# A Competent And Empirical Model Of Distributed Clustering

**A.Vasudeva Rao**

*Associate Professor*

*Dept of CSE, Dadi Institute of Engineering Technology, Anakapalli, Visakshpatnam,JNTUK*

**Abstract: -** ***Categorizing the different types of data over network is still an important research issue in the field of distributed clustering. There are different types of data such as news, social networks, and education etc. All this text data available in different resources. In searching process the server have to gather information about the keyword from different resources. Due to more scalability this process leads more burdens to resources. So we introduced a framework that consists of efficient grouping method and efficiently clusters the text in the form of documents. It guarantees that more text documents are to be clustered faster.***

## 1.INTRODUCTION

Information retrieved from a computer which is already stored in that computer. Information is stored in the form of documents. Computer may not be store the information as same in the documents. It may overwrite the data to computer understandable language. The document may contain abstract at the starting of the document or may the words list. It is must and should process to maintain the document in the computer.

In the practical approaches researchers considered that input document must contain the abstract or tile and some text. This takes more time to process and the document is notated with the main class of the document. The documents stored in the computer must maintain index with understandable classes. It is only possible that the document is fully filtered. Such as the removing of the grammar words and the duplicate words and maintain more frequency terms are in the top position.

In the document indexing is the main part to maintain the numbers of tokens such as keywords in a language. This is only possible of clustering means that grouping of the keywords which have same properties. There are so many types of grouping and some of them is serial search. Serial search is defined as the match queries with every document in the computer and group the files to match the query with respect to keyword and that is so called as cluster representative.

Cluster representative will process the input query and perform search for the documents which is matched. Apart from that the documents which are not matched is eliminated from the group.

There are more number of clusters algorithms are there to cluster the documents . The ultimate goal is clustering algorithms' group a set of documetns into subsets clusters.The algorithms' goal is to grpup similar documents and remaining documents are deviate from the clusters.

Classification of a document into a classification slot and to all intents and purposes identifies the document with that slot. Other documents in the slot are treated as identical until they are examined individually. It would appear that documents are grouped because they are in some sense related to each other; but more basically and they are grouped because they are likely to be wanted together and the logical relationship is the means of measuring this likelihood. In this people have achieved the logical organization in two different ways. Initially through direct classification of the documents and next via the intermediate calculation of a measure of closeness between documents. The basic approach has proved theoretically to be intractable so that any experimental test results cannot be considered to be reliable. The next approach to classification is fairly well documented now and there are some forceful arguments recommending it in a particular form. It is this approach which is to be emphasized here.

This process is used for the document matching. It searches for the document in the clusters which is matching to another document and the matching frequency of the documents. Group with high score frequency which is matching is the new document is assigned to that group. It leads to the retrieval process slow .

Document clustering (or Text clustering) is documents and keyword extraction and fast information retrieval . Document clustering is the use of descriptors. They are sets of words such as word bag that explains the contents in the cluster. Clustering of documents considered to be a centralized process which includes web document clustering for search users. It is divided into two types such as online and offline. Online clustering have efficiency problems than offline clustering.

Most of the classifications are based on binary relationships. These relationships of classification method construct the system of clusters. It is explained in different types such as similarity, association and dissimilarity. Abort the dissimilarity it will be defined mathematically later and the other two parameters are means the association will be reserved for the similarity between objects. Similarity measure is designed to find the equity between the keywords and the documents. Possible similar tokens are grouped together.

There are two types of algorithms such as hierarchical based algorithm which calculations are depends upon the links and the averages of the similarity. Aggregation clustering is more compatible to browsing. These two have their limitations in the efficiency. There is another algorithm that is developed using the K-means algorithm and its features. It has more effiency and reduces the computations in the clustering which also gives accurate results.

In the process of searching the user gives a keyword to search and its displays relevant documents. The internal process is find the similarity between or finding the documents from the resources. For finding the similarity we have different types of similarity measures.

Text Clustering methods are divided into three types. They are partitioning clustering, Hierarchal clustering, fuzzy is clustering. In partitioning algorithm, randomly select $k$ objects and define them as $k$ clusters. Then calculate cluster centroids and make clusters as per the centroids. It calculates the similarities between the text and the centroids. It repeats this process until some criteria specified by the user.

Hierarchical algorithms build a cluster hierarchy; clusters are composed of clusters that are composed of clusters. There is a way from single documents up to the whole text set or any part of this complete structure. There are two natural ways of constructing such a hierarchy: bottom-up and top-down. It puts all documents into one cluster until some criteria reached.

In this paper we introduced new process of clustering. In related work section briefly explained about the traditional clustering algorithms. If text data present in single resource, we can cluster easily because we can gather information from centralized system. Our situation is to cluster text data in different resources such as decentralized systems. In this we use normal clustering algorithms we cannot perfectly cluster the text data. So we used clustering algorithm using some properties of the traditional clustering algorithm and it has the capability to use in distributed systems also. Hierarchical techniques produce a nested sequence of partitions, with a single and all inclusive cluster at the top and singleton clusters of individual points at the bottom. In every intermediate level can be viewed as combining two clusters from the next lower level (or splitting a cluster from the next higher level).

## II. RELATED WORK

Hierarchical techniques and partitional clustering techniques are single level division of the data points. If $K$ is the number of clusters given by the user the clustering algorithm finds all $K$ clusters. The traditional hierarchical clustering which divide a cluster to get two clusters or merge two clusters. Hierarchical method used to generate a division of $K$ clusters and the repeat the steps of a partitional scheme can provide a hierarchical clustering.

There are a number of partitional algorithms and only describe the K-means algorithm which is mainly used in clustering. K-means algorithm based on centroid which represent a cluster. K-means use the centroids and which is the mean or median point of a group of points. Centroid is not an actual data point. Centroid is the most important in a cluster. The values for the centroid are the mean of the numerical attributes and the mode of the categorical attributes.

K-means Clustering :

Partitioned clustering method is related with a centroid and every point is input to the cluster with the distance less to the centroid. Cluster number can be specified by the user only.

The basic algorithm is very simple

The basic K-means clustering technique is shown below. We can explain it later in the following sections.

Traditional K-means Algorithm for finding $K$ clusters.

1. Select $K$ points as the initial centroids.
2. Assign all points to the closest centroids.
3. Re-compute the centroids of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change.

Initial centroids are often chosen randomly. The centroid is (typically) the mean of the points in the cluster. Similarity is measured by Euclidean distance or cosine similarity or correlation and K-means will converge for common similarity measures mentioned above. In the first few iterations

Complexity is O (n * K * I * d)

n = number of points, K = number of clusters, I = number of iterations, d = number of attributes

Similarity calculation is the main part in our proposed work. We use cosine similarity; it is explained in our proposed work. It means algorithm the keywords or tokens are to be clustered up to some criteria to be reached. A key limitation of k-means is its cluster model. It based on clusters that are separated in a way that the mean value related towards the center of cluster. The clusters are expected as similar size and the assignment to the nearest cluster center is the correct assignment.

In k-means algorithm more number of keywords present in the document it takes more time to process and also the computational complexity also high.

### *A) Initial features of Clustering*

There are two types of searching such as central servers and flooding-based searching targeted scalability and efficiency of distributed systems. The central servers disqualified with a linear complexity for storage because they concentrate all resources of data and nodes in one single system. Flooding-based method avoids the management of references on other nodes and they face scalability problems in the communication process.

Distributed hash tables are the main tool for maintaining the structure the distributed systems. It maintains the position of the nodes in the communication system and it has the below properties. They are

It maintains the references to the nodes and it has the complexity $O(\log N)$ where $N$ depicts the number of nodes in the channel. For finding the path of nodes and data items into address and routing to a node leads to the data items for which a certain node is responsible. The queries given by the user reaches the resource by small nodes in the network to the target node. Distributing the identifiers of nodes and equally outputs the system and reduce load for retrieving items should be balanced among all nodes.

Not an every node maintains the individual functionality and equally distributed the work of every node. So distributed hash tables are considered to be very robust against random failures and attacks.

A distributed index provides a definitive answer about results. If a data item is stored in the system and the DHT guarantees that the data is found.

The main initial thing is tokenizing the key words in documents. Tokenizing means dividing the keywords, for this we construct DHT (Distributed hash table). It contains keyword and respective location of the keyword in the document. It also contains frequency of the keyword in the documents. In our work we construct DHT for clusters.

DHT provides lookup for the distributed networks by constructing hash tables. By using DHT distributed networks or systems maintains mapping among the nodes in the network. It maintains more number of nodes. It is very useful in constructing large networks.

In the traditional clustering the data points are clustered up to some criteria reached. But our proposed work clustering applied on all data points no point remains. All data points should be placed in clusters.

## III.PROPOSED WORK

For a given number of documents construct distributed hash table. For every document we construct DHT which contains tokens or terms and keys. These tables are referenced for next clustering process.

Second is similarity between the nodes in the network so we use cosine similarity. In this similarity calculation we consider only the similar properties between the edges. The reason of taking cosine similarity measure is explained below.

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1 it is less than 1 for any other angle. It takes magnitude of two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90° have a similarity of 0 and two vectors opposed to have a similarity of -1 and it is independent of their sign. This similarity is particularly used in positive space, where the outcome is neatly bounded in [0,1].

Note that these are only apply for any number of dimensions and their Cosine similarity is most commonly used in high-dimensional positive spaces. In Information Retrieval and text mining and each term is notionally assigned a different dimension and a document is characterized by a vector where the value of each dimension corresponds to the number of times that term appears. Cosine similarity then gives a useful measure of how similar two documents are likely to be in terms of their subject matter. The technique is also used to measure cohesion within clusters in the field of data mining.

*Cosine distance* is a term often used for the complement in positive space, that is: Dc(A,B)=1-Sc(A,B) . It is important to note and that this is not a proper distance metric as it does not have the triangle inequality property. The same ordering and necessary to convert to trigonometric distance (see below.) One of the reasons for the popularity of Cosine similarity is that it is very efficient to evaluate especially for sparse vectors and only the non-zero dimensions need to be considered.

In our work the cosine similarity between documents and cluster centroids and it is defined as

$$\mathrm{Cos}(d,c)=\sum_{t\varepsilon d}\mathrm{TF}(t,d)*\frac{TF(t,c)}{|d|\,|c|}$$

Next Clustering, Consider two nodes have some documents. On these documents we perform Agglromative Hierarchal Clustering algorithm.

In this it follows the following steps.

→Take all keys words such as data points in the document.

→Cluster the points using the similarity measure, All points placed in clusters.

→Then take least distanced cluster and start index from zero. Then merge all points in the clusters and perform clustering process.

→Order top ten clusters

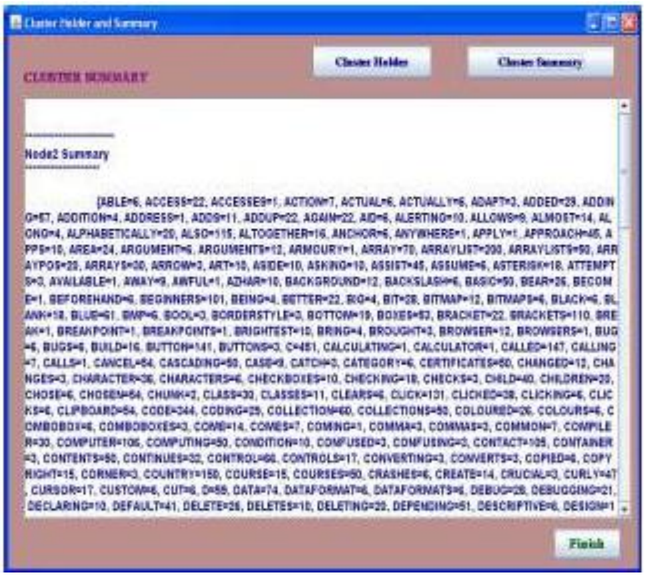
→For every cluster it maintains gist, keywords and the frequency of the keywords of every cluster.

→The cluster which is present in the node that referred as cluster holder.

→If new document appears, it calculates similarity measure with every cluster. The document will place on the highest similarity cluster.

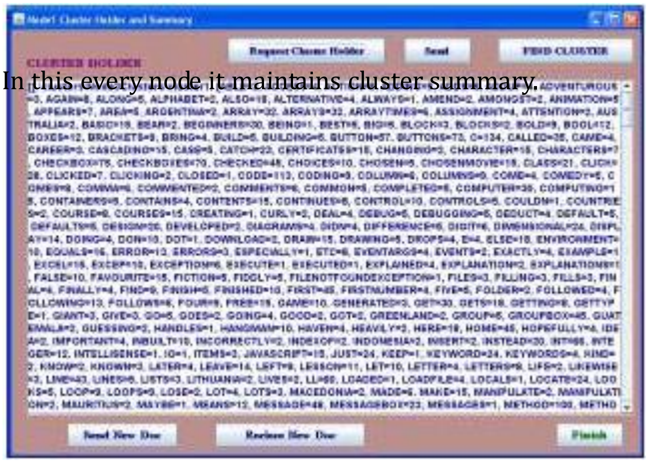
Note that the similarity is compared for new document is with cluster centroids and the new document. The above generated cluster summary is used for calculation of the similarity measure.

The experimental results shown below:



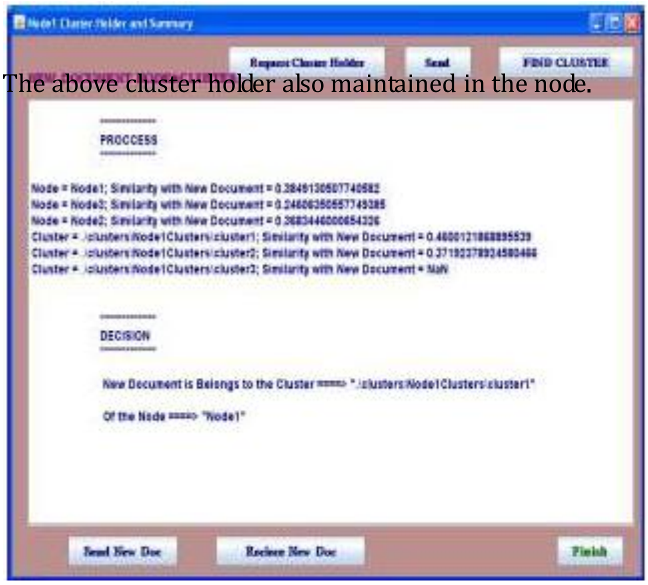
```
---------------
Node2 Summary
---------------

            [ABLE=6, ACCESS=22, ACCESSES=1, ACTION=7, ACTUAL=6, ACTUALLY=6, ADAPT=3, ADDED=29, ADDIN
G=57, ADDITION=4, ADDRESS=1, ADDS=11, ADDUP=22, AGAIN=22, AID=6, ALERTING=10, ALLOWS=9, ALMOST=14, AL
ONG=4, ALPHABETICALLY=20, ALSO=115, ALTOGETHER=16, ANCHOR=6, ANYWHERE=1, APPLY=1, APPROACH=45, A
PPS=10, AREA=24, ARGUMENT=6, ARGUMENTS=12, ARMOURY=1, ARRAY=70, ARRAYLIST=200, ARRAYLISTS=50, ARR
AYPOS=20, ARRAYS=30, ARROW=3, ART=10, ASIDE=10, ASKING=10, ASSIST=45, ASSUME=6, ASTERISK=18, ATTEMPT
S=3, AVAILABLE=1, AWAY=9, AWFUL=1, AZHAR=10, BACKGROUND=12, BACKSLASH=6, BASIC=50, BEAR=26, BECOM
E=1, BEFOREHAND=6, BEGINNERS=101, BEING=4, BETTER=22, BIG=4, BIT=28, BITMAP=12, BITMAPS=6, BLACK=6, BL
ANK=18, BLUE=61, BMP=6, BOOL=3, BORDERSTYLE=3, BOTTOM=19, BOXES=53, BRACKET=22, BRACKETS=110, BRE
AK=1, BREAKPOINT=1, BREAKPOINTS=1, BRIGHTEST=10, BRING=4, BROUGHT=3, BROWSER=12, BROWSERS=1, BUG
=6, BUGS=6, BUILD=16, BUTTON=141, BUTTONS=3, C=451, CALCULATING=1, CALCULATOR=1, CALLED=147, CALLING
=7, CALLS=1, CANCEL=54, CASCADING=50, CASE=9, CATCH=3, CATEGORY=6, CERTIFICATES=50, CHANGED=12, CHA
NGES=3, CHARACTER=36, CHARACTERS=6, CHECKBOXES=10, CHECKING=18, CHECKS=3, CHILD=40, CHILDREN=20,
CHOSE=6, CHOSEN=54, CHUNK=2, CLASS=30, CLASSES=11, CLEARS=6, CLICK=131, CLICKED=38, CLICKING=6, CLIC
KS=6, CLIPBOARD=54, CODE=344, CODING=25, COLLECTION=60, COLLECTIONS=50, COLOURED=26, COLOURS=6, C
OMBOBOX=6, COMBOBOXES=3, COME=14, COMES=7, COMING=1, COMMA=3, COMMAS=3, COMMON=7, COMPILE
R=30, COMPUTER=106, COMPUTING=50, CONDITION=10, CONFUSED=3, CONFUSING=3, CONTACT=105, CONTAINER
=3, CONTENTS=50, CONTINUES=32, CONTROL=66, CONTROLS=17, CONVERTING=3, CONVERTS=3, COPIED=6, COPY
RIGHT=15, CORNER=3, COUNTRY=150, COURSE=15, COURSES=50, CRASHES=6, CREATE=14, CRUCIAL=3, CURLY=47
, CURSOR=17, CUSTOM=6, CUT=6, D=55, DATA=74, DATAFORMAT=6, DATAFORMATS=6, DEBUG=28, DEBUGGING=21,
, DECLARING=10, DEFAULT=41, DELETE=26, DELETES=10, DELETING=20, DEPENDING=51, DESCRIPTIVE=6, DESIGN=1
```

In this every node it maintains cluster summary.



```
[illegible] ADVENTUROUS
=3, AGAIN=8, ALONG=5, ALPHABET=2, ALSO=18, ALTERNATIVE=4, ALWAYS=1, AMEND=2, AMONGST=2, ANIMATION=5
, APPEARS=7, AREA=5, ARGENTINA=2, ARRAY=32, ARRAYS=33, ARRAYTIMES=6, ASSIGNMENT=4, ATTENTION=2, AUS
TRALIA=2, BASIC=19, BEAR=2, BEGINNERS=30, BEING=1, BEST=6, BIG=5, BLOCK=3, BLOCKS=2, BOLD=9, BOOL=12,
BOXES=12, BRACKETS=9, BRING=4, BUILD=5, BUILDING=5, BUTTON=57, BUTTONS=73, C=134, CALLED=35, CAME=4,
CAREER=3, CASCADING=15, CASE=5, CATCH=22, CERTIFICATES=15, CHANGING=3, CHARACTER=15, CHARACTERS=7
, CHECKBOX=75, CHECKBOXES=70, CHECKED=48, CHOICES=10, CHOSEN=5, CHOSENMOVIE=15, CLASS=21, CLICK=
28, CLICKED=7, CLICKING=2, CLOSED=1, CODE=113, CODING=9, COLUMN=6, COLUMNS=9, COMB=4, COMEDY=5, C
OMES=9, COMMA=6, COMMENTED=2, COMMENTS=6, COMMON=5, COMPLETED=5, COMPUTER=35, COMPUTING=1
5, CONTAINERS=5, CONTAINS=4, CONTENTS=15, CONTINUES=6, CONTROL=10, CONTROLS=5, COULDN=1, COUNTRIE
S=2, COURSE=9, COURSES=15, CREATING=1, CURLY=2, DEAL=4, DEBUG=5, DEBUGGING=5, DEDUCT=4, DEFAULT=5,
 DEFAULTS=5, DESIGN=20, DEVELOPED=2, DIAGRAMS=4, DIDN=4, DIFFERENCE=6, DIGIT=6, DIMENSIONAL=24, DISPL
AY=14, DOING=4, DON=10, DOT=1, DOWNLOAD=2, DRAW=15, DRAWING=5, DROPS=4, E=4, ELSE=18, ENVIRONMENT=
10, EQUALS=16, ERROR=13, ERRORS=3, ESPECIALLY=1, ETC=6, EVENTARGS=4, EVENTS=2, EXACTLY=4, EXAMPLE=1
, EXCEL=15, EXCEP=10, EXCEPTION=6, EXECUTE=1, EXECUTED=1, EXPLAINED=4, EXPLANATION=2, EXPLANATIONS=1
, FALSE=10, FAVOURITE=15, FICTION=5, FIDDLY=5, FILENOTFOUNDEXCEPTION=1, FILES=3, FILLING=3, FILLS=3, FIN
AL=4, FINALLY=4, FIND=9, FINISH=5, FINISHED=10, FIRST=45, FIRSTNUMBER=4, FIVE=5, FOLDER=2, FOLLOWED=4, F
OLLOWING=13, FOLLOWS=6, FOUR=9, FREE=15, GAME=10, GENERATED=3, GET=30, GETS=18, GETTING=8, GETTYP
E=1, GIANT=3, GIVE=3, GO=5, GOES=2, GOING=4, GOOD=2, GOT=2, GREENLAND=2, GROUP=5, GROUPBOX=45, GUAT
EMALA=2, GUESSING=2, HANDLES=1, HANGMAN=10, HAVEN=4, HEAVILY=2, HERE=18, HOME=45, HOPEFULLY=4, IDE
A=2, IMPORTANT=4, INBUILT=10, INCORRECTLY=2, INDEXOF=2, INDONESIA=2, INSERT=2, INSTEAD=20, INT=66, INTE
GER=12, INTELLISENSE=1, IO=1, ITEMS=2, JAVASCRIPT=15, JUST=24, KEEP=1, KEYWORD=24, KEYWORDS=4, KIND=
2, KNOW=2, KNOWN=3, LATER=4, LEAVE=14, LEFT=8, LESSON=11, LET=10, LETTER=4, LETTERS=9, LIFE=2, LIKEWISE
=3, LINE=43, LINES=6, LISTS=3, LITHUANIA=2, LIVES=2, LL=60, LOADED=1, LOADFILE=4, LOCALS=1, LOCATE=24, LOO
KS=5, LOOP=9, LOOPS=9, LOSE=2, LOT=4, LOTS=3, MACEDONIA=2, MADE=6, MAKE=15, MANIPULATE=2, MANIPULATI
ON=2, MAURITIUS=2, MAYBE=1, MEANS=12, MESSAGE=48, MESSAGEBOX=23, MESSAGES=1, METHOD=100, METHO
```

The above cluster holder also maintained in the node.



```
-----------
PROCCESS
-----------

Node = Node1; Similarity with New Document = 0.38461305077405B2
Node = Node3; Similarity with New Document = 0.24606350657749385
Node = Node2; Similarity with New Document = 0.36834460000654326
Cluster = .\clusters\Node1Clusters\cluster1; Similarity with New Document = 0.4600121868895539
Cluster = .\clusters\Node1Clusters\cluster2; Similarity with New Document = 0.37192378924580466
Cluster = .\clusters\Node1Clusters\cluster3; Similarity with New Document = NaN

-----------
DECISION
-----------

New Document is Belongs to the Cluster ====> ".\clusters\Node1Clusters\cluster1"

Of the Node ====> "Node1"
```

For new document , the calculations and the assigning is shown above.

## IV.CONCLUSION

In our proposed work we designed a method for clustering of text in distributed systems. For increasing the complexity of calculations our work very useful. In real time applications also it is very helpful. For reducing the resources work and the processing it works efficiently. Compared to traditional process in distribution systems text clustering process faster.

## REFERENCES

[1] Y. Ioannidis, D. Maier, S. Abiteboul, P. Buneman, S. Davidson, E. Fox, A. Halevy, C. Knoblock, F. Rabitti, H. Schek, and G. Weikum, “Digital library information-technology infrastructures,” *Int J Digit Libr*, vol. 5, no. 4, pp. 266 – 274, 2005.

[2] P. Cudr´e-Mauroux, S. Agarwal, and K. Aberer, “Gridvine: An infrastructure for peer information management,” *IEEE Internet Computing*, vol. 11, no. 5, 2007.

[3] J. Lu and J. Callan, “Content-based retrieval in hybrid peer-topeer networks,” in *CIKM*, 2003.

[4] J. Xu and W. B. Croft, “Cluster-based language models for distributed retrieval,” in *SIGIR*, 1999.

[5] O. Papapetrou, W. Siberski, and W. Nejdl, “PCIR: Combining DHTs and peer clusters for efficient full-text P2P indexing,” *Computer Networks*, vol. 54, no. 12, pp. 2019–2040, 2010.

[6] S. Datta, C. R. Giannella, and H. Kargupta, “Approximate distributed K-Means clustering over a peer-to-peer network,” *IEEE TKDE*, vol. 21, no. 10, pp. 1372–1388, 2009.

[7] M. Eisenhardt, W. M¨ uller, and A. Henrich, “Classifying documents by distributed P2P clustering.” in *INFORMATIK*, 2003.

[8] K. M. Hammouda and M. S. Kamel, “Hierarchically distributed peer-to-peer document clustering and cluster summarization,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 681–698, 2009.

[9] H.-C. Hsiao and C.-T. King, “Similarity discovery in structured P2P overlays,” in *ICPP*, 2003.

[10] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in *SIGCOMM*, 2001.

[11] K. Aberer, P. Cudr´e-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, “P-Grid: a selforganizing structured P2P system,” *SIGMOD Record*, vol. 32, no. 3, pp. 29–33, 2003.

[12] A. I. T. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” in *IFIP/ACM Middleware*, Germany, 2001.

[13] C. D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[14] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD Workshop on Text Mining*, 2000.
[15] G. Forman and B. Zhang, "Distributed data clustering can be efficient and exact," *SIGKDD Explor. Newsl.*, vol. 2, no. 2, pp. 34– 38, 2000.
[16] S. Datta, K. Bhaduri, C. Giannella, R. Wolff, and H. Kargupta, "Distributed data mining in peer-to-peer networks," *IEEE Internet Computing*, vol. 10, no. 4, pp. 18–26, 2006.
[17] S. Datta, C. Giannella, and H. Kargupta, "K-Means clustering over a large, dynamic network," in *SDM*, 2006.
[18] G. Koloniari and E. Pitoura, "A recall-based cluster formation game in P2P systems," *PVLDB*, vol. 2, no. 1, pp. 455–466, 2009.
[19] K. M. Hammouda and M. S. Kamel, "Distributed collaborative web document clustering using cluster keyphrase summaries," *Information Fusion*, vol. 9, no. 4, pp. 465–480, 2008.
[20] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer, "Minerva: Collaborative p2p search," in *VLDB*, 2005, pp. 1263– 1266.