An Analytical Approach to Document Clustering Based on Internal Criterion Function

Alok Ranjan  
Department of Information Technology  
ABV-IIITM  
Gwalior, India

Eatesh Kandpal  
Department of Information Technology  
ABV-IIITM  
Gwalior, India

Harish Verma  
Department of Information Technology  
ABV-IIITM  
Gwalior, India

Joydip Dhar  
Department of Applied Sciences  
ABV-IIITM  
Gwalior, India

Abstract—Fast and high quality document clustering is an important task in organizing information, search engine results obtaining from user query, enhancing web crawling and information retrieval. With the large amount of data available and with a goal of creating good quality clusters, a variety of algorithms have been developed having quality-complexity trade-offs. Among these, some algorithms seek to minimize the computational complexity using certain criterion functions which are defined for the whole set of clustering solution. In this paper, we are proposing a novel document clustering algorithm based on an internal criterion function. Most commonly used partitioning clustering algorithms (e.g., k-means) have some drawbacks as they suffer from local optimum solutions and creation of empty clusters as a clustering solution. The proposed algorithm usually does not suffer from these problems and converge to a global optimum, its performance enhances with the increase in number of clusters. We have checked our algorithm against three different datasets for four different values of k (required number of clusters).

Keywords—Document clustering; partitioning clustering algorithm; criterion function; global optimization

I. INTRODUCTION

Developing an efficient and accurate clustering algorithm has been one of the most favorite areas of research in various scientific fields. Various algorithms have been developed over a period of years [2, 3, 4, 5]. These algorithms can be broadly classified into agglomerative [6, 7, 8] or partitioning [9] approaches based on the methodology used or into hierarchical or non-hierarchical solutions based on the structure of solution obtained.

Hierarchical solutions are those which are in the form of a tree called dendograms [15], which can be obtained by using agglomerative algorithms, in which, first each object is assigned to its own cluster and then pair of clusters are repeatedly joined until a certain stopping condition is not satisfied. On the other hand, partitioning algorithms such as k-means [5], k-medoids [5], graph-partitioning-based [5] consider whole data as a single cluster and then find clustering solution by bisecting or partitioning it into number of predetermined classes. However, a repeated application of partitioning application can give a hierarchical clustering solution.

There always involves tradeoffs between a clustering solution quality and complexity of algorithm. Various researchers have shown that partitioning algorithms in terms of clustering quality are inferior in comparison to agglomerative algorithms [10]. However, for large document datasets they perform better because of small complexity involved [10, 11].

Partitioning algorithms work using a particular criterion function with the prime aim to optimize it, which determines the quality of clustering solution involved. In [12, 13] seven criterion functions are described categorized into internal, external and hybrid criterion functions. The Best way to optimize these criterion functions in partitioning algorithmic approach is to use greedy approach as in k-means. However the solution obtained may be sub-optimal because many a times these algorithms converge to a local-minima or maxima. Probability of getting good quality clusters depends on the initial clustering solution [1]. We have used an internal criterion function and proposed a novel algorithm for initial clustering based on partitioning clustering algorithm. In particular we have compared our approach with the approach described in [1] and implementation results show that our approach performs better then the above method.

II. Basics

In this paper documents have been represented using a vector-space model [14]. This model visualizes each document, d as a vector in the term-space or more in more precise way each document d is represented by a term-frequency (T-F) vector.

\[ d_{tf} = (t_{f1}, t_{f2}, \ldots, t_{fm}), \]

where \( t_{fi} \) denotes the frequency of the \( i_{th} \) term in the document. In particular we have used a term-inverse document frequency (tf-idf) term weighing model [14]. This model works better when some terms appearing more frequently in documents having little discrimination power need to be de-emphasized. Value of idf is given by \( \log (N \)
solution by optimizing a criterion function defined over Internal criterion functions account for finding clustering
A. solution in the context of internal criterion function.

Given a set \( A \) of documents, we define clustering as a technique to group similar documents together without the prior knowledge of group definition. Thus, we are interested in finding \( k \) smaller subsets \( S_i \) (\( i = 1, 2, \ldots, k \)) of \( A \) such that documents in same set are more similar to each other while documents in different sets are more dissimilar. Moreover, our aim is to find the clustering solution in the context of internal criterion function.

A. Internal Criterion Function

Internal criterion functions account for finding clustering solution by optimizing a criterion function defined over documents which are in same set only and doesn’t consider the effect of documents in different sets.

The criterion function we have chosen for our study attempts to maximize the similarity of a document within a cluster with its cluster centroid [11]. Mathematically it is expressed as

\[
\text{Maximize } T = \sum_{r=1}^{R} \sum_{d \in S} \cos (d_i, C_r)
\]

Where, \( d_i \) is the \( i_{th} \) document and \( C_r \) is the centroid of the \( r_{th} \) cluster.

IV. ALGORITHM DESCRIPTION

Our algorithm is basically a greedy one, unlike other partitioning algorithm (e.g. k-means) it generally does not converge to a local minimum.

Our algorithm consists of mainly two phases (i) initial clustering (ii) refinement.

A. Initial clustering

This phase consists of determining initial clustering solution which is further refined in refinement phase, with the assumption

In this phase of algorithm, our aim is to select \( K \) documents, hereafter called seeds, which will be used as initial centroid of \( K \) clusters required.

We select the document which has minimum sum of squared distances from the previously selected documents. In the process we get the document having largest minimum distance from previously selected documents, i.e., document which is not in the neighborhood of currently present documents.

Let at some time we have \( m \) documents in the selected list, we check the sum \( S = \sum_{i=1}^{k} (\text{Dist}(d_i, a))^2 \) for all documents a in set A, where set A contains the documents having largest sum of distances from previously selected m documents, and finally the document having minimum value of \( S \) is selected as the \( (m+1) \)th document. We continue this operation until we have \( K \) documents in the selected list.

1) Algorithm:

\[\text{Step1: } \text{DIST} \leftarrow \text{adjacency matrix of document vectors}\]

\[\text{Step2: } R \leftarrow \text{regulating parameter}\]

\[\text{Step3: } \text{LIST} \leftarrow \text{set of document vectors}\]

\[\text{Step4: } N \leftarrow \text{number of document vectors}\]
Step 5: \( K \leftarrow \) number of clusters required

Step 6: \( \text{ARR\_SEEDS} \leftarrow \) list of seeds initially empty

Step 7: Add a randomly selected document to \( \text{ARR\_SEEDS} \)

Step 8: Add to \( \text{ARR\_SEEDS} \) a new document farthest from the residing document in \( \text{ARR\_SEEDS} \)

Step 9: Repeat steps 10 to 13 while \( \text{ARR\_SEEDS} \) has less than \( K \) elements

Step 10: \( \text{STORE} \leftarrow \) set of pair (sum of distances of all current seeds from each document, document ID)

Step 11: Add in \( \text{STORE} \) the pair (sum of distances of all current seeds from each document, document ID)

Step 12: Repeat Step 13 \( R \) times

Step 13: Add to \( \text{ARR\_SEEDS} \) the document having least sum of squared distances from available seeds

Step 14: Repeat 15 and 16 for all remaining documents

Step 15: Select a document

Step 16: Assign selected document to the cluster corresponding to its nearest seed

Now we have \( K \) seeds in \( \text{ARR\_SEEDS} \) each representing a cluster. For the remaining \( N-K \) documents, each document is assigned to the cluster corresponding to its nearest seed.

B. Refinement

The refinement phase consists of many iterations. In each iteration all the documents are visited in random order, a document \( d_i \) is selected from a cluster and it is moved to other \( k-1 \) clusters so as to optimize the value of criterion function. If a move leads to an improvement in the criterion function value then \( d_i \) is moved to that cluster. As soon as all the documents are visited an iteration ends. If in an iteration there are no documents remaining, such that their movement leads to improvement in the criterion function, the refinement phase ends.

1) Algorithm:

Step 1: \( S \leftarrow \) Set of clusters obtained from initial clustering

Step 2: Repeat steps 3 to 9 until even a single document moved between clusters

Step 3: Unmark all documents

Step 4: Repeat steps 5 to 9 while each document is not marked

Step 5: Select a random document \( X \) from \( S \)

Step 6: If \( X \) is not marked, perform Steps 7 to 9

Step 7: Mark \( X \)

Step 8: Search cluster \( C \) in \( T \) in which \( X \) lies

Step 9: Move \( X \) to any cluster other than \( C \) by which the overall criterion function value of \( S \) goes down. If no such cluster exists don't move \( X \).

V. IMPLEMENTATION DETAILS

To test our algorithm we have coded it and the older one in Java Programming language. The rest of this section describes about the input dataset and cluster quality metric entropy which we have used in our paper.

A. Input Dataset

For testing purpose we have used both a synthetic dataset and a real dataset.

1) Synthetic Dataset

This dataset contains a total 15 classes from different books and articles related to different fields such as art, philosophy, religion, politics etc. The description is as follows.
| Class label   | Number of documents | Class label   | Number of documents |
|--------------|---------------------|--------------|---------------------|
| Architecture | 100                 | History      | 100                 |
| Art          | 100                 | Mathematics  | 100                 |
| Business     | 100                 | Medical      | 100                 |
| Crime        | 100                 | Politics     | 100                 |
| Economics    | 100                 | Sports       | 100                 |
| Engineering  | 100                 | Spiritualism | 100                 |
| Geography    | 100                 | Terrorism    | 100                 |
| Greek        | 100                 | Mythology    | 100                 |

In this paper we used entropy measure for determine the quality of clustering solution obtained. Entropy value for a particular k-way clustering is calculated by taking the average of entropies obtained from ten executions. Then these values are plotted against four different values of k, i.e., number of clusters. Experimental results are shown in the form of graphs [see Figure 1-3]. The first graph is obtained using the synthetic dataset having 15 classes. The second one is obtained using dataset re0 [16] and the third one is obtained using dataset re1 [16]. The results reveals that the entropy values obtained using our novel approach is always smaller, hence it is better then [1]. Also it is obvious from the graphs that the value of entropy decreases with the increase in the number of clusters as expected.

2) **Real Dataset**

It consists of two datasets namely re0 and re1 [16]

| Data | Source             | Number of documents | Number of classes |
|------|--------------------|---------------------|-------------------|
| re0  | Reuters-21578      | 1504                | 13                |
| re1  | Reuters-21578      | 1657                | 25                |

**B. Entropy**

Entropy measure uses the class label of a document assigned to a cluster for determining the cluster quality. Entropy gives us the information about the distribution of documents from various classes within each cluster. An ideal clustering solution is the one in which all the documents of a cluster belong to a single class. In this case the entropy will be zero. Thus, the smaller value of entropy denotes a better clustering solution.

Given a particular cluster $S_r$ of size $N_r$, the entropy [1] 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 number of classes available in the dataset, and $N_r^i$ is the number of documents belonging to the $i_{th}$ class that were assigned to the $r_{th}$ cluster. The total entropy will be given by the following equation

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

VI. **RESULTS**
VII. CONCLUSIONS

In this paper we have successfully proposed and tested a new algorithm that can be used for accurate document clustering. We know that the most of the previous algorithms have a relatively greater probability to trap in local optimal solution. Unlike them this algorithm has a very little chance to trap in local optimal solution, and hence it converges to a global optimal solution. In this algorithm, we have used a completely new analytical approach for initial clustering which refines result and it gets even more refined after the completion of refinement process. The performance of the algorithm enhances with the increase in the number of clusters.

REFERENCES

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

[2] Cui, X.; Potok, T.E.; Palathingal, P., "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, 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] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," Knowledge Discovery and Data Mining, 1999, pp. 16-22.

[11] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," KDD Workshop on Text Mining Technical report of University of Minnesota, 2000.

[12] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," Mach. Learn., vol. 55, no. 3, pp. 311-331, June 2004.

[13] Y. Zhao and G. Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," in CIKM ’02: Proceedings of the eleventh international conference on Information and knowledge management. ACM Press, 2002, pp. 515-524.

[14] G. Salton, "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer," Addison-Wesley, 1989.

[15] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," Data Mining and Knowledge Discovery, vol. 10, no. 2, pp. 141-168, March 2005.

[16] http://glaros.dtc.umn.edu/gkhome/fetch/sw/chuto/datasets.tar.gz