ISSN: 2321-7782 (Online)

Volume 3, Issue 9, September 2015

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Similarity Measure Based Hierarchical Clustering Of Documents

M. Latha[1]
Ph.D., Research Scholar,
P.G. and Research Department of Computer Science,
J.J. College of Arts And Science, Pudukkottai,
Affiliated to Bharathidasan University, Trichy, India

Dr. K. Subramanian[2]
Vice Principal & Research Co-Ordinator,
P.G. and Research Department of Computer Science,
J.J. College of Arts And Science, Pudukkottai,
Affiliated to Bharathidasan University, Trichy, India

Abstract: Clustering is a method of grouping similar objects. Representation the data by smaller quantity clusters evade obvious fine details present in the dataset, but accomplish simplification in data processing and document processing. Akin documents are grouped together to form a cluster, if their cosine similarity measure is less than a specified threshold value. These papers mainly focuses on document clustering, similarity measures in hierarchical clustering and develop a novel clustering algorithm named Advanced Similarity Measure Clustering Algorithm. The new algorithm is compared with many existing algorithm to find the clustering efficiency and computational time. The hierarchical document clustering algorithm bestows an authentic technique of distinguishing clusters of similar objects and implements the vital requisites of clustering based on similarity and dissimilarity.

Keywords: Document Clustering, Text mining, Similarity measure, Hierarchical Method.

I. INTRODUCTION

For fast information retrieval, filtering of data, speedy extraction of meaningful data and to organize documents, Document clustering is the best choice. This technique is intimately allied to data clustering. In general Document clustering techniques commonly depends on single term analysis of the document data set, such as the Vector Space Model. To attain more accurate document clustering, many informative features including phrases and their weights plays an important role. Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering has a wide range of applications such as automatic categorization of documents, grouping and clustering search engine results, building of documents etc. For these applications Hierarchical Clustering method provides a better platform in achieving better result. This paper focuses on two vital parts of successful Hierarchical document clustering.

The first part is a document index model which helps index construction of the document set with a weight based on efficiency and accuracy, instead relying on single-term indexes alone. It offers effective and well-organized phrase matching that is used to gauge the similarity between documents. This model is flexible, efficient and accurate in identifying the similarities present in the document sets.

The second part mainly focuses on maximizing the precision of clusters by adeptly analyzing the pair wise document similarity distribution present inside clusters. Most of the existing technique readily available across the globe selects the next frequent item set which represent the next cluster to minimize the overlapping between the documents that contain both the item set. As a result the clustering formed depends heavily on the order of picking up the item sets. This method does not follow a sequential order of selecting clusters. In the proposed approach, the main work is to develop a novel hierarchal algorithm for document clustering which enhances efficiency & performance to the maximum and to make use of the cluster overlapping occurrence to form meaningful clusters. The proposed technique concentrates on a novel way to compute the overlap rate and improves efficiency and reduces execution time.

Hierarchical techniques produce a nested sequence of partitions, with a single cluster at the top and singleton clusters of individual points at the bottom. Each intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level). The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendogram.

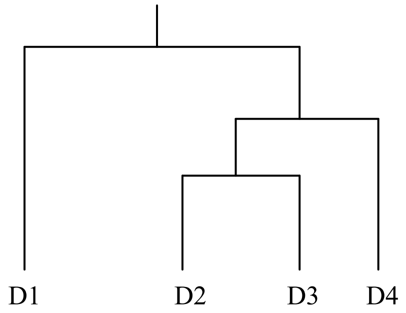

Diagram 1: Hierarchial Clustering Tree

II. Experimental Work

A. HIGH DIMENSIONALITY

Each unique word in the document set constitutes a dimension. So there may be 15 to 20 thousands dimensions. This type of high dimensionality largely influences the scalability and efficiency of many existing clustering algorithms available across.

B. HIGH VOLUME OF DATA

In text mining, processing of data about 10 thousands to 100 thousands documents are involved.

C. CONSISTENTLY HIGH ACCURACY

Some existing algorithms work fine for a particular type of document sets, but may not perform equally well in some others type. Generalization of algorithm to work for all type of document sets is a major hindrance.

III. hierarchical Analysis Model

A hierarchical clustering algorithm creates a hierarchical breakdown of the given set of data objects. Depending on the breakdown approach, hierarchical algorithms are classified as agglomerative (merging) or divisive (splitting)

A. AGGLOMERATIVE MODEL

Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.

B. DIVISIVE MODEL

Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

C. AGGLOMERATIVE HIERARCHICAL CLUSTERING PROCEDURE AS FOLLOWS:

1. Compute and calculate the similarity between all pairs of clusters in the data set, i.e., calculate a similarity matrix whose ij^{th} *entry* gives the similarity between the i^{th} and j^{th} clusters.
2. Merge the most similar (closest) two clusters in the data set to form basic cluster.
3. Update the similarity matrix to reflect the pair wise similarity between the new cluster and the original cluster.
4. Repeat Step 2 and 3 until the cluster merges into a single one.

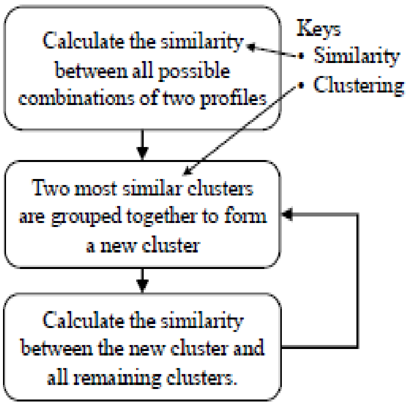
Calculate the similarity between all possible combinations of two profiles
Keys
• Similarity
• Clustering
Two most similar clusters are grouped together to form a new cluster
Calculate the similarity between the new cluster and all remaining clusters.

Diagram 2: Hierarchical Clustering

D. THE FOLLOWING ARE THE STEPS:

Step1 – Begin the process by assigning one item in the data set to a cluster, and if there are N items in the data set, the number of clusters will be N , each cluster contains one item. Let the distances (similarities) between the clusters be the same as the distances (similarities) between the items they contain.

Step 2 – Now find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less with the help of tf - idf.(Term frequency – Inverse document Frequency is a numerical value that is calculated to replicate how important a word to a document)

Step 3 - Compute distances (similarities) between the new cluster and each of the old clusters.

Step 4 - Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

E. DIVISIVE HIERARCHICAL CLUSTERING

This method does the reverse by starting with all objects/items in one cluster and subdividing them into smaller pieces. Divisive methods are not generally available, and rarely have been applied. Of course there is no point in having all the N items grouped in a single cluster but, once the complete hierarchical tree is obtained and need k clusters, k-1 longest links are eliminated.

F. INTRA-CLUSTER SIMILARITY TECHNIQUE (IST)

This hierarchical technique looks at the similarity of all the documents in a cluster to their cluster centroid and is defined by where *d* is a document in cluster, *X*, and ***c*** is the centroid of cluster *X*. The choice of which pair of clusters to merge is made by determining which pair of clusters will lead to smallest decrease in similarity. Thus, if cluster *Z* is formed by merging clusters *X* and *Y*, then we select *X* and *Y* so as to maximize Sim(*Z*) – (Sim(*X*) + Sim(*Y*)). Note that Sim(*Z*) – (Sim(*X*)+Sim(*Y*)) is non-positive.

$$sim(x) = {}_{d \in x} cosine(d, c)$$

IV. PROPOSED WORK

The TF-IDF is a text statistical-based technique which has been widely used in many search engines and information retrieval systems. Assume that there is a set of 1000 documents and the task is to compute the similarity between two given documents (or a document and a query). The following describes the steps of acquiring the similarity.

DOCUMENT PRE-PROCESSING STEPS

A. TOKENIZATION

A document is treated as a string or set of words, and then partitioned into a list of tokens. This process breaks the stream of texts into words, phrases and symbols.

B. REMOVING STOP WORDS

Stop words are frequently occurring, insignificant words. This step filters out the common words used in most of the documents to facilitate phrase search.

C. STEMMING WORD

This step is the process of reducing tokens to their root form.

D. DOCUMENT REPRESENTATION

Generating N-distinct words from the document and call them as index terms (or the vocabulary). The document collection is then represented as N-dimensional vector in term space.

E. TFIDF ANALYSIS

By considering the factors term frequency (TF) and inverse document frequency (IDF) it is possible to assign weights to search results and therefore ordering them statistically and providing them a value of importance. Put another way a search result's score Ranking is the product of TF and IDF:

TFIDF = TF * IDF where

TF = C / T

where

C = number of times a given word appears in a document ,

T = total number of words in a document.

Document IDF = D / DF

Where

D = total number of documents in a corpus,

DF = total number of documents containing a given word.

F. ALGORITHM ASMC

```
L ← Empty Cluster List
For each document D do
For each Cluster C in L do
Calculate similarity measure S
If S value finds a close Pair Then
        Form Cluster $C_{new}$ ( Merge Cluster)
Update similarity matrix
        While ( New Cluster = Old Cluster )
                Merge Cluster $C_{result}$ to singleton
        End While
End If
End For
End For
Return Cluster $C_{result}$
```

The proposed algorithm ASMC is applied to a dataset D1 comprising of 300 documents and checked for the performance and compared with the existing methodologies K-nearest Neighbor and single pass clustering. From the analysis the proposed algorithm performed exceedingly well with respect to performance related to accuracy, efficiency and time computation. From the graph shown in fig 3, the computational time of the ASMC relatively low when compared to single pass and k-NN algorithms.

The entropy value is calculated for the three algorithms using standard formula

$$E_j = -\sum_i p_{ij} \log(p_{ij})$$

Where p_{ij} is the probability that a member of cluster j belongs to class i.

$$E_C = \sum_{j=1}^{m} \left(\frac{N_j}{N} \times E_j\right)$$

Where N_j is the size of the cluster, N is total number of objects

Method	Entropy
k-NN	0.756
Single pass	0.619
ASMC	0.114

Table: 1

G. AUTOMATIC CLASSIFICATION

TDIDF can also be applied to indexing/searching to create browse lists hence, automatic classification is possible. Consider the table where each word is listed in a sorted TFIDF order: Given such a list it would be possible to take the first three terms from each document and call them the most significant subject “tags”. Thus, Document #1 is about Bank, Loan, and computers. Document #2 is about Insurance, accidents, and cars. Document #3 is about Education, textbooks, and authors. Probably a better way to assign “aboutness” to each document is to first denote TFIDF lower bounds and then assign terms with greater than that score to each document. Assuming lower bounds of 0.2, Document #1 is about Bank and loans, Document #2 is about Insurance, accidents, and cars, and Document #3 is about Education, textbooks and authors.

DATASET 1		
Document 1	Document 2	Document 3
WORD	WORD	WORD
Bank	Insurance	Education
Cheque	Fire	Degree
Loan	Money	Textbooks
Draft	Accident	Study
Money	Casuality	Loan
Accounts	Life	Subject
Branch	Coverage	Mark
Authorized	Cars	Author
Single	Bike	Position

Table:2

H. CUMULATIVE DOCUMENT

The cumulative document is the sum of all the documents, containing meta-tags from all the documents. We find the references in the input base document and read other documents and then find references in them and so on. Thus in all the documents their meta-tags are identified, starting from the base document.

V. CONCLUSION

Proper clustering algorithm should be applied for the given data set since, choosing a clustering algorithm, however, can be a cumbersome task. Most of the algorithms generally assume some implicit structure and process in the data set. Another issue to keep in mind is the kind of input and tools that the algorithm requires. The input documents processed using the clustering technique based on similarity (distance) measure and the computational time as well as the entropy value is found. This paper has a proposed of a novel hierarchical clustering algorithm based on the overlap rate for cluster merging. The proposed method indicates that it can decrease the time consumption cost, reduce the space complexity and improve the accuracy and precision of clustering. The hierarchical document clustering algorithm provides a natural way of distinguishing clusters and implementing the basic requirement of clustering as high within-cluster similarity and between-cluster dissimilarity.

References

1. Bruce Moxon "Defining Data Mining, The Hows and Whys of Data Mining, and How It Differs From Other Analytical Techniques" Online Addition of DBMS Data Warehouse Supplement, August 1996.
2. Willet, Peter "Parallel Database Processing, Text Retrieval and Cluster Analyses" Pitman Publishing, London, 1990.
3. M. Charikar, C. Chekuri, T. Feder and R. Motwani, Incremental Clustering and DynamicInformation Retrieval, Proceeding of the ACM Symposium on Theory of Computing, (1997), 626-634.
4. R. Gil-García and A. Pons-Porrata, Dynamic hierarchical algorithms for document clustering, Pattern Recognition Letters, 31 (2010), 469-477.
5. S. Guha, R. Rastogi and K. Shim, CURE: An efficient clustering algorithm for large databases, Information Systems, 26 (2001), 35-58.
6. K. Koutroumbas and S. Theodoridis, Pattern Recognition, Academic Press, (2009).
7. M. Kantardzic ,Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons,(2003).
8. D.T. Pham and A.A. Afify, Engineering applications of clustering techniques, Intelligent Production Machines and Systems, (2006), 326-331.
9. L. Feng, M-H Qiu, Y-X. Wang, Q-L. Xiang, Y-F. Yang and K. Liu, A fast divisive clustering algorithm using an improved discrete particle swarm optimizer, Pattern Recognition Letters, 31 (2010).
10. A.K. Jain, M.N. Murty and P.J. Flynn, Data clustering: A review, ACM Computing Surveys, 31(1999), 264-323.
11. N. A. Yousri, M. S. Kamel and M. A. Ismail, A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities, Pattern Recognition, 42 (2009), 1193-1209.
12. Athman Bouguettaya "On Line Clustering", IEEE Transaction on Knowledge and Data Engineering Volume 8, No. 2, April 1996.
13. Euripides G.M. Petrakis and Christos Faloutsos "Similarity Searching in Medical Image Databases", IEEE Transaction on Knowledge and Data Engineering Volume 9, No. 3, MAY/JUNE 1997.