ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Document Clustering In Distributed Environment

J.Jayabharathy

Assistant Professor, Dept. of CSE, Pondicherry Engineering College, Puducherry, India.

ABSTRACT: Document clustering has emerged as a widely used technique with the increase in large number of documents that is getting accumulated day by day in various fields like news groups, government organizations, Internet and digital libraries. The increase in the number of data sets leads to the increase in difficulty in organizing and retrieving the data. Data can be retrieved easily and more efficiently when it is grouped in a specific category. A good document clustering algorithm should have high intra-cluster similarity and less inter- cluster similarity. This process could be achieved in a centralized or distributed environment. The centralized approach increases the overhead of the process as numerous documents need to be fetched and clustered at the same time as a whole. In distributed environment the input documents are partitioned among a number of systems which perform the clustering of the documents individually. These clusters in turn communicate with the peers to perform clustering recursively. Hierarchical clustering is made use of here, as clusters iteratively communicate providing the higher level of clusters. At the end of the process one global cluster is obtained. The reduction in the overhead makes the document clustering in distributed environment more feasible. In order to gain more accuracy the nodes should be connected or the granularity of information exchanged between the nodes should be increased.

KEYWORDS: Document clustering, Distributed Environment, K-Means Algorithm, Peer to peer, Hierarchical Clustering

I. INTRODUCTION

Data mining in distributed environments is known as DDM, and sometimes as Distributed Knowledge Discovery (DKD). The central assumption in DDM is that data are distributed over a number of sites and that it is desirable to derive, through data mining techniques, a global model that reflects the characteristics of the whole data set. Huge data sets are being collected daily in different fields; e.g., retail chains, banking, biomedicine, astronomy, and so forth, but it is still extremely difficult to draw conclusions or make decisions based on the collective characteristics of such disparate data. Four main approaches for performing Document Clustering in distributed environment can be identified.

- A common approach is to bring the data to a central site, then apply centralized clustering on the collected data. Such approach clearly suffers from a huge communication and computation cost to pool and mine the global data. In addition, we cannot preserve data privacy in such scenarios.
- A smarter approach is to perform local clustering at each site to produce a local model. All local models can then be transmitted to a central site that combines them into a global model.
- Another smart approach is for each site to carefully select a small set of representative data objects and transmit it to a central site, which combines the local representatives into one global representative data set. Clustering can then be carried on the global representative data set.
- A more departing approach does not involve centralized operation, and thus belongs to the peer-to-peer (P2P) class of algorithms [1]. In P2P DDM, sites communicate directly with each other to perform the data mining task.

II. RELATED WORK

In distributed data mining is that data are distributed over a number of sites and that it is desirable to derive, through data mining techniques, a global model that reflects the characteristics of the whole data set[2]. Quality of the global model derived from the data should be either equal or comparable to a model derived using a centralized method.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Finally, in some situations when local data are sensitive and not easily shared, it is desirable to achieve a certain level of privacy of local data while deriving the global model.

Applications. Applications of DDM are numerous and are usually manifested as distributed computing projects. They often try to solve problems in mathematics and science. Specific areas and sample projects include: astronomy, biology, climate change, physics, cryptography, and biomedicine. Those projects are usually built on top of a common platform providing low level services for distributed or grid computing.

Grid computing is a broad sub category of distributed computing. It is considered as "virtual supercomputer" composed of a network of loosely-coupled computers, acting in concert to perform very large tasks. Grid is a type of a distributed system which supports the sharing and coordinated use of resources, independently from their physical type and location. The availability of low cost powerful computers coupled with popularity of the internet and high speed networks have led the distributed computing environment to be mapped from classical distributed to grid environments [7].

A similar system can be found in [8], but the problem is posed from the information retrieval point of view. In this work, a subset of the document collection is centrally partitioned into clusters, for which “cluster signatures” are created. Each cluster is then assigned to a node, and later documents are classified to their respective clusters by comparing their signature with all cluster signatures. Queries are handled in the same way, where they are directed from a root node to the node handling the cluster most similar to the query.

Document Clustering Methods

The main goal of clustering is maximizing both the homogeneity within each cluster and the heterogeneity among different clusters. In other words, objects that belong to the same cluster should be more similar to each other than objects that belong to different clusters. The problem of measuring similarity is usually tackled indirectly, i.e., distance measures are used for quantifying the degree of dissimilarity among objects, in such a way that more similar objects have lower dissimilarity values. Compared with structured data in database, because of lingual diversity, text data with unstructured form is more diversiform and complex. Researchers have shown that under the current automatic understanding level of natural language, word is still the best unit for text representation and processing [2].

The dimensionality can be cut down by removing stop-words and words with high frequency. Stop words are usually given as a word list. Most of these words are conjunctions or adverbs which have no contribution to cluster process, and sometimes have negative influence. Words with high frequency which can be gotten in word frequency dictionary appear in most documents, so they are not helpful for cluster either. Words appear in no more than three documents and at least 33% of all documents can be removed.

Document Clustering is increasingly widespread. It is finding application in browsing, in improving the similarity of search tools and in automatically generating thesauri. In query analysis clustering has been used for transforming a free text query into a fuzzy Boolean constraint. The popularity of Yahoo! demonstrates the potential of categorization for presenting information on the World Wide Web [5].
Document Clustering has been extensively investigated as a methodology for improving document search and retrieval. The general assumption is that mutually similar documents will tend to be relevant to the same queries, and, hence, that automatic determination of groups of such documents can improve recall by effectively broadening a search request. Typically a fixed corpus of documents is clustered either into an exhaustive partition, disjoint or otherwise, or into a hierarchical tree structure. In the case of a partition, queries are matched against clusters and the contents of the best scoring clusters are returned as a result, possibly sorted by a score. In the case of a hierarchy, queries are processed downward, always taking the highest score branch, until some stopping condition is achieved. The subtree at that point is then returned as a result. Hybrid strategies are also available. These strategies are essentially variations of near neighbor search where nearness is defined in terms of the pairwise document similarity measure used to generate the clustering. Indeed, cluster search techniques are typically compared to direct near-neighbor search and are evaluated in terms of precision and recall.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

Document Clustering has also been studied as a method for accelerating near-neighbor search, but the development of fast algorithms for near-neighbor search has decreased increased in that possibility

III. PROPOSED ALGORITHM

A. HIERARCHICAL PEER – TO PEER CLUSTERING:

An approach for distributed document clustering based on structured peer to peer network architecture is our objective. The HP2PCTC model is based on static hierarchical structure that is designed up front, upon which the peer network is formed. The goal is to achieve a flexible DDM model that can be tailored to various scenarios. The proposed model is called the **Hierarchically Distributed P2P Clustering for Text Documents (HP2PCTC)**. HP2PCTC is a hierarchically distributed P2P architecture for scalable distributed clustering of horizontally partitioned data. A scalable distributed clustering system should involve hierarchical distribution. A hierarchical processing strategy allows for delegation of responsibility and modularity.

The Peer to Peer architecture is based on the data they have access to. On moving up the hierarchy, clusters are merged from lower levels involves a hierarchy of P2P neighborhoods, in which the peers in each neighborhood are responsible for building a clustering solution, using P2P communication in the hierarchy. At the root of the hierarchy, one global clustering can be derived. The model deviates from the standard definition of P2P networks, which typically involve loose structure (or no structure at all), based on peer connections that are created and dropped frequently. The HP2PCTC model, on the other hand, is based on static hierarchical structure that is designed up front, upon which the peer network is formed. Using the HP2PCTC model, we can partition the problem in a modular way, solve each part individually, and then successively combine solutions if it is desired to find a global solution.

The model lends itself to real-world structures, such as hierarchically distributed organizations or government agencies. In such scenario, different departments or branches can perform local clustering to draw conclusions from local data. Parent departments or organizations can combine results from those in lower levels to draw conclusions on a more holistic view of the data. HP2PCTC is a hierarchically distributed P2P architecture for scalable distributed clustering of horizontally partitioned data. The communication between nodes is accomplished by their supernodes. Supernodes are representative from each node. The notion of a node accompanied by a supernode can be applied recursively to construct a multilevel overlay hierarchy of peers; i.e., a group of supernodes can form a higher level neighborhood, which can communicate with each other on that particular level of hierarchy. The number of files is distributed across a number of nodes such that each node has a separate number of input files of their own which reduces the overhead due to the collection of large number of files and the operation to be performed on them. This type of hierarchy is shown in fig 1.

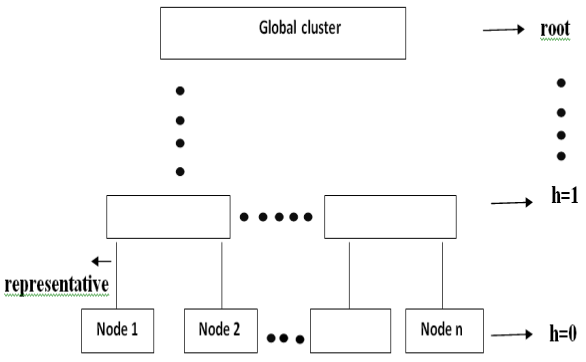


Fig 1 Hierarchy Structure

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

In the Fig 1 each node acts as a separate environment or system. The files are randomly distributed to each node. Each node locally clusters its documents and those clustered documents acts as the representative for the next level of clustering

B. DESCRIPTION OF THE DISTRIBUTED DOCUMENT CLUSTERING ALGORITHM

The steps followed in the process of Distributed Document Clustering Algorithm are given below:

The documents that are to be clustered are distributed among various nodes included in the distributed environment. The documents in each node are preprocessed using the famous Porter Stemmer algorithm.

Using the Stemmer algorithm the frequent word in each document is found. With the help of these frequent words the documents in each node are locally clustered using k-means algorithm. These locally clustered documents communicate with each other to form the next level of hierarchy. The process continues until a global cluster is available in all the nodes.

Porter Stemmer Algorithm

The Porter Stemmer algorithm is the traditional method for text preprocessing. It stems all the prefixes and suffixes of the word and ignores all the stop words. So the document contains only the root words. These root words are further used for finding the most frequently occuring word in the document. This frequent word is used for further clustering of the document.

K-Means Algorithm

The k-means clustering requires co-ordinates to be assigned for each data points for obtaining the similarity measure. After finding the frequent word each document is compared with a standard dictionary containing the common and most frequently occurring words of a particular domain. These dictionaries are provided for clustering the documents under domains we require to group. Documents that fall under the same domain will be assigned co-ordinates that are nearby and those belong to different domains have a high difference in co-ordinates. Such way the clusters will have more intracluster similarity and less intercluster similarity. These co-ordinates are assigned by a random function. The co-ordinates of each data points or documents is shown along with the frequent word of the document for better understanding. The k-means clustering algorithm requires the number of clusters k and the documents to be given as input. Hence the clustering will be based on the number of clusters given as an input. If the input document doesn't match any domain it will be clustered into a separate group of similar documents. The pseudo code for k-means algorithm

Initialize $\mathbf{m}_i$, i = 1,…,k, for example, to k random $\mathbf{x}^t$

Repeat

For all $\mathbf{x}^t$ in X

$b_i^t \leftarrow 1$ if $\| \mathbf{x}^t - \mathbf{m}_i \| = \min_j \| \mathbf{x}^t - \mathbf{m}_j \|$

$b_i^t \leftarrow 0$ otherwise

For all $\mathbf{m}_i$, i = 1,…,k

$\mathbf{m}_i \leftarrow$ sum over t ($b_i^t \mathbf{x}^t$) / sum over t (b_i^t)

Until $\mathbf{m}_i$ converge

The vector **m** contains a reference to the sample mean of each cluster. **x** refers to each of our examples, and **b** contains our "estimated labels"

Hierarchical Clustering Algorithm

The HP2PCTC algorithm is a distributed iterative clustering process. It is a centroid-based clustering algorithm, where set of cluster centroids is generated to describe the clustering solution. In HP2PCTC, each neighborhood converges to a set of centroids that describe the data set in that neighborhood. The distributed clustering strategy within a single neighborhood is similar to the parallel K-means algorithm [17] in that the final set of centroids of a neighborhood will be identical to those produced by centralized K-means on the data within that neighborhood. Other neighborhoods, either on the same level or at higher levels of the hierarchy, may converge to another set of centroids. Once a neighborhood converges to a set of centroids, those centroids are acquired by the supernode of that neighborhood. The

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

supernode, in turn as part of its higher level neighborhood, collaborates with its peers to form a set of centroids for its neighborhood. This process continues hierarchically until a set of centroids is generated at the root of the hierarchy.

C. IMPLEMENTATION DETAILS

The k-means and hierarchical clustering algorithm have been implemented using Java Netbeans software. The databases used to store the nodes or systems connected in LAN and their corresponding files and clusters are created using Mysql. The components used in the clustering process are the following.

- Stemmer → Stems the words in the document by removing the stop words and prefixes and suffixes of the word
- Frequent word finder → Finds the most frequently occurring word in the document
- Word counter → Counts the total number of words and calculate the word frequency

Configuring Ftp Sites

Each system is installed with IIS and FTP server. A virtual directory is created in the FTP site and the files to be given as input are included in the directory. A system that is presently connected is shown in the user interface. Any site can be added or removed from the network. When a site is added all the input files are added to the site and is included in clustering. First he FTP object is invoked and is connected to include all the files in the network.

Porter Stemmer Algorithm

Text Preprocessing is accomplished using Porter Stemmer algorithm. The documents that are available in each system in the virtual directory are taken. To this Porter Stemmer algorithm is applied. By using Porter Stemmer algorithm, The words in the document are converted to lower case. The stop words are stored in a file and compared with the document taken for preprocessing and thereby stop words are removed The suffix stripping process will reduce the total number of terms and hence reduce the size and complexity of the data in the system. Hence the suffixes like 'ed', 's', 'es' etc are removed from the words. From the remaining set of stemmed words, a count is set for every particular word. If a word occurs many times its count will be incremented by one for each occurrence of that particular word in the document. Using the above method the most occurring word in every document is found.

K-means clustering

K-Means assigns each document in the dataset to only one of the initial clusters. Each record is assigned to the nearest cluster (the cluster which it is most similar to) using a similarity measure. The similarity measure is calculated using the most frequent word obtained in each document. The frequent word is matched with the dictionary maintained for each domain and if the word matches with any domain, co-ordinates(x,y) is assigned to that document based on the domain it belongs to. For eg, if a document's frequent word matches with the computer domain, its co-ordinates(x,y) will be assigned any number between 10 and 20. Similarly if the document's frequent word matches with the medical domain, co-ordinates will be assigned between 200 and 300. Now the documents whose co-ordinates are near will be clustered together. The preceding steps are repeated until stable clusters are formed and the K-Means clustering procedure is completed. Stable clusters are formed when new iterations or repetitions of the K-Means clustering algorithm does not create new clusters. Now k clusters are created and we can view the clusters formed from each local site.

Hierarchical clustering of locally clustered documents

The hierarchical clustering is formed with the help of local tree formed in each local site. The local trees now communicate with each other to form the global tree which contains the documents in all nodes.

IV. SIMULATION RESULTS

Centralized and Proposed Hierarchically Distributed P2P Clustering for Text Documents (HP2PCTC)

The performance of the clustering algorithm is measured based on the clustering speed. The two clustering algorithms namely Centralized and Hierarchical clustering algorithms have been implemented in Java. Experiments were carried out to compare the performances of these two algorithms under different number of documents. The variation of clustering speed with the change in number of documents is studied for these algorithms. By comparing the clustering

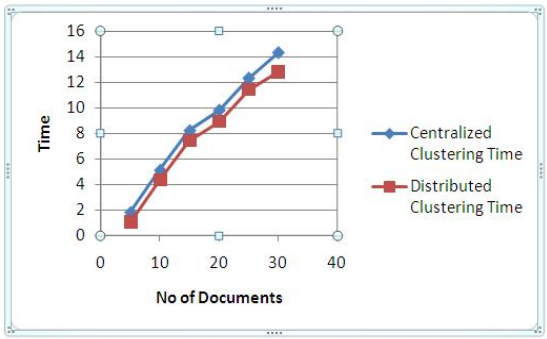
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2015

speed for various numbers of documents, we can see that the Hierarchical clustering algorithm has relatively good performance and the graph is shown in Fig. 2. This shows that the time taken for the distributed clustering algorithm is less compared to the centralized clustering algorithm. This could be achieved without compromising the quality of the clusters.



Performance of Centralized versus Distributed Clustering Algorithm

Fig 2 Performance of Centralized versus Distributed Clustering Algorithm

V. CONCLUSIONS AND FUTURE ENHANCEMENTS

This paper discusses about novel architecture and algorithm for distributed clustering, the HP2PCTC model, which allows building hierarchical networks for clustering data. We demonstrated the flexibility of the model, showing that it achieves comparable quality to its centralized counterpart while providing significant speedup and that it is possible to make it equivalent to traditional distributed clustering models.

For future work, we plan to extend this model to be dynamic, allowing nodes to join and leave the network, which requires maintaining a balanced network in terms of partitioning and height. This will also lead us to a way to find the optimal network height for certain applications. We also plan to extend it to allow merging and splitting of complete hierarchies. We are also investigating the possibility of making the clustering algorithm more global by allowing centroids to cross neighborhoods through higher levels; i.e., clusters at lower level neighborhoods should be a function of higher level centroids. We believe that this will create an opportunity for better global clustering solutions but on the expense of computational complexity.

REFERENCES

1. Yi Peng, Gang Kou, Yong Shi, Zhengxin chen , " A Hybrid Strategy for Clustering Data Mining Documents," IEEE international conference on data mining-workshops,2006
2. Khaled M. Hammouda and Mohamed S.kamel, "Hierarchically Distributed Peer-to-Peer Document Clustering and Cluster Summarization," IEEE transactions on knowledge and data engineering, vol. 21 , no.5, May 2009
3. N.F. Samatova, G. Ostrouchov, A. Geist, and A.V. MelechkoRACHET: "An Efficient Cover-Based Merging of Clustering Hierarchies from Distributed Datasets," Distributed and Parallel Databases, vol. 11, no. 2, pp. 157-180, 2002.
4. M.F. Porter, "An Algorithm for Suffix Stripping," Program, vol. 14, no. 3, pp. 130-137, July 1980.
5. Jiawei Han and Micheline Kamber", Data Mining Concepts and techniques", , Second Edition.
6. Craig Silverstein Xerox Palo Alto Hinrich Schiitze, "Projections For Efficient Document Clustering", Research Centre.roceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, Pages 74-81, 2007
7. Neeraj Nehra, R. B. Patel and V. K. Bhat, "Load Balancing with Fault Tolerance and Optimal Resource Utilization in Grid Computing", *Information Technology Journal 6 (6): 784-797*, 2007.
8. J. Li and R. Morris, "Document Clustering for Distributed Fulltext Search," Proc. Second MIT Student Oxygen Workshop, Aug. 2002.