International Journal of Mathematical Archive-2(9), 2011, Page: 1572-1577
Available online through www.ijma.info ISSN 2229 – 5046

EVALUATING PERFORMANCE OF PARTITION BASED DOCUMENT CLUSTERING ALGORITHMS FOR INFORMATION RETRIEVAL

[1]P. Prabhu* and [2]Dr. R. Jeyshankar

[1]Assistant Professor in Information Technology, Directorate of Distance Education Alagappa University, Karaikudi, Tamilnadu, India

[2]Assistant Professor, Department of Library and Information Science, Alagappa University, Karaikudi, Tamilnadu, India

E-mail: [1]pprabhu70@gmail.com, [2]jeyshankar71@gmail.com

(Received on: 29-08-11; Accepted on: 10-09-11)

ABSTRACT

Information Retrieval (IR) is an emerging subfield of information science concerning representation, storage; access and retrieval of information. Current research areas within the field of IR include searching and querying, ranking of search results, navigating and browsing information, optimizing information representation and storage and document classification and clustering. Within information retrieval, clustering of documents has several promising applications, all concerned with improving efficiency and effectiveness of the retrieval process. This paper focus on performance evaluation of two Partition based Document clustering algorithms. Firstly, various steps for preprocessing the documents for clustering are discussed. Second Partition based algorithms like k-means and Spherical k-means a variant of the k-means algorithm that uses cosine similarity is discussed. Finally, the performance evaluation of the algorithm is investigated with different execution of the program on the various document collections as a post processing. The execution time for each algorithm is also analyzed and the results are compared with one another.

Keywords: *Information Retrieval, document clustering, document classification, k-means, Spherical k-means.*

I INTRODUCTION

The study of information retrieval is not new to computer science. The core technology has not changed significantly over the last twenty years. Most IR systems are based on inverted indices, which, for each keyword in the language, store a list of documents containing that keyword. The Vector Space model provides a different way of looking at the same information, but is not used as often in practice. A vector space implementation stores a lists of (keyword, frequency) pairs for each document in the data set. This allows a set of documents to be visualized as points in an *n* dimensional space, where *n* is the total number of keywords in the language. The applications of document clustering in information Retrieval include finding similar documents, search result clustering and faster and better searching. In this research, the performance of two partition based document clustering algorithms is analyzed.

II DOCUMENT CLUSTERING

Cluster analysis organises data by abstracting underlying structure either as a grouping of individuals or as a hierarchy of groups. There are various clustering algorithms are proposed in the literature like Partition, Hierarchical, Density based etc., Here Partition algorithms are presented for the study. Clustering algorithms can be applied in many fields include finding groups of customers with similar behavior, Gene Expression analysis, document classification and clustering, clustering weblog data to discover groups of similar access patterns etc.,

Document clustering can be defined as the automatic discovery of document clusters/groups in a document collection, where the formed clusters have a high degree of association (with regard to a given similarity measure) between members. The aim of a good document clustering scheme is to minimise intra-cluster distances between documents, while maximising inter-cluster distances. A distance measure thus lies at the heart of document clustering. Several ways for measuring the similarity between two documents exist, some are based on the vector model (e.g. Cosine distance or Euclidean distance) while others are based on the Boolean model (e.g. size of intersection between document term sets). Clustering documents from web/xml/other repositories involves many stages. Each stage has

---

**Corresponding author: [1]P. Prabhu*, *E-mail: pprabhu70@gmail.com*

multiple sub stages. The following steps are involved in document clustering.

1. Load the web/xml/other document repositories.
2. Preprocessing.
3. Clustering.
4. Post processing.

The following section discusses the above mentioned steps in detail.

IV PREPROCESSING

In document clustering, preprocessing include everything from the basic task of converting the indexes into a suitable data representation (e.g. a term-document matrix) to more advanced techniques such as various kinds of parsing the xml document, tokenization, stop word removal, stemming and term weighting.

(1) Parsing the XML document
All the markup tags are removed to parse the documents using a parser to take the information inside the body tag into a new file.

(2) Tokenization
The text corpus as seen in the screenshot above after parsing is cumbersome and has to be tokenized. Tokenization is the process of breaking parsed document text into chunks, called tokens. This process includes removing the punctuations and the text is lowercased.

(3) Stop words Removal
Next after tokenization, it is needed to remove stop words from the list of words. Stop words like is, are, with, the, from, to etc that occur in almost every document are be removed to proceed further which doesn't provide any use to for weighted index being so common.

(4) Stemming
Stemming refers to the process of reducing terms to their stems or root variants. For example Finding-> Find; Hardware -> Hard etc. Stemming reduces the computing time as different form of words is stemmed to form a single word.

(5) Building Inverted Index
Indexing is nothing but refinement i.e. a sufficient general description of a document such that it can be retrieved with a query that contains the same subject as the document and vice versa. Inverted index file contains an inverted file entry that stores a list of pointers to all occurrences of that term in the main test for every term in the lexicon, where each pointer is, in effect, the number of a document in which the term appears. There are two types of inverted index. A record level inverted index consists of a list of references to documents for each term. But for further processing we need significant terms that are obtained from dimensionality reduction. This is a major difficulty in text categorization of feature space i.e. total number of terms considered. We need to reduce the number of terms in the collection which is done by dimensionality reduction.

Once significant terms are obtained, the next step is to find the term frequency and document frequency in order to form vectors for processing clustering algorithm. The Figure 1 shows the Frequency of Terms/words obtained for com.sys.ibm.pc.hardware of 20-Newsgroup Dataset.

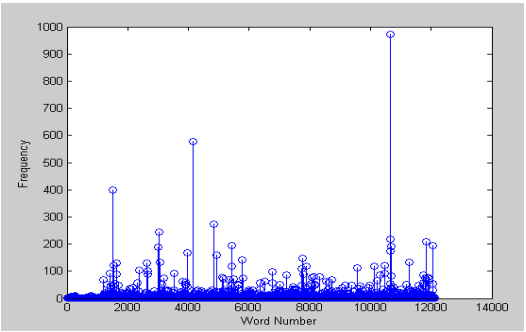


Figure 1: Frequency of Terms/words obtained for 20 Newsgroup com. sys. ibm. pc. hardware Dataset

(6) TF * IDF Calculation

Term Frequency and Inverse Document Frequency is a weight often used in text mining and information retrieval. It is a measure of how important a word is to a document in a collection. Term Frequency is defined as the total count of word that is repeated in a document. Inverse Document Frequency is defined as the total number of times the word occurs in the entire documents i.e. number of documents containing the significant word. Thus the term frequency is given by,

$$tf_{i,j} = n_{i,j} / \sum_k n_{k,j} \quad (1)$$

Where n_{ij} is the number of times the significant term ti occurs in document dj and the denominator is the sum number of times all the terms occur in document dj The inverse document frequency is obtained by dividing the number of documents by the number of documents containing the term, and then the logarithm of that quotient given by,

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : t_i \in d\}|} \quad (2)$$

Here, |D| is the total number of documents in the corpus, $|\{d : t_i \in d\}|$ is the number of documents where the term ti appears (that is ni,j is not equal to 0. If the term is not in the corpus, this will lead to a division by zero. Therefore it is common to use $1 + |\{d : t_i \in d\}|$, Then we define TF-IDF given by,

$$(\text{tf-idf})_{i,j} = tf_{i,j} \times idf_i \quad (3)$$

TF * IDF matrix, which representing a vector space model formed by documents where each row represents vector or document and columns show the dimensions of that vector. For example, the following shows the tf*idf matrix,

	Term1	Term2	Term3	….	Term m
Doc 1					
Doc 2					
Doc 3					
.					
Doc n					

After obtaining tf-idf matrix, the most commonly used dimensionality reduction technique especially Principal component Analysis (PCA) is applied to get the reduced dataset.

Dimensionality Reduction using PCA

PCA is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. The main advantage of PCA is that once you have found these patterns in the
data, and you compress the data, i.e. by reducing the number of dimensions, without much loss of information. Here PCA is applied to get the reduced dataset. This reduced data set is given as input to clustering algorithm. The following figure 2 shows patterns of the reduced dataset of 20-Newsgroup com.sys.ibm.pc.hardware.

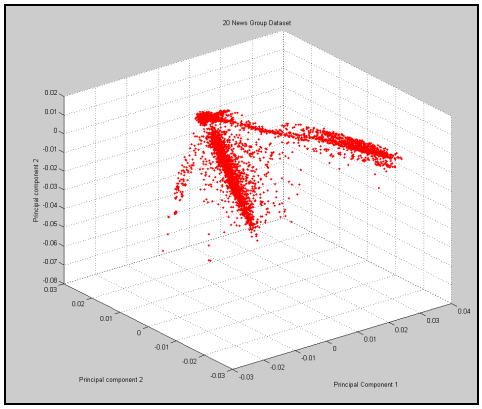


Figure 2: 20- Newsgroup com.sys. ibm. pc. hardware Dataset

V. PARTITION CLUSTERING ALGORITHMS

Partition clustering algorithm splits the data points into k partition, where each partition represents a cluster. The partition is done based on certain objective function. The k-means and Spherical k-means clustering algorithms are discussed here.

A.K-means Clustering Algorithm

The K-means algorithm, one of the most widely used clustering techniques. K-means clustering is an algorithm to classify or to group your objects based on attributes/features into K number of group. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. The K-means method uses the Euclidean distance measure, which appears to work well with compact clusters. The steps of the K-means algorithm are described in brief as follows:

Step 1. Select the number of clusters as k.
Step 2. Select k seeds as centroids of the k clusters. The seeds may be selected randomly.
Step 3. Compute the Euclidean distance of each object in the dataset from each of the centroids.
Step 4. Allocate each object the cluster it is nearest to based on the distances computed in the step 3.
Step 5. Compute the centroids of the clusters by computing the means of the attribute values of the objects in each cluster.
Step 6. Check if the stopping criterion has been met.If yes go to step 7, else go to Step 3
Step 7. One may decide to stop at this stage heuristically until stopping criterion is met.

Though k-means is simple to implement and provide results, the clusters formed failed for new distance metrics. It fails when the documents size is too large and takes lot of time to run for few metrics.

B. Spherical k-means algorithm

The spherical k-means algorithm, i.e., the k-means algorithm with cosine similarity, is a popular method for clustering high-dimensional text data [13]. In this algorithm, each document as well as each cluster mean is represented as a high-dimensional unit-length vector. However, it has been mainly used in batch mode. That is, each cluster mean vector is updated, only after all document vectors being assigned, as the (normalized) average of all the document vectors assigned to that cluster. This algorithm partitions the high dimensional unit sphere using a collection of great hyper circles, and hence we shall refer to this algorithm as the spherical k-means algorithm [4]. The algorithm computes a disjoint partitioning of the document vectors, and, for each partition, computes a centroid normalized to have unit Euclidean norm. This algorithm has a number of advantages from a computational perspective: it can exploit the sparsity of the text data, and converges quickly (to a local maxima).

Dataset Used

To perform clustering, the document collection used for research is obtained from “reuters-21578, Distribution 1.0 Test collection”. There are 21578 newswire stories classified into several sets of categories by personnel from Reuters Ltd. and Carnegie Group, Inc in 1987 and formatted in 1991.There are total 674 categories in Reuters-21578 collection. In the thesis, we concentrate on Reuters_000 document dataset taken from Reuters-21578 Collection. Another data set used is the publically available 20 Newsgroups data set. It is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. Here the comp.sys.ibm.pc.hardware data set is used for clustering. The original data was preprocessed to strip the news messages from the email headers and special tags and eliminate the stop words and stem words to their root forms. Then the words were sorted on the inverse document frequency (IDF), and some words were removed if the idf values were too small or too large. The K-means and Spherical k-means clustering algorithms were applied to group the documents.

RESULT AND DISCUSSION

As a post processing the clustered documents are evaluated with different number of clusters and dataset for performance benchmark.The Table 1 shows the result of Reuters000 dataset of Reuters 21578 Collection data set is clustered using two partition algorithms for various number of clusters. The execution time and No of Iterations are tabulated.

Reuters 21578 Collection Data set			
Number of clusters	K-Means		Sk-means Number of Iterations = 2
	No. of Iterations	Time (sec)	Time (Sec)
5	2	0.023734	0.024587
6	3	0.028651	0.024198
7	3	0.032455	0.023785
8	5	0.044443	0.024917
9	3	0.046097	0.025472
10	2	0.023889	0.025062

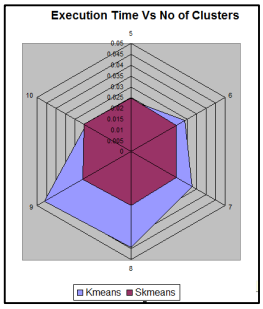
Table1: Result of Reuters000 of Reuters 21578 Dataset

The Table 3 shows the result of comp.sys.ibm.pc.hardware dataset of 20 Newsgroup Collection is clustered using two algorithms for various numbers of clusters. The execution time and Number of Iterations are tabulated.

20 News Group com.sys.ibm.pc.hardware Dataset				
No of Documents :6298 Terms : 12105				
Number of clusters	K-Means		Sk-means	
	No. of Iterations	Time (sec)	No. of Iterations	Time (Sec)
50	13	0.980118	11	0.634009
100	12	1.572822	12	1.229864
200	12	2.774488	8	1.451309
300	13	4.251139	11	2.616421
500	10	5.348480	8	2.926688

Table: 2 Result of 20 News Group Collections

The Figure 3 and 4 shows the comparison between clustering of document using k-means and sk-means Algorithms.



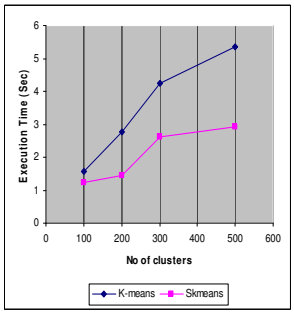


Fig.3 Reuters 000 Dataset

Fig 4. 20 News Group Dataset

The result of the clustering algorithm with various run shows that the spherical k-means algorithm is outperformed with less execution time than the k-means clustering. The following is the group of seven documents identified as a cluster using the clustering algorithm.

Document 320: thanx
Document 496: thanx in advance
Document 2699: thanx
Document 4290: thanx in advance for any info
Document 5185: thanx
Document 5879: thanx in advance
Document 6001: thanx

CONCLUSION AND FUTURE WORK

In information retrieval, the process of manually categorising the pages of an electronic / website document is often tedious and expensive. Document clustering has thus often been used to automatically categorise a search result into clusters. In this paper, two partition based k-means and Spherical k-means Clustering algorithm is used over the document collection taken from Reuters-21578 and 20 newsgroups dataset and performance is tested. As a result of various runs of clustering, spherical k-means clustering algorithm performs well than k-means for document clustering. Other clustering algorithms with different dataset may be applied for performance benchmark as a future work.

REFERENCES

[1] A. El-Hamdouchi and P. Willet, Comparison of Hierarchic Agglomerative Clustering Methods for Document Retrieval, The Computer Journal, Vol. 32, No. 3, 1989.

[2] Cai. D, He. X, and Han.J,"Document Clustering Using Locality Preserving Indexing," IEEE Trans. Knowledge and Data Eng.,Vol.17,no.12, Dec.2005.

[3] Gerald Kowalski, Information Retrieval Systems – Theory and Implementation, Kluwer Academic publishers, 1997.

[4] I. S. Dhillon and D. M. Modha, "Concept Decompositions for Large Sparse Text Data using Clustering", Machine % Learning, 42:1, pages 143-175, Jan, 2001.

[5] Jiawei Han, Micheline Kamber, Data Mining concepts and Techniques, Morgan Kaufmann Publishers, San Fracisco, CA, USA.

[6] J.Hyma, Y.Jhansi and S.Anuradha, A new hybridized approach of PSO & GA for document clustering, International Journal of Engineering Science and Technology, Vol.2 (5), 2010, 1221-1226.

[7] Margaret H.Dunham, Data Mining Introductory and Advanced Topics, Pearson Education in SouthAsia.

[8] Mehmed Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, IEEE Press & John Wiley, November 2002.

[9] Michael J.A.Berry Gordon Linoff, Mastering Data Mining, John Wiley & son's ptd Ltd, Singapore 2001.

[10] Manu Konchady, Text Mining Application Programming, Cengage Learning India Pvt Ltd. Fourth Indian Reprint 2009

[11] P.Prabhu and N.Anbazhagan, "Improving the performance of k-means clustering for high dimensional dataset', International Journal of Computer Science and Engineering", Vol 3. No.6. June 2011 Pg 2317-2322.

[12] P.Prabhu,' Method for Determining Optimum Number of Clusters for Clustering Gene Expression Cancer Dataset', International Journal of Advance Research in Computer Science (Volume 2 No. 4,July-August 2011) pg 315-318.

[13] Zhong, S. Efficient online spherical k-means clustering Proc. IEEE Int. Joint Conf. Neural Networks. July 31 - August 4, 2005. Montreal, Canada.] Proceedings 2005 IEEE International Joint Conference on Neural Networks 2005 (2005) Volume: 5, Publisher: IEEE, Pages: 3180-3185

AUTHORS PROFILE

P. Prabhu is working as Assistant Professor in Information Technology, Directorate of Distance Education, Alagappa University, Karaikudi, Tamilnadu, India. He received Master's Degree in Computer Applications from Bharathiar University, Coimbatore in 1993, and M.Phil degree in Computer Science from Bharthidasan University, Trichirappalli in 2005. He has published many articles in National and International Journals. He has presented papers in National and International Conferences. His research area is Data Mining, Information Retrieval and Computer Networks.

Dr. R. Jeyshankar M. A, M. L. I. Sc, M. Phil, P.G.D.C.A Ph. D is working as Asst. Professor, in the Department of Library and Information Science, Alagappa University, Karaikudi, Tamil Nadu and passed twice UGC – NET. He has published about a dozen articles in the refereed journals from India. He also presented and attended above two dozen national and international conferences, seminars, workshops and symposia. His areas of specialization include webometrics, Scientometrics, User Studies, Information Retrieval and Digital Libraries.

****

