Ontology Based Fuzzy Document Clustering Scheme for Distributed P2P Network

By Thangamani M, Dr. P. Thangaraj

Engineering College

Abstract: Text document clustering is the technique used to group up the document with the reference to the similarity. It is widely used in digital library environment. Ontology is playing an increasingly important role in knowledge management and the semantic web. Distributed data clustering means clustering is done in distributed site. Peer-to-peer (P2P) system provides shared access to resources that are spread over the network. To address the problem of modularity, flexibility, and scalability in distributed peer-to-peer, a hierarchically distributed peer-to-peer (HP2PC) architecture clustering algorithm is used. By analyzing all types of techniques for distributed document cluster, Semantic P2P is determined to be better. In this paper semantic analysis mechanism is enhanced with fuzzy document clustering scheme in p2p distributed network, which has a capability of providing better clustering result than Semantic distributed p2p clustering. Our experiments are conducted on 3 datasets: 20-Newsgroup, Reuter’s-21578 and IEEE abstract with key index. The evaluation result indicates the better clustering result by using the fuzzy concept with semantic distributed p2p proposed techniques.

Keywords: Clustering, text mining, ontology, distributed data mining, peer-to-peer network, document clustering

Classification: GJCST Classification: FOR Code: 100503, 080202, 100601

Strictly as per the compliance and regulations of:

© 2011 Thangamani M, Dr. P. Thangaraj. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/, permitting all non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Ontology Based Fuzzy Document Clustering Scheme for Distributed P2P Network

Thangamani . M*, Dr. P.Thangaraj

Abstract - Text document clustering is the technique used to group up the document with the reference to the similarity. It is widely used in digital library environment. Ontology is playing an increasingly important role in knowledge management and the semantic web. Distributed data clustering means clustering is done in distributed site. Peer-to-peer (P2P) system provides shared access to resources that are spread over the network. To address the problem of modularity, flexibility, and scalability in distributed peer-to-peer, a hierarchically distributed peer-to-peer (HP2P) architecture clustering algorithm is used. By analyzing all types of techniques for distributed document cluster, Semantic P2P is determined to be better. In this paper semantic analysis mechanism is enhanced with fuzzy document clustering scheme in p2p distributed network, which has a capability of providing better clustering result than Semantic distributed p2p clustering. Our experiments are conducted on 3 datasets: 20-Newsgroup, Reuter’s-21578 and IEEE abstract with key index. The evaluation result indicates the better clustering result by using the fuzzy concept with semantic distributed p2p proposed techniques.

Keywords: Clustering, text mining, ontology, distributed datamining, peer-to-peer network, document clustering.

I. Introduction

A tremendous growth in the volume of online text documents available on the Internet, digital libraries, news sources, and company-wide intranets. It has been forecasted that these documents will become the predominant data type stored online. This provides a huge opportunity to make more effective use of these collections and there is a growing need for tools to deal with text documents. Document clustering, is an important task that can help people to find information on these huge resources. Libraries act as the common ground of serving the society with the infinite source of knowledge. Digital library is evolved as the offspring of traditional manual oriented libraries. All the books, articles and materials are maintained in an electronic form in digital libraries. To group documents of various types and to extract important content from the collection of large text, clustering algorithms that depend upon similarity and ontology measure can be used [3] [4].

In many applications, a document may include multiple topics and thus may relate to multiple categories at the same time, resulting in the requirement of fuzzy document clustering. On the other hand, due to its effectiveness in discovering clusters with overlapping boundaries, fuzzy clustering algorithms are able to reveal more accurate cluster structures within the document collections. Statement of problem: As large amount of text documents are available in the World Wide Web, business document management system has evinced the significant task of separating texts dynamically into new categories for every intelligence business management. Text clustering algorithms employed at present involve problems of practical applicability and low accuracy as they take up term frequency based method of document clustering. Furthermore, they do not cater to local optimization based clustering and term relationship in extracting documents. The accuracy of clustering is not optimized one. The complication will occur when clustering the document in different peers and affect the clustering process. To optimize this problem by incorporating the fuzzy technique.

II. Previous Research

In document clustering, unlabeled documents are represented in vector space model (VSM), where each document is a vector in the word space and each element of the vector indicates the frequency of the corresponding word also called term or feature in the document. The typically, the data are of very high dimensional and sparse, which poses a big challenge to conventional clustering algorithms such as k-means [1]. In high dimensional data, clusters often exist in subspaces rather than in the entire space [2]. One
solution to this problem in text subspace clustering [4],[5],[6], which aims to discovering the document clusters in different subspaces of the original word space. Fuzzy clustering [7] in contrast to the usual (crisp) methods does not provide hard clusters, but returns a degree of membership of each object to all the clusters. In [8], a feature-weighting algorithm combined with the fuzzy K prototypes algorithm was presented. An algorithm named fuzzy W-K-Means [9] was proposed, which are difficult to estimate in practice. L.chen et al presents, a new algorithm named Fuzzy projection clustering (FPC) [10] [11].

Distributed Data Mining (DDM) started gaining importance in the late 1990s. Although, it is still one of the budding areas of research, the literature on DDM constitutes a sizeable portion in data mining broadly. Data mining in distributed environments is sometimes known as DDM and sometimes as Distributed Knowledge Discovery (DKD). In DDM, it is presumed that data are distributed over a number of sites and that it is desirable to derive those data through data mining process. It is accepted as a global model as it reflects the characteristics of the entire data set. A number of challenging issues arise while developing DDM that are:
- Communication model and complexity,
- Quality of global model and
- Privacy of local data.

Methods that involve low communication complexity are feasible to be developed especially in mobile application like sensor networks but communication in such an application consumes battery power. The quality of the global model derived from the data set must be either equal or compatible to a model extracted in a centralized method. Even in situations wherein local data are quite sensitive and cannot be easily shared, it would be desirable to obtain a degree of privacy in deriving the global model. Although not yet proven, deriving high-quality models usually require sharing as much data as possible but at the cost of higher communication and privacy.

Commonly, the two different types of data distribution present are: homogeneous and heterogeneous. In homogeneous, the first type, data are partitioned horizontally across the sites; i.e., each site holds a subset of the original data. In heterogeneous, the second type, data are partitioned vertically; i.e., each site holds a subset of the attribute space and the data among sites are linked via a common key.

**Exact versus approximate DDM algorithms**: A DDM algorithm can be described as either exact or approximate algorithm. Exact algorithms produce a final model that is identical to a hypothetical model being generated by a centralized process having access to the full data set. The hypothetical process is modeled on an exact distributed clustering algorithm. The exact algorithm works on the data subsets, Di, of each node combined into one data set, D, first; then on centralized clustering algorithm, A, and performs clustering process on the entire data set. The clustering solutions derived are then distributed by intersecting the data subsets from the global clustering solution. Approximate algorithms, on the other hand, develop a model that closely approximates a centralized model. Most DDM research studies focus on approximate algorithms as they tend to produce comparable results with far less complexity [12].

**Communication models**: Communication between nodes in distributed clustering algorithms can be categorized into three classes: 1) communicating models, 2) Communicating representatives, and 3) communicating actual data. First class involves calculation of local models that are then sent to peers or to a central site. Models often comprise cluster centroids, e.g., P2P K-means[13] [14] cluster dendograms, e.g., RACHET [15] [31], or generative models, e.g., DMC Merugu and J.Ghosh [16]. In the second case, nodes select a number of representative samples of the local data to be sent to a central site for global model generation, as in the case of KDEC distributed clustering algorithm [17] and the DBDC algorithm J.Da silva [18]. In the last model of communication, nodes exchange actual data objects; i.e., data objects can change sites to facilitate construction of clusters that exist only in certain sites like the case of collaborative clustering scheme [19]. A similar scheme can be found in [20], but it involves a problem which lies actually in information retrieval. In this, a subset of the document collection is centrally partitioned into clusters creating “cluster signatures”. Each cluster is then assigned to a node. Later on, documents are ordered into respective clusters by comparing their signatures with all other cluster signatures. Queries are also handled in the same way. They are directed from root node to the node handling the cluster most similar to the query. In the latest issue of IEEE Internet Computing [21], a few algorithms highlighting the state of the art in DDM were given. Qing Ma et al. [22], presented query based clustering algorithm. Chang liu [23] recommended query based search for structure p2p network but this method would consume lot of space for storage in peers. S.Datta et al [24] depicted an exact local algorithm for monitoring K-means clustering R. Wolff et al as well as S.Datta et al advocated approximate local K-means clustering algorithm for P2P networks.

Although K-means monitoring algorithm does not generate distributed clustering, normally, it helps centralized K-means process know when to recompute the clusters by monitoring the distribution of centroids across peers and trigger reclustering in case the data distribution getting changed over time. On the other hand, P2P K-means algorithm S.Datta et al [14] updates the centroids at each peer as per the information received from their immediate neighbors. This process gets terminated when the information received does not
result in significant update of the centroids of all peers. The P2P K-means algorithm finds its roots in a parallel implementation of K-means as proposed by Dhillon and Modha.

The author Deise de Brum Saccol [25], proposed a system that involved an increase in the semantic through considering both information search and storage. Chang Liu et al [26] introduced distributed document clustering for search engine. Qing He et al [27] propounded a text clustering algorithm based on frequent term sets for peer-to-peer networks. R. Wolff et al [28] presented, thresholding based Data mining in Peer-to-Peer Systems for local optimization. Author [29] introduced semantic-based P2P Resource Management System yet it supports only local optimization. Many other introduced centroid based algorithm for text categorization [32]. A new algorithm presented for fuzzy document clustering in [33].

Unfortunately, most of the above methods make little of these in the context of high dimensional clustering that lead to unstable clustering results. We will improve these in the context of high dimensional clustering that corresponds to a subset of dimensions in the data space. A cluster of documents is situated in a subspace of the original space.

In this paper, also we will present application of fuzzy in distributed semantic P2P network.

III. Motivation

It is obvious from the discussion made earlier that the existing techniques do not afford better accuracy. The time utilized for active clustering of documents is more if large databases are taken up for clustering. In the case of determining the initial clusters also, varying clusters would result for the same dataset. The proposed clustering algorithm involves the grouping of electronic documents, extracting important contents from the document collection and supports effective management of digital library documents. Contents of digital documents are analyzed and grouped into various categories. The system is designed to link n number of libraries with n number of digital documents. In short, content analysis, document grouping and content extraction are the main functions of the proposed system.

The usage of Fuzzy will provide better clustering. In this paper Semantic information retrieval is enhanced with Fuzzy to yield the better clustering accuracy for large databases.

IV. The R-FPC Algorithm

Given a vector space model, the documents vectors may be presented by $x_1, x_2, \ldots, x_n$, where $x_i=(x_{i1}, x_{i2}, \ldots, x_{id})$ and $d$ stands for the number of unique words in the model, $n$ denotes the total number of documents, $x_{ij}$ is the normalized word frequency of the $j$th term in the document. $x_i$ is a data point in the $d$-dimensional space. Let $\{C_1, C_2, \ldots, C_K\}$ be the $K$ document clusters, where $C_k$ denotes a partition of document collections. The membership of $x_i$ to $C_k$ is denoted as $u_{ki}$.

In text subspace clustering, each category of documents is characterized by a subset of terms in the vocabulary that corresponds to a subset of dimensions in the data space. An extended Gaussian model is build for subspace clustering and derived the above objective function. Here $\alpha_k$ is the mixture coefficient of $k$th Gaussian component, $\nu_k=(\nu_{k1}, \nu_{k2}, \ldots, \nu_{kd})$ and $\sigma_k$ denote mean and covariance of the $k$th Gaussian, respectively. V and W denote the mean and weight matrix for all the $K$ clusters, respectively.

Since all inputs $x_1, x_2, \ldots, x_n$ are available, the learning of all the parameters via minimizing Eqn.(1) can be implemented by the Expectation Maximization (EM) algorithm in a batch way. The contraction of R-FPC algorithm is based on FPC. R-FPC algorithm introduces a new procedure called R-Greedy to build a robust initial condition for the algorithm.

Algorithm 1 R-FPC

Input: $x_1, x_2, \ldots, x_n$, K and a termination criterion $\varepsilon$

Output: $U = \{u_{ki}|k=1,2,\ldots,K,i=1,2,\ldots,n\}$ and the associated weights matrix $W$.

begin
1 Initialization
1.1 Let $p$ be the number of iteration, $p = 0$;
1.2 Call R-Greedy to initialize the $V^{(0)}$ and $W^{(0)}$;
1.3 Set $u_{ki} = 1/K$ for $k=1,2,\ldots,K$ and $i=1,2,\ldots,n$;
Set $\alpha_k$ and $\sigma_k$ to an constant.
2 Repeat
2.1 Set $p=p+1$
2.2 Use Eqn.(3) to calculate $U(p)$
2.3 Use Eqn.(2) to calculate $W(p)$
2.4 Update $\alpha_k(k=1,2,\ldots,K)$ using Eqn.(5)
end
2.5 Update \( \alpha_k \) for \( k = 1, 2, \ldots, K \) using Eqn. (6)
2.6 Use Eqn. (4) to calculate \( V(p) \)
   \[ u_k = (\Sigma_{i=1}^{K} w_i/\Sigma_{i=1}^{K} u_i) \]

   \[ v_k = \frac{\Sigma_{i=1}^{K} u_i x_i}{\Sigma_{i=1}^{K} u_i} \]

   \[ \alpha_k = (u_k/n) \alpha \]

   \[ \sigma_k^2 = 1/d \Sigma_{i=1}^{K} u_i \Sigma_{j=1}^{K} w_i \Sigma_{i=1}^{K} u_i (x_i - v_j)^2 \]

   \[ w_{ij} = (1/X_i + \delta)^2 / (\Sigma_{j=1}^{K} 1/X_j + \delta)^2 \]

   \[ \delta = 1/n d \Sigma_{i=1}^{K} (x_i - \bar{x})^2 \]

   where \( \bar{x} \) is the mean feature value of the entire data set.

   The above measurement of fuzzy membership degree is in an exponential type and is based on the weighted Euclidean distance, which is quite different from the one used in the FCM-based algorithm, such as the classical FCM and the newly designed algorithm fuzzy W-k-means. The means of Gaussian, i.e., the cluster center \( V \), can be calculated by

   \[ v_k = \frac{\Sigma_{i=1}^{K} u_i x_i}{\Sigma_{i=1}^{K} u_i} \]

   \[ \alpha_k = (u_k/n) \alpha \]

   \[ \sigma_k^2 = 1/d \Sigma_{i=1}^{K} u_i \Sigma_{j=1}^{K} w_i \Sigma_{i=1}^{K} u_i (x_i - v_j)^2 \]

   \[ w_{ij} = (1/X_i + \delta)^2 / (\Sigma_{j=1}^{K} 1/X_j + \delta)^2 \]

   Following FWKLM, to ensure that the denominator of Eqn.(2) is always larger than 0, adjust the denominator by adding an additional factor \( \delta = 1/n d \Sigma_{i=1}^{K} (x_i - \bar{x})^2 \), where \( \bar{x} \) is the mean feature value of the entire data set. The proof of Eqn.(2) can be found. Similarly, the membership matrix \( U \) in each iteration is updated by

   \[ u_{ij} = \frac{\Sigma_{k=1}^{K} \delta \exp (-1/2d \alpha_k^2 \Sigma_{j=1}^{K} w_{ij} (x_j - v_k)^2)}{\Sigma_{k=1}^{K} \delta \exp (-1/2d \alpha_k^2 \Sigma_{j=1}^{K} w_{ij} (x_j - v_k)^2) - 1} \]

   \[ \alpha_k = (u_k/n) \alpha \]

   \[ \sigma_k^2 = 1/d \Sigma_{i=1}^{K} u_i \Sigma_{j=1}^{K} w_i \Sigma_{i=1}^{K} u_i (x_i - v_j)^2 \]

   \[ w_{ij} = (1/X_i + \delta)^2 / (\Sigma_{j=1}^{K} 1/X_j + \delta)^2 \]

   \[ \delta = 1/n d \Sigma_{i=1}^{K} (x_i - \bar{x})^2 \]

   It can be seen that the R-FPC is an extension to the FCM algorithm by adding multiple steps to estimate the parameters of the clustering model. Therefore the algorithm is able to converge within a finite number of iterations. The time complexity is \( O(hndK) \), where \( h \) is the total number of iterations.

a) **THE R-GREEDY Method**

The R-Greedy aims to provide a method for choosing the stable K cluster centers and their initial subspaces for RFPDC. Most existing algorithms only consider the selection of initial K cluster centers by random selection or the Greedy technique. For example, the Greedy technique chooses the first center random, and selects the others such that they are far from one another, and from the first chosen center. It is important to remark that such technique measures the distance between data points by considering all features of the space. In high dimensional spaces, the data are inherently sparse; the distance between every pair of points is almost the same for a wide variety of distance functions.

The subspaces where the initial clusters are situated should be taken into account at the initialization stage. It is because the true distances between data points will be distorted by noisy attributes in the high dimensional data space. Virtually all existing soft subspace clustering algorithms evenly set the feature weights with all entries equal to a constant. The R-Greedy is an extension of the Greedy technique by considering these special characteristics of high dimensional data clustering, as follows.

**Algorithm 2 R-Greedy**

Input: \( x_1, x_2, \ldots, x_n \) and \( K \)

Output: \( V^{(0)}, W^{(0)} \)

begin
1 Initialization
\[ 1.1 \text{Use Eqn. (8) to choose the first cluster center } V; \]
\[ 1.2 \text{Use Eqn. (7) to calculate } w_{ij} \text{ for } j = 1, 2, \ldots, d; \]

2 For \( k = 2 \) to \( K \) do
\[ 2.1 \text{For each point } x, x \notin \{V, V_2, \ldots, V_{k-1}\}; \]
\[ \text{calculate } dist(x) = \min_{j=1, 2, \ldots, k-1} \Sigma_{j=1}^{K} w_{ij}(x_i - v_k)^2 \]
\[ 2.2 \text{Choose the point } x_k \text{ as the } k^{\text{th}} \text{ cluster center using the following rule:} \]
\[ l = \arg \max_{i=1, 2, \ldots, n} dist(x_i) \]
\[ 2.3 \text{Use Eqn. (7) to calculate } w_{ij} \text{ for } j = 1, 2, \ldots, d; \]

3 Output \( V \) as \( V^{(0)} \) and \( W \) as \( W^{(0)} \)
End

There are two major extensions in R-Greedy comparing with the traditional Greedy technique. Since the cluster centers with random selection may result in unstable clustering results, especially on the high dimensional data, a determinable point is selected as the first center at first. Secondly, R-Greedy searches other well-scattered centers using the weighted Euclidean distance function, which calculates the distances between data points in individual subspaces.

The initial subspaces for all chosen cluster centers are computed based on Eqn.(2). In particular,

\[ w_{ij}^{(0)} = (1/X_i^{(0)} + \delta)^2 / (\Sigma_{j=1}^{K} 1/X_j^{(0)} + \delta)^2 \]

©2011 Global Journals Inc. (US)
with

$$X_{ij}^{(0)} = \Sigma_{n=1}^{n} (x_{ij} - \nu_{ij})^2$$

The task of step 1.1 is to choose a point from the dataset those candidates to be one of the centers of the underlying document clusters. $tf$ representation is used for documents, each entry of the data is proportional to the term frequency and has been normalized. Therefore, the length of a vector in such representation is able to measure the relevance degree of the corresponding document with its topic, to some extent. The first cluster center is selected according to the following rule:

$$v_1 = \arg\max_{x_1, x_2, \ldots, x_n} \Sigma_{j=1}^{d} x_{ij}$$

(8)

The time complexity of R-Greedy is $O(ndK)$. More importantly, the R-Greedy can always generate determinable initial conditions for the clustering algorithm. Since fuzzy clustering is generally better than hard clustering at avoiding local minima, using R-Greedy the R-FPC can achieve the robust clustering results with better performance than existing text subspace clustering algorithms.

The proposed system is designed with hierarchical peer-to-peer document clustering (HPEPC). In addition, it uses the distributed K-means clustering algorithm. To perform the document clustering using the semantic analysis mechanism. The fuzzy logic technique is used for the clustering process. The ontology is used for semantic analysis. It also uses Java language and Oracle relational database for application environment.

**Document Preprocess:** The documents are maintained in text file format. The contents of the documents are parsed and converted into the vector space model. The stop word elimination and stemming process are used to reduce the vector size. The system maintains a stop word repository. The stop words in the documents are removed using the repository. The stemming process analyzes the suffix value for the terms. The base term is extracted using the stemming process. The porter-stemming algorithm is used in the system. The document details are updated into the database. The system also updates the term list into the database.

**Term Frequency Estimation:** Term frequency refers to the number of occurrences of a term in a document. Document Frequency (DF) of a term depicts the number of documents in which a term occurs. The IDF can be referred to as the Inverse Document Frequency [9]. Documents can be indexed by taking into consideration Term Frequency (TF) and Inverse Document Frequency (IDF) values. The TF and IDF values are calculated based on term count and document count. The fuzzy clustering scheme is applied on the term collection. The term weights are used for the comparison process. The term cluster requires high vector size for the clustering process. IDF and term weight are estimated with the formula given below.

**Term Frequency = Number of times a particular term occurs in the given document**

**Document Frequency = Number of documents having the target term**

**Inverse Document Frequency = Importance of the target term in the collection**

**Node Management:** Clustering is built up for peer-to-peer networks. Peer node information is generated and distributed to other nodes. Each peer node upholds various document collections. Super nodes hierarchically control the peer nodes. Each super node manages the peer node information. The peer node list demonstrates details on the peer node. Peer node document details are registered in the super node. Users can view the documents in the selected peer. The peer node updates super nodes with all
information. Each super node is hierarchically connected to other similar super node. The main super node treats these super nodes as its peers. The peer nodes can dynamically join and leave network environment. This dynamic join and leave operations created an impact upon the clustering process. Documents are clustered through extracting information from the peer.

Semantic Analysis: Semantic of terms are analyzed to find out the relationship between terms. These term relationships are extracted through the use of multidomain ontology. Concepts, sub concepts and their synonym, meronym and hypernym values are maintained in the ontology. Semantic weights are estimated by taking into consideration term relationship and frequency values.

Clustering process: There are two types of clustering process that are carried out in the proposed system. Local clusters are constructed with the help of the documents in the local node while global clusters are created with documents from all peer nodes. The results of the local clusters are used for generating global clusters. Local optimization based clustering process uses initial centroid estimation in local document collection. On the other hand, global optimization scheme uses centroid estimation from all peer nodes. Term weights and Fuzzy semantic weights are taken into consideration in this clustering process.

VI. **HP2PC Distributed Architecture**

The proposed system is centered on HP2PC. HP2PC is a hierarchically distributed P2P architecture for scalable distributed clustering of horizontally partitioned data. This paper propounds that a scalable distributed clustering system should involve hierarchical distribution because hierarchical processing allows for delegation of responsibility and modularity.

Central to this hierarchical architecture design is the formation of neighborhoods. A neighborhood is a group of peers forming a logical unit of isolation in an unrestricted open P2P network. Peers within a neighborhood can communicate directly but they cannot communicate with peers in other neighborhoods. Each neighborhood is connected to a supernode. Communication between neighborhoods takes place through their respective supernodes. This model reduces flooding usually encountered in large P2P networks. The notion of a neighborhood united 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 other neighborhoods on the same level of hierarchy.
VII. **HP2PC Distributed Clustering Algorithm**

HP2PC is an algorithm introduced by Hammouda and Kamel in [29]. The algorithm stands for a distributed iterative clustering process. It is a centroid-based clustering algorithm, wherein a set of cluster centroids is generated to formulate clustering solution. In HP2PC, each neighborhood converges into a set of centroids describing the data set in that neighborhood. Distributed clustering algorithm within a single neighborhood is similar to K-means algorithm. The final set of centroids of a neighborhood would 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 hierarchy, may converge into another set of centroids.

Once a neighborhood gets converged, the centroids are obtained by the super node of that neighborhood. The super node, in turn, as part of its higher level neighborhood, works together with its peers to form a set of centroids for its neighborhood. This process continues until a set of centroids gets engendered at the root of the hierarchy.

There are certain steps that can enhance centroid estimations that are as follows:

1. Let \( P_i \) be peers and all peers be \( P_1 \) to \( P_n \).
2. Centroid is estimated using a set of peer node.  
3. In each peer node, document cluster centroids are randomly selected and grouped into initial centroid.
4. Super node collects the entire initial centroid document from peer nodes.
5. The centroid for the entire peer is selected from initialized estimated document collection.
6. Documents are selected with help of the distance interval information.
7. Weight for each document is compared with the centroid values of the clusters.
8. Documents are transferred based on weight most similar to the cluster centroid.
9. The centroid for each cluster is reestimated.
10. This process is repeated until the centroids for successive iterations are equal.

The Fuzzy optimized Hierarchical peer-to-peer distributed document clustering also constructed by the above step by term weight is replaced by semantic fuzzy weight for each document.

VIII. **Ontology for Concept Relationships**

Ontology is a storehouse of concept relationships for all domains. The domain related terms are collected and grouped together with their relationships. Varieties of relationships are available to reflect concept relations such as "similar", "kind of" and "type of". The concepts and their associated terms are maintained in a hierarchical manner. Each semantic is analyzed with the ontology collections.

Multi domain ontology is constructed for the peer-to-peer clustering system in three levels. First level is created with the WorldNet dictionary used to fetch term relations. Second level lies with updation of the ontology. Finally, expert knowledge is sought to improve the ontology information constructed. The Protégé 4.1 tool is used to build the ontology depending upon concept, sub concept, terms and their relationship.
IX. **Experiment and Result**

**a) Dataset Description**

Clustering experiments were performed on three different characteristic and sizeable document data sets. Table 1 lists the data sets used for evaluation. IEEE Abstracts, 20NG and RCV1 are the standard text mining data sets handled. IEEE Abstracts were manually collected from IEEE web site. A brief description of each data set is given below.

IEEE Abstracts consist of a collection of 5,000 articles from data mining domain from IEEE Journals. They are from 30 categories, a few among them were data warehousing, databases and so forth and they have rather an unbalanced distribution. They have been used for research in document clustering. The journal abstract page is designed using HTML. HTML pages are downloaded and transformed into text data. Text documents are converted through eliminating the HTML tag elements from web documents. The text contents are maintained in separate text files. The list of journals from IEEE considered for clustering are: Biomedical Engineering, Circuits and Systems, Communications and Computer Graphics and Application.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. of Document</th>
<th>No. of Classes</th>
<th>Minimum Class Size</th>
<th>Maximum Class Size</th>
<th>No. of Unique Terms</th>
<th>Average terms/ Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE Abstracts</td>
<td>5,000</td>
<td>30</td>
<td>47</td>
<td>192</td>
<td>6245</td>
<td>104</td>
</tr>
<tr>
<td>20NG</td>
<td>18,828</td>
<td>20</td>
<td>47</td>
<td>986</td>
<td>18,330</td>
<td>151</td>
</tr>
<tr>
<td>RCV1</td>
<td>23,149</td>
<td>103</td>
<td>12</td>
<td>794</td>
<td>8,244</td>
<td>196</td>
</tr>
</tbody>
</table>

20NG refers to the standard 20-newsgroup data set. It contains 18,828 documents from 20 Usenet newsgroups divided into 20 balanced categories. Each category consists of 1,000 documents assigned to it. The 20NG data sets are available at http://people.csail.mit.edu/jrennie/20Newsgroups/[34].

RCV1 is a subset of 23,149 documents selected from the standard Reuters RCV1 text categorization data set[35], converted from the original Reuters RCV1 data set by Lewis et al. Each document is a news article on certain topics like earnings, commodities, acquisitions, grain, copper, etc. The documents in the RCV1 data set are assigned with multiple labels. In order to properly evaluate the clustering algorithms using single-label validity measures, the labels of the documents are restricted in the data set.

Each datasets individually used in the texting process. The clustering and the performance always are done for each datasets. Finally all the datasets are grouped into single document repository. The performance analysis is carried out on the entire data collection with 47,000 documents. The following analysis describe the performance measures for entire data collection.

**b) Experimental Setup**

A peer-to-peer network environment is constructed with 65 peer nodes. Each peer node is configured with Intel dual core processor with a speed of P3 with 1 GHz and 1 Giga bytes of memory and 512 MB RAM. Windows XP operating system is loaded in the peer nodes. Java language is employed in the system.

Peer-to-peer environment is used for evaluating the HP2PC, Enhanced Hierarchical Peer-to-Peer document Clustering (EHP2PC) and Semantic analysis based EHP2PC algorithms. During the analysis, data are partitioned randomly over all nodes of the network. The number of clusters is specified in such a way that it corresponds to the actual number of classes in each data set. A set of centroids is chosen randomly by each supernode. The centroids are distributed to all nodes in its neighborhood at the beginning of the HP2PC process. Clustering is invoked at level 0 neighborhoods and is propagated to the root of the hierarchy. The global centroid optimization scheme is used in the EHP2PC and SEHP2PC algorithms. Centroid is estimated with the documents that are selected from the peer nodes by distance analysis mechanism. Cluster quality and communication factors are also analyzed.

**c) Evaluation Methods**

F-measure, purity, entropy and separation index are used to evaluate the accuracy of the clustering algorithms. Speed up measure also is used to evaluate the communication criteria for the peer nodes. Network size and height parameters are also analyzed in the system.

i. **F-measure**

The F-measure is a harmonic combination of the precision and recall values used in information retrieval. Each cluster obtained can be considered as
the result of a query, whereas each pre-classified set of documents can be considered as the desired set of documents towards that query. The precision \( P(i, j) \) and recall \( R(i, j) \) of each cluster \( j \) for each class \( i \) is calculated.

If \( n_i \) is the number of members of the class \( i \), \( n_j \) is the number of members of the cluster \( j \), and \( n_{ij} \) is the number of members of the class \( i \) in the cluster \( j \), then \( P(i, j) \) and \( R(i, j) \) can be defined as

\[
P(i, j) = \frac{n_{ij}}{n_j}, \quad (9)
\]

\[
R(i, j) = \frac{n_{ij}}{n_i} \quad (10)
\]

The corresponding F-measure \( F(i, j) \) is defined as

\[
F(i, j) = \frac{2 \cdot P(i, j) \cdot R(i, j)}{P(i, j) + R(i, j)} \quad (11)
\]

Then, the F-measure of the whole clustering result is defined as

\[
F = \sum_i \frac{n_i}{n} \max_j (F(i, j)) \quad (12)
\]

wherein \( n \) is the total number of documents in the data set. In general, the larger the F-measure is, the better the clustering result is [10].

ii. Purity

The purity of a cluster represents the fraction of the cluster corresponding to the largest class of documents assigned to that cluster. Thus, the purity of the cluster \( j \) is defined as

\[
Purity(j) = \frac{1}{n_j} \max_i (n_i) \quad (13)
\]

The overall purity of the clustering result is a weighted sum of the purity values of the clusters as given below:

\[
Purity = \sum_j \frac{n_j}{n} Purity(j) \quad (14)
\]

In general, if the purity value is larger, the clustering result is better [14]

iii. Entropy

Entropy reveals the homogeneity of a set of objects. It is used to indicate the homogeneity of a cluster and thus is referred to as cluster entropy. Lower cluster entropy produces more homogeneous clusters. The entropy of a pre-labeled class of objects can be measured to evince the homogeneity of a class with respect to the generated clusters. The less fragmented a class across clusters, the higher its entropy, and vice versa and thus is referred to as class entropy.

Cluster entropy. For every cluster \( c_j \) in the clustering result \( c \), we compute \( n(l, c_j)/n(c_j) \), the probability that a member of cluster \( c_j \) belongs to class \( l \). The entropy of each cluster \( c_j \) is calculated using the standard formula

\[
E_{c_j} = -\sum_l \frac{n(l, c_j)}{n(c_j)} \log \frac{n(l, c_j)}{n(c_j)} \quad (15)
\]

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of entropies for each cluster is weighted by the size of each cluster:

\[
E_{c} = \sum_{j=1}^{n(c)} \frac{n(c_j)}{n(D)} \times E_{c_j} \quad (16)
\]

Class entropy [18], [21]. A drawback of cluster entropy is that it rewards small clusters. It means that if a class is fragmented across many clusters, it would still get a low entropy value. To encounter this problem, the class entropy can also be calculated.

The entropy of each class \( l_i \) is calculated using

\[
E_{l_i} = -\sum_j \frac{n(l_i, c_j)}{n(l_i)} \log \frac{n(l_i, c_j)}{n(l_i)} \quad (17)
\]

where the sum is taken across all clusters. The total entropy for a set of classes is calculated as the weighted average of the individual class entropies:

\[
E_{l} = \sum_{i=1}^{n(l)} \frac{n(l_i)}{n(D)} \times E_{l_i} \quad (18)
\]

As with cluster entropy, a drawback of class entropy is that if multiple small classes are lumped into one cluster, their class entropy would still be small.

Overall entropy [18], [21]: To avoid the drawbacks of cluster and class entropy, their values can be combined into an overall entropy measure

\[
E_c(\alpha) = \alpha E_c + (1 - \alpha) E_l \quad (19)
\]

In the experiments, \( \alpha \) to 0.5 are set.

The quality of clustering at different levels of the hierarchy is evaluated. At level \( h = 0 \), the quality of clustering for each neighborhood is evaluated with regard to the subset of data in the neighborhood, i.e.

\[
E_{c} = E_{c^r} \big|_{D^r} \quad (20)
\]

where \( c^r \) is the set of clusters obtained for neighborhood \( r \), and \( D^r \) is the union of data sets of all nodes in that neighborhood \( D^r = \bigcup_{q \in r} D_q \).

At level \( h > 0 \), the clustering acquired by a supernode is evaluated with respect to the data subset
of the nodes at the level 0 reachable from the supernode. Thus, evaluation of the clustering acquired at the root node reflects the quality based on the entire data set.

iv. **Separation Index**

Separation Index (SI) is another cluster validity measure that utilizes cluster centroids to measure the distance between clusters as well as distance between points in a cluster and their respective cluster centroid. It is defined as the ratio of average within-cluster variance (cluster scatter) to the square of the minimum pairwise distance between clusters:

\[
SI = \frac{\sum_{v=1}^{N_v} \sum_{x \in v} dist(x, m_v)^2}{\sum_{v=1}^{N_v} \sum_{x \in v} dist(m_v, m_v)^2}^{1/2} \tag{21}
\]

where \(m_v\) is the centroid of cluster \(c_v\), and \(dist_{min}\) is the minimum pairwise distance between cluster centroids. Clustering solutions with more compact clusters and larger separation have lower Separation Index, thus lower values indicate better solutions. This index is more computationally efficient than other validity indices, such as Dunn’s index [11], which is also used to validate clusters that are compact and well separated. In addition, it is less sensitive to noisy data.

v. **Speedup**

Speedup is a measure of the relative increase in speed of one algorithm over the other. For evaluating HP2PC, speedup is calculated as the ratio of time taken in the centralized case (\(T_c\)) to the time taken in the distributed case (\(T_d\)), including communication time, i.e.,

\[
S = \frac{T_c}{T_d} \tag{22}
\]

To take communication time into consideration in the peer networks, the time taken to transmit a message from one node to another on a 100 Mbps link is factored. Thus, the time required to transmit a message of size \(|M|\) bytes is calculated as

\[
T_M = \frac{|M|}{(100,000,000/8)} \text{seconds}
\]

During the communication, each time when a message is sent from (or received by) one node to another, its time is calculated and is added to the total time taken by that node. Since, in a real environment, all nodes on the same level of the hierarchy run in parallel, the total time taken by that level is calculated as the maximum time taken by any node. The time taken by different levels is added to arrive at the global \(T_d\).

d) **Performance Analysis**

Clustering performance is measured under the above-specified peer-to-peer environment with entire collection of three different data sets. Each data set is verified with their actual class information. The experimental analysis is carried out using the HP2PC, EHP2PC and SEHP2PC algorithms. Accuracy level and network complexity factors are considered in the analysis. The F-measure, purity, entropy and separation index metrics are used to verify the cluster accuracy information. The speedup factor is applied in measuring the network complexity levels. The network size is estimated with peer count values. The network height indicates the details of peer connectivity hierarchy. The experimental analysis is performed under different peer node levels. The system is implemented to perform fuzzy text document grouping with the support of semantic analysis.

<table>
<thead>
<tr>
<th>N</th>
<th>HP2PC</th>
<th>EHP2PC</th>
<th>SEHP2PC</th>
<th>FSEHP2PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.496</td>
<td>0.588</td>
<td>0.683</td>
<td>0.847</td>
</tr>
<tr>
<td>5</td>
<td>0.510</td>
<td>0.602</td>
<td>0.701</td>
<td>0.855</td>
</tr>
<tr>
<td>10</td>
<td>0.523</td>
<td>0.617</td>
<td>0.718</td>
<td>0.862</td>
</tr>
<tr>
<td>15</td>
<td>0.539</td>
<td>0.633</td>
<td>0.735</td>
<td>0.872</td>
</tr>
<tr>
<td>20</td>
<td>0.556</td>
<td>0.649</td>
<td>0.754</td>
<td>0.897</td>
</tr>
<tr>
<td>25</td>
<td>0.573</td>
<td>0.665</td>
<td>0.771</td>
<td>0.903</td>
</tr>
<tr>
<td>30</td>
<td>0.590</td>
<td>0.683</td>
<td>0.789</td>
<td>0.908</td>
</tr>
<tr>
<td>35</td>
<td>0.603</td>
<td>0.702</td>
<td>0.807</td>
<td>0.916</td>
</tr>
<tr>
<td>40</td>
<td>0.619</td>
<td>0.722</td>
<td>0.825</td>
<td>0.921</td>
</tr>
<tr>
<td>45</td>
<td>0.636</td>
<td>0.741</td>
<td>0.841</td>
<td>0.939</td>
</tr>
<tr>
<td>50</td>
<td>0.651</td>
<td>0.762</td>
<td>0.859</td>
<td>0.948</td>
</tr>
<tr>
<td>55</td>
<td>0.668</td>
<td>0.783</td>
<td>0.876</td>
<td>0.955</td>
</tr>
<tr>
<td>60</td>
<td>0.686</td>
<td>0.805</td>
<td>0.895</td>
<td>0.969</td>
</tr>
<tr>
<td>65</td>
<td>0.706</td>
<td>0.823</td>
<td>0.913</td>
<td>0.978</td>
</tr>
</tbody>
</table>

The F-measure analysis is carried out for the HP2PC, EHP2PC, SEHP2PC and FSEHP2PC algorithms. The F-measure values are shown in Table 2. Figure 4 shows that the F-measure value of FSEHP2PC method is better than the other methods.
Purity is also used to measure the cluster accuracy levels. The purity is estimated for the HP2PC, EHP2PC, SEHP2PC and FSEHP2PC method as shown in table 3. Figure 5 shows that the purity value of FSEHP2PC algorithm is better than other three methods.

**TABLE 3: Purity analysis of HP2PC, EHP2PC, SEHP2PC and FSEHP2PC Scheme**

<table>
<thead>
<tr>
<th>N(p)</th>
<th>HP2PC</th>
<th>EHP2PC</th>
<th>SEHP2PC</th>
<th>FSEHP2PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.569</td>
<td>0.646</td>
<td>0.768</td>
<td>0.887</td>
</tr>
<tr>
<td>5</td>
<td>0.581</td>
<td>0.661</td>
<td>0.780</td>
<td>0.895</td>
</tr>
<tr>
<td>10</td>
<td>0.594</td>
<td>0.675</td>
<td>0.793</td>
<td>0.891</td>
</tr>
<tr>
<td>15</td>
<td>0.605</td>
<td>0.691</td>
<td>0.804</td>
<td>0.902</td>
</tr>
<tr>
<td>20</td>
<td>0.618</td>
<td>0.707</td>
<td>0.815</td>
<td>0.904</td>
</tr>
<tr>
<td>25</td>
<td>0.632</td>
<td>0.723</td>
<td>0.827</td>
<td>0.909</td>
</tr>
<tr>
<td>30</td>
<td>0.644</td>
<td>0.737</td>
<td>0.840</td>
<td>0.918</td>
</tr>
<tr>
<td>35</td>
<td>0.657</td>
<td>0.750</td>
<td>0.851</td>
<td>0.929</td>
</tr>
<tr>
<td>40</td>
<td>0.669</td>
<td>0.762</td>
<td>0.863</td>
<td>0.935</td>
</tr>
<tr>
<td>45</td>
<td>0.682</td>
<td>0.778</td>
<td>0.875</td>
<td>0.942</td>
</tr>
<tr>
<td>50</td>
<td>0.695</td>
<td>0.792</td>
<td>0.886</td>
<td>0.956</td>
</tr>
<tr>
<td>55</td>
<td>0.709</td>
<td>0.805</td>
<td>0.898</td>
<td>0.963</td>
</tr>
<tr>
<td>60</td>
<td>0.722</td>
<td>0.819</td>
<td>0.908</td>
<td>0.984</td>
</tr>
<tr>
<td>65</td>
<td>0.736</td>
<td>0.832</td>
<td>0.919</td>
<td>0.996</td>
</tr>
</tbody>
</table>
Network size is an important factor in the peer-to-peer environment analysis. The node count is denoted as the network size. The peer nodes are arranged in different hierarchy levels. The hierarchy level information is referred to as the height. All the analyses are carried out with hierarchy level 3. The peer nodes are arranged under the hierarchy tree environment. Data are communicated at the hierarchy levels.

Experiments on different network sizes and heights were performed and their effect on clustering accuracy (Entropy and SI) and speedup over centralized clustering were measured. Table 4 and Table 5 summarize the result of entropy and Separation Index analysis respectively for entire dataset. Network size is analyzed with the cluster accuracy levels. The distributed clustering accuracy stays almost the same even if the network size increases. This is evident through both the Entropy and SI. Figure 6 and 7 show that the Entropy and Separation Index value of FSEHP2PC method is better than other three methods.

**TABLE 4: Entropy Analysis of HP2PC, EHP2PC, SEHP2PC and FSEHP2PC**

<table>
<thead>
<tr>
<th>N(p)</th>
<th>HP2PC</th>
<th>EHP2PC</th>
<th>SEHP2PC</th>
<th>FSEHP2PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.729</td>
<td>0.871</td>
<td>0.995</td>
<td>1.248</td>
</tr>
<tr>
<td>5</td>
<td>0.723</td>
<td>0.859</td>
<td>0.972</td>
<td>1.201</td>
</tr>
<tr>
<td>10</td>
<td>0.696</td>
<td>0.814</td>
<td>0.944</td>
<td>1.186</td>
</tr>
<tr>
<td>15</td>
<td>0.996</td>
<td>1.188</td>
<td>1.352</td>
<td>1.558</td>
</tr>
<tr>
<td>20</td>
<td>1.025</td>
<td>1.217</td>
<td>1.423</td>
<td>1.637</td>
</tr>
<tr>
<td>25</td>
<td>1.026</td>
<td>1.227</td>
<td>1.459</td>
<td>1.679</td>
</tr>
<tr>
<td>30</td>
<td>0.956</td>
<td>1.089</td>
<td>1.236</td>
<td>1.492</td>
</tr>
<tr>
<td>35</td>
<td>0.951</td>
<td>1.062</td>
<td>1.217</td>
<td>1.469</td>
</tr>
<tr>
<td>40</td>
<td>1.049</td>
<td>1.299</td>
<td>1.497</td>
<td>1.702</td>
</tr>
<tr>
<td>45</td>
<td>1.326</td>
<td>1.574</td>
<td>1.802</td>
<td>1.941</td>
</tr>
<tr>
<td>50</td>
<td>1.102</td>
<td>1.339</td>
<td>1.586</td>
<td>1.773</td>
</tr>
<tr>
<td>55</td>
<td>1.472</td>
<td>1.753</td>
<td>2.017</td>
<td>2.187</td>
</tr>
<tr>
<td>60</td>
<td>1.324</td>
<td>1.553</td>
<td>1.779</td>
<td>2.102</td>
</tr>
<tr>
<td>65</td>
<td>1.310</td>
<td>1.523</td>
<td>1.746</td>
<td>2.079</td>
</tr>
</tbody>
</table>
Each node can update its centroids at the end of iteration based on all information received from all other nodes. This process implies that increasing the network size does not affect the accuracy of clustering. The HP2PC algorithm is enhanced with centroid optimization factors. The centroid values are selected from various nodes. Optimal distance based centroid is chosen for the clustering process.

The initial centroid is distributed to all nodes. The peer nodes perform the clustering operation based on common distributed centroid value. During iteration, the centroid is shared by all the nodes. All cluster iterations use similar centroid for document assignment. This initial centroid optimization improves the accuracy levels of the peer-to-peer clustering process. The eHP2PC algorithm is designed with the global centroid optimization scheme. The ontology based semantic analysis is integrated with the EHP2PC algorithm and transformed into SEHP2PC algorithm. Finally Fuzzy technique is applied for optimization of clustering result. Table 4 also shows the accuracy level and speed up analysis of the HP2PC, EHP2PC, SEHP2PC and FSEHP2PC method. The accuracy level is improved under the EHP2PC than the HP2PC based model. FSEHP2PC also improves the accuracy level based on term relationship as shown in figure 8.
Global centroid optimization creates an impact upon the clustering accuracy and inter-cluster distance. The accuracy gets reflected under entropy analysis. Distance information is also considered in the centroid initialization process. The distance between the selected transaction weights are estimated and high distance intervals are considered in the centroid selection process. Cluster distance is determined with respect to the initial centroid selection interval. The inter-cluster distance can be termed as separation index. The separation index values are improved under the EHP2PC algorithm. The clusters are built with high distance values. The SEH2PC algorithm also produces clusters with long distance intervals.
Network size and height based analysis show that, for networks of the same size, larger network heights cause clustering accuracy to drop. It is not surprising that this is the case, since at higher levels metaclustering of lower level centroids is expected to produce some deviation from the true centroids. Since the increase in hierarchy height has the biggest effect on the accuracy of the resulting clustering accuracy, a strategy based on the SI measure can be adopted to select the most appropriate hierarchy for a certain application.

Table 7 summarizes the Accuracy level of HP2PC, EHP2PC, SEHP2PC and FSEHP2PC. Table 7 also shows the F-Measure, purity, Entropy and Separation Index for fuzzy clustering (FC) in peer to peer individual stand alone machine.

<table>
<thead>
<tr>
<th>Technique</th>
<th>F-measure</th>
<th>Purity</th>
<th>Entropy</th>
<th>Separation Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC</td>
<td>0.542</td>
<td>0.608</td>
<td>0.768</td>
<td>0.288</td>
</tr>
<tr>
<td>HP2PC</td>
<td>0.496</td>
<td>0.569</td>
<td>0.729</td>
<td>0.144</td>
</tr>
<tr>
<td>EHP2PC</td>
<td>0.588</td>
<td>0.646</td>
<td>0.871</td>
<td>0.379</td>
</tr>
<tr>
<td>SEHP2PC</td>
<td>0.683</td>
<td>0.768</td>
<td>0.995</td>
<td>0.503</td>
</tr>
<tr>
<td>FSEHP2PC</td>
<td>0.847</td>
<td>0.887</td>
<td>1.248</td>
<td>1.973</td>
</tr>
</tbody>
</table>

Table 7 also shows how fuzzy clustering is applied to clustering on more web and XML documents. The system can be adopted to cluster multilingual documents. Multi document summarization techniques can be embedded with the system to extract cluster summary information.

X. Conclusions and Future Direction

In this paper, describes HP2PC model and SEHP2PC Model. Then the fuzzy weight optimization mechanism can integrated with the SEHP2PC scheme to improve the clustering accuracy levels. The dynamic nodes join and leave operations are also included in the system.

The following aspects are considered in the future development of the system. The Portable Document Format (PDF) and Rich Text Format (RTF) documents can be clustering by the future version of fuzzy document clustering system. Fuzzy can be applied to clustering on more web and XML document. The system can be adopted to cluster multilingual documents. Multi document summarization techniques can be embedded with the system to extract cluster summary information.
References Références Referencias


33. Reuters-21578, Distribution 1.0

This page is intentionally left blank