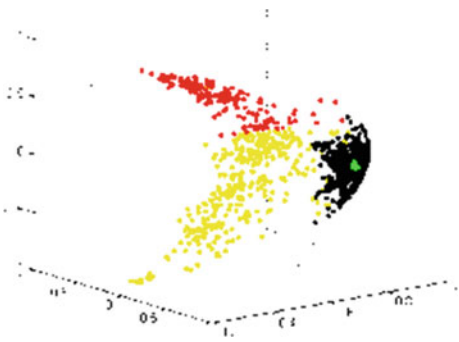

Fig. 13 Clustering results with proposed algorithm for large value of α

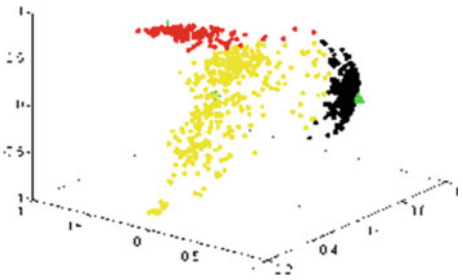

Fig. 14 Clustering results with proposed algorithm for large value of β

5 Conclusion

In this article, we propose a solution to the document clustering problem using genetic algorithms. In order to demonstrate the effectiveness of the GA based document clustering algorithm, several synthetic and real-life data sets with large number of dimensions and the number of clusters have been considered. This paper proposes a modified fitness function using the nearest neighbor criteria to improve the quality of the clusters formed. The speed of convergence of our proposed algorithm is comparable to the classical K-Means algorithm. The scalar parameter α and β can be adjusted by the user to control the quality of the cluster. Further work is in progress to improve the algorithm, particularly in the automatic setting of α and β.

References

1. Akter, R., Chung, Y.: An evolutionary approach for document clustering. IERI Procedia **4**, 370–375 (2013)
2. Kalogeratos, A., Likas, A.: Document clustering using synthetic cluster prototypes. Data Knowl. Eng. **70**(3), 284–306 (2011)
3. Matthews, S.G., Gongora, M.A., Hopgood, A.A., Ahmadi, S.: Web usage mining with evolutionaryextraction of temporal fuzzy association rules. Knowl.-Based Syst. **54**, 66–72 (2013)
4. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Coello Coello, C.: A survey of multiobjective evolutionary algorithms for data mining: part i. IEEE Trans. Evol. Comput. **18** (1), 4–19 (2014)
5. Nasir, J.A., Varlamis, I., Karim, A., Tsatsaronis, G.: Semantic smoothing for text clustering. Knowl.-Based Syst. **54**, 216–229 (2013)
6. Premalatha, K., Natarajan, A.M.: Genetic algorithm for document clustering with simultaneous and ranked mutation. Modern Appl. Sci. **3**(2), (2009)
7. Rana, C., Jain, S.K.: An evolutionary clustering algorithm based on temporal features for dynamic recommender systems. Swarm Evol. Comput. **14**, 21–30 (2014)
8. Singh, V.K., Tiwari, N., Garg, S.: Document clustering using k-means, heuristic k-means and fuzzy c-means. In: Computational Intelligence and Communication Networks (CICN), 2011 International Conference on, pp. 297–301. IEEE (2011)
9. Song, W., Qiao, Y., Park, S.C., Qian, X.: A hybrid evolutionary computation approach with its application for optimizing text document clustering. Expert Syst. Appl. **42**(5), 2517–2524 (2015)