Volume 3, Issue 7, July 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

# Survey on Hierarchical Document Clustering Techniques Fihc & $F^2$ Ihc

**Ms. Devika Deshmukh**[1] **, Mr. Sandip Kamble** [2]
*Computer Technology,*
*RGCER RTM Nagpur University, India*

**Mrs. Pranali Dandekar**[3]
*Information Technology*
*PBCE, RTM Nagpur University, India*

***Abstract*— Document clustering is an effective tool to manage information overload. By grouping similar documents together, we enable a human observer to quickly browse large document collections[18], make it possible to easily grasp the distinct topics and subtopics (concept hierarchies) in them, allow search engines to efficiently query large document collections [16] among many other applications. Hence, it has been widely studied as a part of the broad literature of data clustering. One such survey of existing clustering literature can be found in Jain et. al[19s].**

***Keywords*— Incremental Clustering, Hierarchical Clustering, Datasets, Fuzzy Frequent Item set, Literature review, datasets**

## I. Introduction

Clustering is a process of partitioning a set of data (or objects) into a set of meaningful subclasses called clusters. Hierarchical document clustering organizes clusters into a tree or a hierarchy that facilitates browsing. Document clustering algorithms are important in organizing documents generated from streaming on-line sources, such as, Newswire and Blogs. However, this is a relatively unexplored area in the text document clustering literature. In order to browse and organize documents smoothly, hierarchical clustering techniques have been proposed to cluster a collection of documents into a hierarchical tree structure. Despite that, there still exist several challenges for hierarchical document clustering, such as high dimensionality, scalability, and accuracy. The often studied document clustering algorithms are batch clustering algorithms, which require all the documents to be present at the start of the exercise and cluster the document collection by making multiple iterations over them. But, with the advent of online publishing in the World Wide Web, the number of documents being generated everyday has increased considerably. Popular sources of informational text documents such as Newswire and Blogs are continuous in nature. To organize such documents naively using existing batch clustering algorithms one might attempt to perform clustering on the documents collected so far. But, this is extremely time consuming, if not impossible, due to the sheer volume of documents. One might be tempted to convert the existing batch clustering algorithms into incremental clustering algorithms by performing batch clustering on periodically collected small batches of documents and then merge the generated clusters. However, ignoring for the moment the problem of deciding on an appropriate time window to collect documents, there will always be a wait time before a newly generated document can appear in the cluster hierarchy. This delay would be unacceptable in several important scenarios, e.g., financial services, where trading decisions depend on breaking news, and quick access to appropriately classified news documents is important. A clustering algorithm in such a setting needs to process the documents as soon as they arrive. This calls for the use of an incremental clustering algorithm. Hierarchical clustering algorithms can differ in their operation. Agglomerative clustering methods start with each object in a distinct cluster and successively merge them to larger clusters until a stopping criterion is satisfied. Alternatively, divisive algorithms begin with all objects in a single cluster and perform splitting until a stopping criterion is met. Both agglomerative and divisive hierarchical algorithms are static in the sense they never undo what was done previously, which means that objects which are committed to a cluster in the early stages, cannot move to another cluster. In other words, once a cluster is split or two clusters are merged, the split objects will never come together in one cluster or the merged objects will be never in the same cluster, no matter whether the splitting or the merging is the correct action or not. But in practice, some splitting or merging actions may not be correct and there is a need to rearrange the partition. This problem is a cause for inaccuracy in clustering, especially for poorly separated data sets.

## II. Related Literature Survey

Hierarchical and Partitioning methods are two major categories of clustering Algorithm. A hierarchical clustering method works by grouping data objects into a tree of clusters. Hierarchical clustering methods can be further classified as either agglomerative of divisive depending on whether the hierarchical decomposition is formed in a bottom up (merging) or top down (splitting) fashion. The quality of pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split has done. If a particular merge or split decision is a poor choice the method cannot backtrack and correct it. For improving the quality of cluster integration of hierarchical agglomeration with iterative relocation method is emphasized.

Given a database of n objects, a partitioning method construct k partitions of the data where each partition represents a cluster and k < = n. Thus it classifies the data into k groups where each group contain at least one object and each object must belong to exactly one group. Lifeng Wang, Hui Song, Xiaoqiang Liu in 2010 proposed a novel Multi-Representation Indexing Tree (MRIT) algorithm for constructing a hierarchy that satisfies arbitrary shape clusters with a good performance. Compared with the Indexing tree algorithm, a cluster is decomposed into several sub clusters and is represented as a union of the sub clusters rather than the centre of the cluster. Similarity of a document to one cluster is the distance to the nearest neighbour among the cluster's representative points. The experimental results on a variety of domains demonstrate that our algorithm can produce a quality cluster. It's insensitive to document input order, and efficient in terms of computational time.

Nachiketa Sahoo in 2006 proposed methods to carry out incremental hierarchical clustering of text documents. They proposed A Cobweb-based algorithm for text document clustering where word occurrence attributes follow Katz's distribution. They evaluate the performance of said algorithm and existing algorithms on large real world document datasets. Khaled M. Hammouda & Mohamed S. Kamel proposed an Incremental Document Clustering Using Cluster Similarity Histograms. An incremental document clustering algorithm is introduced, which relies only on pair-wise document similarity information. Clusters are represented using a Cluster Similarity Histogram, a concise statistical representation of the distribution of similarities within each cluster, which provides a measure of cohesiveness. The measure guides the incremental clustering process. Complexity analysis and experimental results are discussed and show that the algorithm requires less computational time than standard methods while achieving a comparable or better clustering quality. Dwi H. Widyantoro & Thomas R. Ioerger & John Yen proposed an Incremental approach aims to construct a hierarchy that satisfies the homogeneity and the monotonicity properties. Working in a bottom-up fashion, a new instance is placed in the hierarchy and a sequence of hierarchy restructuring process is performed only in regions that have been affected by the presence of the new instance. The experimental results on a variety of domains demonstrate that the algorithm is not sensitive to input ordering, can produce a quality cluster hierarchy, and is efficient in terms of its computational time.

Xiaoke Su, Yang Lan, Renxia Wan, and Yuming Qin in 2009 proposed A Fast Incremental Clustering Algorithm. The algorithm restricts the number of the final clusters and reads the original dataset only once. At the same time an inter-cluster dissimilarity measure taking into account the frequency information of the attribute values is introduced. It can be used for the categorical data. The experimental results on the mushroom dataset show that the proposed algorithm is feasible and effective. It can be used for the large-scale data set. Arnaud Ribert, Abdel Ennaji, Yves Lecourtier proposed An Incremental Hierarchical Clustering algorithm which updates the hierarchical representation of the data instead of re-computing the whole tree when new patterns have to be taken into account. Memory gains, evaluated for a real problem (handwritten digit recognition) allow to treat databases containing 7 times more data than the classical algorithm.

## III. Incremental Hierarchical Clustering

- Although hierarchical clustering techniques enable one to automatically determine the number of clusters in a data set, they are rarely used in industrial applications, because a large amount of memory is required when treating more than 10,000 elements.
- To solve this problem, the proposed method proceeds by updating the hierarchical representation of the data instead of re-computing the whole tree when new patterns have to be taken into account.

## IV. Frequent Itemset Based Hierarchical Clustering (Fihc)

It is a cluster centered because it measures the cohesiveness of a cluster directly using frequent item sets. Documents in the same cluster are expected to share more common item sets than those in different clusters. FIHC assigns documents to the best cluster from among all available clusters (frequent item sets). FIHC uses frequent item sets to construct clusters and to organize clusters into topic hierarchy

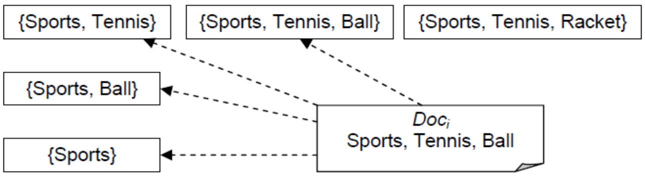
*Steps For FIHC Algorithm*

1) Construct Initial Clusters
2) Build Cluster Tree
3) Prune the Cluster Tree

### A. *Constructin Cluster*

- FIHC utilizes global frequent itemset as the cluster label to identify the cluster.
- For each document best initial cluster is identified and the document is assigned to the best matching initial cluster.
- The goodness of cluster Ci for a Document Docj is measured by some score function using cluster frequent items of initial clusters. Thus each document belong to exactly one cluster

*Example:*

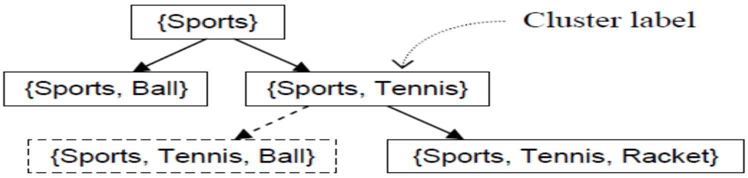


:

### B. Building Cluster Tree

- Each cluster except the root has exactly one parent.
- The topic of parent cluster is more general than the topic of child cluster and they are similar to a certain degree
- Each cluster uses a global frequent k- item set as its cluster label..
- The cluster tree is built bottom up by choosing the best parent at level k-1 for each cluster at level k.

*Example:*



### C. Pruning Cluster Tree

- The cluster tree can be broad and deep which is not suitable for browsing.
- To avoid this, merge the child cluster to its parent cluster if they are having high inter cluster similarity
- The parent cluster now will have all the documents of the child cluster.
- FIHC reduces the dimensionality of the document set.

## V. Fuzzy Frequent Item Set Based Hierarchical Clustering Algorithm

It uses the fuzzy association rule mining algorithm to improve the accuracy of Frequent item set based Hierarchical Clustering (FIHC) method. This approach is a integration of fuzzy set concepts and the association rule mining to find interesting fuzzy association rules from given transactions. The fuzzy association rule mining is a good method because it is easily understandable and realistic for integrating linguistic terms with fuzzy sets

*Steps of Fuzzy frequent item set based Hierarchical Clustering algorithm*

- First the key terms will be extracted from the document set and each document is preprocessed into designated representation for the following mining process. Here hybrid feature selection method will be used to reduce unimportant terms for each document.
- Second, to discover the set of relevant fuzzy frequent item set efficiently, fuzzy association mining rule is used.
- This algorithm calculates three fuzzy values i.e. low, medium, high for each term based on its frequency.
- The derived fuzzy frequent itemsets contain key terms to be regarded as the labels of candidate clusters.
- In third stage, the document will be clustered into hierarchical cluster tree based on these candidate clusters. The cluster tree will be build in top down fashion.

## VI. The Framework Of F$^2$ Ihc

There are three stages in our framework as shown in Fig. 1. We explain them as follows:

1) Document pre-processing: By document pre-processing, the frequency of each term within a document is counted.
2) Candidate clusters extraction: Use our fuzzy association rule mining algorithm to find fuzzy frequent item sets, which are then used to form the candidate clusters.
3) The cluster tree construction: Build the Document-Cluster Matrix (DCM) for assigning each document to a fitting cluster.
   Then, a pruned hierarchical cluster tree will be built.

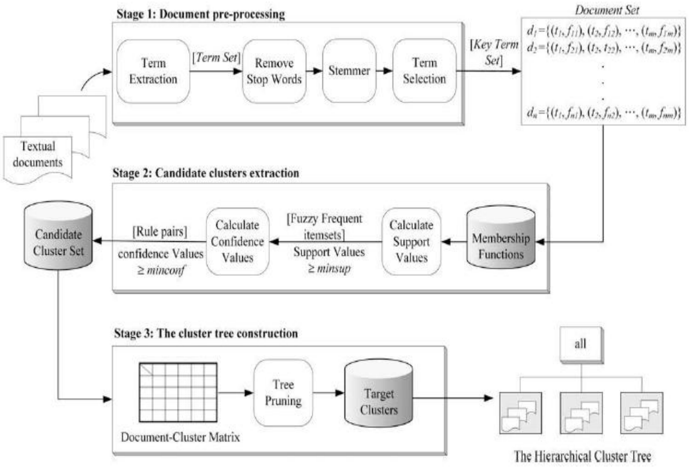

Stage 1: Document pre-processing
Term Extraction
[Term Set]
Remove Stop Words
Stemmer
Term Selection
[Key Term Set]
Document Set
$d_1=\{(t_1,f_{11}),(t_2,f_{12}),\cdots,(t_m,f_{1m})\}$
$d_2=\{(t_1,f_{21}),(t_2,t_{22}),\cdots,(t_m,f_{2m})\}$
$d_n=\{(t_1,f_{n1}),(t_2,f_{n2}),\cdots,(t_m,f_{nm})\}$
Textual documents
Stage 2: Candidate clusters extraction
Candidate Cluster Set
[Rule pairs]
confidence Values ≥ minconf
Calculate Confidence Values
[Fuzzy Frequent itemsets]
Support Values ≥ minsup
Calculate Support Values
Membership Functions
Stage 3: The cluster tree construction
Document-Cluster Matrix
Tree Pruning
Target Clusters
all
The Hierarchical Cluster Tree

Fig.1. The framework of our approach

## VII. Proposed Methodology

This paper propose an effective Fuzzy Frequent Item set-Based Hierarchical Clustering ($F^2$IHC) approach based on the fuzzy association rule mining to ameliorate the accuracy quality of FIHC. Our approach can be distinguished into the following stages:

1) In the first stage, the key terms will be extracted the document set, and each document is pre-processed into the designated representation for the following mining process. In this stage, a hybrid feature selection method will be used to effectively reduce the unimportant terms for each document.
2) In the second stage, to discover a set of relevant fuzzy frequent item sets efficiently, we will propose a fuzzy association rule mining algorithm for text. In this algorithm document is regarded as a transaction, and those term frequency values in a document as the quantitative values. A frequent item set, is defined as a set of words that occur together in some minimum fraction of documents in a cluster. By employing pre-defined membership functions, our algorithm calculates three fuzzy values, i.e., Low, Mid, and High regions, for each term based on its frequency to discriminate the degree of importance of the term within a document in the mining process. The derived fuzzy frequent item sets contain key terms to be regarded as the labels of candidate clusters.
3) In the final stage, the documents will be clustered into a hierarchical cluster tree based on these candidate clusters. The cluster tree will be built in a top-down fashion to recursively select the parent clusters at level k for dividing the documents into its suitable children clusters at level k. Notice that the clusters generated by our algorithm are crisp partitions for assigning a document to exactly one clusters.

## VIII. Experimental Evaluation

Experimental results show that $F^2$ IHC has higher accuracy than FIHC, UPGMA and bisecting k-means when compared on five standard datasets. $F^2$ IHC produces consistent and high quality clusters. The performance of the proposed algorithm is effectively evaluated by comparing with several popular hierarchical document clustering algorithms like bisecting k-means [12, 13, 14]. The produced results are then fetched into the same evaluation program to ensure a fair comparison. The datasets that are used in this paper are listed in table1 below. Classic data set [15] is combined from the four classes CACM, CISI, CRAN, and MED of computer science, information science, aerodynamics and medical articles. Data set Re0 are taken from Reuters – 21578 Text Categorization Test Collection Distribution 1.0 [16]. All the two data sets are real data set.

TABLE1:
STATISTICS FOR TEST DATASETS

| Data Set | No. Of Docs | No. of Classes | Class Size | Avg Class Size | No. of Terms |
|---|---|---|---|---|---|
| Classic | 7094 | 4 | 1033-3203 | 1774 | 12009 |
| Re0 | 1504 | 13 | 11-608 | 116 | 2886 |

## IX. Performance Evaluation By F-Measure

We used the F- measure to evaluate the accuracy of the clustering algorithms. The F- measure is a combination of precision and recall values used in the information retrieval. Each cluster obtained can be considered as a result of query. Whereas each pre-classified set of documents can be considered as a desired set if documents for that query. We treat each cluster if it was the result of a query and each class as if it was the relevant set of documents for a query. The recall, precision and F-measure for natural class Ki and cluster Ci are calculated as follows:

Recall (Ki;Cj) = nij / |Ki| (1)

Precision (Ki;Cj) = nij / |Cj| (2)

Where nij is the number of members of class Ki in cluster Cj .

The corresponding F-Measure F (Ki, Cj) is defined as:

F(Ki;Cj)=[2*Recall(Ki;Cj)*Precision(Ki;Cj) ]/ [ Recall(Ki;Cj) + Precision(Ki;Cj) ]

F(Ki;Cj) represents the quality of cluster Cj in describing class Ki. While computing F (Ki, Cj) in a hierarchical structure, all the documents in the subtree of Cj are considered as the documents in Cj . The overall F-measure, F(C), is the weighted sum of the maximum F-measure of all the classes as defined below:

$$F(C) = \sum_{Ki \in K} |Ki|/|D| \max_{Cj \in C}\{F(Ki,Cj)\}$$

Where, K denotes the set of natural classes; C denotes all clusters at all levels; │Ki│ denotes the number of documents in class Ki And │D│ denotes the total number of documents in the data set.

## X. Conclusion

Experiments show that clustering with reduced dimensionality is significantly more efficient and scalable. FIHC is also accurate. Clustering accuracy of FIHC consistently outperforms other methods. Number of documents is growing continuously and it is infeasible to rebuild the cluster tree upon every arrival of new document. Using FIHC a new document can be assign to most similar cluster but the clustering accuracy may degrade in the course of time . This article describes an original algorithm to update a numerical taxonomy after addition of new patterns in a database. Tests have shown that using this algorithm allow to progressively perform a hierarchical clustering of big sets of data, which can then contain seven times more elements than using the classical algorithm. In spite of the computational cost of the method, it can be said that the two significant reduction of the computational cost of the towards the use of hierarchical clustering in industrial applications, thus offering an interesting alternative to partitioned clustering techniques. Further works should lead to significant reduction of the computational cost of the method by integrating already available optimization algorithms and adding more efficient rules of building.

## *References*

[1] A. K. Jain, M.N. Murty and PJ .Flynn, "*Data clustering: a review*", ACM Computing Surveys, vol. 3I (3), pp. 264-323, 1999.

[2] X. Rui, *"Survey of clustering algorithms"*, IEEE Transactions on Neural Networks, Vol. 16, No. 3, pp. 645- 678, 2005.

[3] Lewis. D. D., Yang. Y. Rose, T. G., & Li. F. *"RCV1: A new Benchmark collection for text categorization research"*. Journal of Machine Learning Research, 5, 361-397, 2004

[4] Lifeng Wang, Hui Song, Xiaoqiang Liu, "Incremental Document Clustering Using Multirepresentation Indexing Tree", 2010.

[5] Nachiketa Sahoo, "Incremental Hierarchical Clustering of Text Documents", 2006

[6] Khaled M. Hammouda Mohamed S. Kamel, "Incremental Document Clustering Using Cluster Similarity Histograms".

[7] Dwi H. Widyantoro & Thomas R. Ioerger & John Yen "An Incremental Approach to Building a Cluster Hierarchy", 2002

[8] Xiaoke Su, Yang Lan, Renxia Wan, and Yuming Qin, "A Fast Incremental Clustering Algorithm",Proceedings of the 2009 International Symposium on Information Processing (ISIP'09), Huangshan, P. R. China, August 21-23, 2009, pp. 175-178

[9] Arnaud Ribert, Abdel Ennaji, Yves Lecourtier, "An Incremental Hierarchical Clustering", pp. 586-591.

[10] M. Srinivas and C. Krishna Mohan, "Efficient Clustering Approach using incremental and Hierarchical Clustering Methods", 2010 IEEE

[11] Chun-Ling Chen, Frank S.C Tseng, Tyne Liang *"Mining fuzzy frequent itemsets for hierarchical document clustering"*, Information Processing and Management 46, 193-21, 2010.

[12] L. Zhuang, and H. Dai. 2004. A Maximal Frequent Itemset Approach for Document Clustering.Computerand Information Technology, CIT. The Fourth International Conference, pp. 970 – 977.

[13] R. C. Dubes and A. K. Jain. 1998. Algorithms for Clustering Data. Prentice Hall college Div, Englewood Cliffs, NJ, March.

[14] G.Karypis.2002.Cluto 2.0clustering http://wwwusers.cs.umn.edu/˜ karypis/cluto

[15] Classic. ftp://ftp.cs.cornell.edu/pub/smart/.

[16] H. Han, B. Boley, M. Gini, R. Gross, K. Hastings,G. Karypis, V. Kumar,B. Mobasher, and J. Moore. 1998. Webace: a web agent for document categorization and Exploration. In Proceedings of the second international.

[17] Chun-Ling Chen, Frank S.C. Tseng , Tyne Liang 2010. Mining fuzzy frequent itemsets for hierarchical document clustering

[18] Douglass R. Cutting, David R. Karger, Pedersen Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval , Interface Design and Display, pages 318–329, 1992.

[19] A. K. Jain, M. N. Murty, and P. J. sFlynn. Data clustering: a review. ACM Computing Surveys , 31(3):264–323, 1999.