INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND APPLICATIONS
VOLUME X, ISSUE IV, APRIL 2016, WWW.IJCEA.COM, ISSN 2321-3469

A STUDY ONSEED POINT SELECTION METHODS FOR TEXT DOCUMENT CLUSTERING USING K-MEANS

Y. Sri Lalitha
Associate Professor , Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

ABSTRACT:

The steady and amazing progress of storage media has given a great boost to the database and information technologies to form huge repositories of structured (databases) and unstructured (text) data, easily available at a mouse click. Discovering valuable information, hidden in these repositories is not a trivial task. Partitional Clustering algorithms exhibit best performance in high dimensional data and by nature Text documents are sparse and are in high dimensional. But, this technique suffers from two drawbacks. *Converges to local optimum solution, clustering results are sensitive to Seed document selection and a Need to Indicate Number of partitions prior to clustering process.* This work addresses the problem of determining the best seed point for efficient clustering. This study implemented sequence, random buckshot methods and the proposed *rank based seed selection method* to study the effect of clustering quality and accuracy.

Keywords : Clustering, Seed Point Selection, Initial Centroid Selection

[1] INTRODUCTION

The evolving features of hardware with storage and processing power has given a great boost to form huge repositories of structured (databases) and unstructured (text) data, and available at a mouse click. Discovering valuable information, hidden in these repositories is not a trivial task. Organizing and analyzing the information by manual means is not possible. Hence techniques

from the field of data mining are employed in organization, analysis, and transformation of data into required knowledge. The focus of this work is Document Clustering. Document Clustering is the process of building meaningful groups of related documents. Increased interest in developing methods that can help users to effectively navigate, summarize and organize this information with the ultimate, goal of helping them to find the relevant documents, has attracted researchers. The ever increasing importance of Document Clustering and the expanded range of its applications lead to the development of novel and optimized algorithms. One such work in Partitional Clustering optimized process.

In section 2 a brief overview of existing methods and the problem statement, in section 3 discusses the document clustering process, section 4 deals with the methodology of the proposed work, in section 5 results of the work will be analysed and section 6 provides conclusions and future work.

[2] RELATED WORKS

The task of organizing and categorizing to the diverse need of the user by manual means is a complicated job, hence a machine learning techniques named Document Clustering is very useful. Document clustering is broadly categorized as Hierarchical and Partitionaltechniques.

Hierarchical Clustering : Hierarchical Clustering techniques are widely studied in these works [4-8,13, 15, 19]. These techniques are of two types, agglomerative type and divisive type clustering. ***Agglomerative*** *:*Proceeds with one text document in one cluster and in every iteration, it combines the clusters with high similarity. ***Divisive****:*Proceeds with Single set, initially containing whole text documents, in each iteration, it partitions the cluster into two and repeats the process until each cluster contains one document. Hierarchical clustering produces nested partitions, with a document dataset at the beginning of hierarchy, called the root, and single document partitions at the end of hierarchy called leaves. Each intermediate hierarchy called non-leaf partition is treated as merging of two partitions from the immediate lower hierarchy or partitioning an immediate higher hierarchy into two sub-hierarchies. The results of this type of clustering are graphically presented in a tree like structure, called dendogram. By disconnecting the dendogram at different levels one can obtain better clustering results. Thus, dendogram provides valuable descriptions and visualization of data clusters and document categorization, particularly when relations of hierarchy exist in the data. This is a widely used approach in IRS. Agglomerative hierarchical clustering uses *Single linkage, Group-Average Linkage, Complete Linkage and Un-weighted Pair Group Method with Arithmetic Mean linkage (UPGMA).* UPGMA is a more accurate clustering mechanism for text documents. Hierarchical clustering techniques are considered to be more reliable they compare all pairs of documents, but are inefficient due to their time complexity of $O(n^2)$.

Partitional Clustering : Partitional Clustering forms flat partitions, or in other words, single level partitions of documents and is applicable in the datasets where inherent hierarchy is not needed. If number of clusters to form is K, Partitional approach finds all the required partitions (*K)* at a time. Various works on Partitional Clustering can be seen in [1, 2, 10, 11-12, 16-19, 30-32] these references. In contrast, with each iteration Hierarchical clustering splits or merge partitions based on the type of approach chosen, divisive or agglomerative. Hierarchical clustering, forms flat sets of K partitions by disconnecting the dendogram and similarily,

Repeated application of Partitional clustering derives Hierarchical clustering. The most widely used methods of *Partitional Clustering* are kMedoid ,kMeans[21].It is known that document clustering suffers from curse of high dimensionality and partitional clustering is best suitable technique in high dimensional data, hence variant of K-means are widely applied to Document clustering. K-means uses centroid to partition documents, centroid is a representative document of a cluster which is a mean or median of a set of documents.

ProblemStatement: Partitional Clustering algorithmskMeans exhibit best performance in high dimensional data and by nature Text documents are sparse and are in high dimensional. But, this technique suffers from two drawbacks. *Converges to local optimum solution, clustering results are sensitive to Seed document selection and there is a Need to Indicate Number of partitions prior to clustering process*

The clustering quality and performance in terms of convergence varies with the selection of initial centroids. The proposed work studies the problem of determining the best seed point for quick convergence and effective clustering. This study implemented sequence, random buckshot methods and the proposed *rank based seed selection method* to study the effect of clustering quality and accuracy.

[3] DOCUMENT CLUSTERING PROCESS

The technique of organizing huge document collections into subsets of meaningful groups with very little prior knowledge is called Document Clustering. Typical Document Clustering contains the following Stages[21].

Figure 1 : Document Clustering Process

A. Document Dataset :Collected Datasets from standard repositories Reuters-21578, 20 Newsgroups and Classic are primary benchmark datasets. Reuters-21578 is a text categorization test collection, and is a major resource for research in information retrieval, machine learning, and other corpus-based research.[88] and Classic another dataset [89], this dataset consists of 4 different document collections: CACM, CISI, CRAN, and MED. These collections can be downloaded as one file per collection.This dataset is usually referred to as Classic3 dataset (CISI, CRAN and MED only), and sometimes referred to as Classic4 dataset.

B. Pre-processing Documents : Pre-processing consists of steps that take as input a plain text document and output a set of tokens (which can be single terms or n-grams) to be included in the vector model. These steps typically consist of: filtering, tokenization, stemming, Stopword removal and Pruning. *Filtering :* The process of removing special characters and punctuation which is more critical in the case of formatted documents, such as web pages, where formatting tags can either be discarded or identified and their constituent terms attributed different weights. *Tokenization:* Splits sentences into individual tokens. *Stemming :*The process of reducing words to their base form, or stem. For example, the words “connected,” “connection”, “connections” are all reduced to the stem “connect.” Porter’s

algorithm is the de facto standard stemming algorithm.*Stopword* removal: The most common words that will not hold any discriminative power in clustering such as "the, of, from etc" are discarded. *Pruning:* Removes words that appear with very low frequency throughout the corpus. The underlying assumption is that these words, even if they had any discriminating power, would form too small clusters to be useful. A pre-specified threshold is typically used, e.g. a small fraction of the number of words in the corpus. Sometimes words which occur too frequently (e.g. in 40% or more of the documents) are also removed.

C. Feature Representation : In order to reduce the complexity of the documents and make them easier to handle, the document have to be transformed from the full text version to a document vector which describes the contents of the document. In this work Vector space model also known as bag of words is used to represent features.In the vector space model of IR, documents are represented as vectors of features representing the terms that occur within the collection. Vector Space Model (VSM) represents a document as a vector of terms in which each dimension corresponds to a term (or a phrase). An entry of a vector is non-zero if the corresponding term (or phrase) occurs in the document.In this model, each document, d, is considered to be a vector, d, in the term-space (set of document "words"). In its simplest form, each document is represented by the (TF) vector, $d_{tf} = (tf_1 , tf2 , \ldots, tf_n)$,where tf_i is the frequency of the i^{th}term in the document. In addition, we use the version of this model that weights each term based on its inverse document frequency (IDF) in the document collection which discounts frequent words with small weight,as little discriminating power. Finally, in order to account for documents of different lengths, each document vector is normalized so that it is of unit length. Three vector model representations are Boolean vector space model, Inverse document frequency vector space model and Frequency count vector space model.*Boolean Vector Space Model :*In this Model the Documents and the terms are represented in dimensions in that if term occur in document then that one represent by one otherwise zero. Only the presence (1) or absence (0) of a term is included in the vector space model.*Term Frequency Vector Space Model :* Term frequency we show frequency by how many times the words occur in the document. The term frequency tft,d of term t in document d is defined as the number of times that t occurs in d.Generally, for a document d and a term t, the weight of t in d is given as:W (d, t) = TF (d, t). *Inverse Document Frequency Vector Space Model :*If we are interested in the frequency of a term in the set of documents then we use Inverse Document Frequency.IDF meaning is importance of each term inversely proportional to the number of documents that contain that term. This is commonly find out by multiplying the frequency of each term i by log(n/dfi). Where n is the total number of documents in the collection, and dfi is the number of documents that contain term i (i.e., document frequency).
Thus, the tf–idf representation of the document d is: dtf-idf=[tf1 log(n/df1),tf2 log(n/df2), . . . , tfD log(n/dfD)].

D. Document Similarity Measures : Key input to Clustering is Similarity or Distance measure. Similarity indicates the strength of relatedness between two documents, while distance is a measure of divergence between two documents. Although Term document matrix can be used to cluster documents, often Similarity Matrix SM is employed. SM is an N by N matrix, with N indicating the dataset size and contains pair-wise similarities of documents

under consideration. The clustering methods employ similarity values in two measures namely comparative and quantitative measures. In comparative measure the two data items say X and Y are compared with respect to Z to determine which of the two items are similar to Z. In quantitative measure, the data items are considered similar based on a threshold specified [22, 23]. For simple multidimensional data Euclidean or Minkowki measures are employed. As the dimensions increases this simple measure may not be the right measure to find the similarity between documents. Variety of similarity/distance measures were proposed in literature a detail view of it can be obtained in [24 – 28]. Cosine measure is most widely used measure, in the context of text documents. The correlation between the terms of the documents represented in document vector gives the similarity of two documents. The association of two documents is measured using cosine angle of two document vectors. Documents are said similar when this cosine value is maximum.

$$Sim_{Cos}\left(Doc_i, Doc_j\right) = \frac{\sum_{t=1}^{T}\left(Doc_{t,i} \times Doc_{t,j}\right)}{\sqrt{\left(\sum_{t=1}^{T}\left(Doc_{t,i}\right)^2 \times \sum_{t=1}^{T}\left(Doc_{t,i}\right)^2\right)}}$$ **Cosine Similarity**

where $Doc_{i,t}$ is the weight of term t of i^{th} document vector. When the two documents are exactly same the cosine value is 1, whereas, if there is no term in common then, their document vectors are orthogonal and consequently the cosine value is zero.

E. Document Clustering : Partitional clustering algorithms compute a k -way clustering of a set of documents either directly or via a sequence of repeated bisections. A direct k -way clustering is commonly computed as follows. Initially, a set of k documents is selected from the collection to act as the seeds of the k clusters. Then, for each document, its similarity to these k seeds is computed, and it is assigned to the cluster corresponding to its most similar seed. This forms the initial k -way clustering. This clustering is then repeatedly refined so that it optimizes the desired clustering criterion function. The process is repeated till convergence. kMeans algorithm for Document Clustering is given below.

Algorithm :kMeans

1. Initially select k documents as k centroids.
2. Allot documents to centroids of highest similarity.
3. Recalculate cluster centers formed in step 2
4. Repeat Step 2 and 3 until the difference in results of previous and current iteration is nil.

kMeans takes extremely small number of iterations to converge. Observations from [3, 5], suggests for an effective clustering lesser than 5 iterations are sufficed [20]. It is efficient and scalable, for its linear time complexity. The defect of this approach is it is sensitive to the initial seed documents considered and requires establishing the number of partitions ‘K’ much prior to the clustering process. Incorrect seed selection or incorrect K value may lead to poor clustering results.

F. Cluster Quality Evaluation: For clustering evaluation, two measures of cluster “goodness” or quality are used, internal and external. One type of measure allows us to compare different sets of clusters without reference to external knowledge and is called an internal

quality measure, where the"overall similarity" based on the pair wise similarity of documents in a cluster is used. The other type of measures evaluates how well the clustering is working by comparing the groups produced by the clustering techniques to known classes. This type of measure is called an external quality measure, where entropy and f-measure are the measures to calculate cluster accuracy.

[4] PROPOSED WORK

The k-means algorithm start with initial cluster centroids, and documents are assigned to the clusters iteratively in order to minimize or maximize the value of the global criterion function. It is known that the clustering algorithms based on this kind of iterative process are computationally efficient but often converge to local minima or maxima of the global criterion function. There is no guarantee that those algorithms will reach a global optimization. Since different sets of initial cluster centroids can lead to different final clustering results, starting with a good set of initial cluster centroids is one way to overcome this problem. In this work a comparison of different initial cluster centroids detection methods are studied. Buckshot, Fractionation, Sequence Seed Points, Random seed points and the Rank Based methods are analysed.

i. **Sequential selection** of seed points the K documents are picked up consecutively and are considered as K seeds. In general First K documents or K documents from a specified document are selected as Initial centres and starts clustering. This approach is very simple to implement but takes more time in forming the final clusters as the number of iterations will be more to converge. Since, there is no guarantee that most unrelated documents are selected as seeds, with this approach and often takes more time to accomplish the Clustering process.
ii. **Random approach** of initial centres, a random function is used to select K initial documents for K clusters. In general, random initial centres like greedy design finds the near optimal solution rather than optimal global solution. This approach starts with initial random documents and then searches the neighborhood of the initial documents for a better solution. Though this approach yields better results but with each run there is a possibility of deriving different results owing to random function. Random function is used in this process, so one cannot say that the initial documents chosen are from different categories of the dataset and converges quickly. But most often this method is used and in many cases it performed better than sequential approach.
iii. **Buckshot** selects a random sample of size $\sqrt{(kn)}$ documents and apply the clustering process. The centres of the clusters formed with the sample are considered to be the k initial centres of the whole document dataset. This algorithm runs in time O(KN). Since random sampling is employed, the algorithm doesn't yield the same results on the same corpus when repeated runs take place [10].
iv. **Fractionation** divides the dataset into buckets of same size and to each of these buckets, the agglomerative clustering algorithm is applied separately. Then the clusters are considered as individual documents and entire process is applied repeatedly till the required number of partitions K, are formed.Centroids of the resulting K clusters are said to be the initial centers of the dataset [10].

v. **Rank based** initial centers uses neighbor and link measure to determine the most unrelated initial centers. In this approach the similarity measure with neighbors is used to resolve the most unrelated documents. Ranks are assigned to the similarity and links. Rank similarity is set to zero for the least similarity value, similarly, Rank link is assigned zero for least number of neighbors. Further these rank similarity and rank link are summed to get the rank of initial center candidates. It considers the candidates with highest rank (least rank value) as initial centers of the dataset. The overhead of this approach is that it has to rank by sorting K+1 entries three times, for ranking similarity value, by ranking number of neighbors and ranking the summation of rank similarity and number of neighbors ranks to determine the initial centers.

For example, let's consider a data set S containing 6 documents, {d1, d2, d3, d4, d5, d6} whose neighbour matrix is as shown In below figure. When θ=0.3; k=3 and nplus=1; Sm has four documents : Sm={d4, d1, d2, d3}. Next, we obtain the cosine and link values between every pair of documents in Sm, and then rank the document pairs in ascending order of their cosine and link values, respectively. For a pair of documents di and dj, let's define rankcos(di,dj) be its rank based on the cosine value, ranklink(di,dj) be its rank based on the link value, and rank(di,dj) be the sum of rankcos(di,dj) and ranklink(di,dj). For both rankcos(di,dj) and ranklink(di,dj), a smaller value represents a higher rank, and 0 corresponds to the highest rank.

	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	0	1	0	0
d_2	1	1	0	1	0	0
d_3	0	0	1	1	1	0
d_4	1	1	1	1	0	0
d_5	0	0	1	0	1	0
d_6	0	0	0	0	0	1

Figure 2 : Neighbor matrix with threshold 0.3

Similarity measurement between initial centroid candidates.

d_i, d_j	cos	$rank_{cos}$	link	$rank_{link}$	$rank_{d_i,d_j}$
d_1, d_2	0.35	2	3	3	5
d_1, d_3	0.10	1	1	0	1
d_1, d_4	0.40	3	3	3	6
d_2, d_3	0	0	1	0	0
d_2, d_4	0.50	4	3	3	7
d_3, d_4	0.60	5	2	2	7

Figure 3 : Similarity measurement between initial centroids candidates

Rank values of the candidate sets of initial centroids.

com_k	${}_4C_2$ pairs of centroid candidates	$rank_{com_k}$
$\{d_1, d_2, d_3\}$	$\{d_1, d_2\}, \{d_1, d_3\}, \{d_2, d_3\}$	6
$\{d_1, d_2, d_4\}$	$\{d_1, d_2\}, \{d_1, d_4\}, \{d_2, d_4\}$	18
$\{d_1, d_3, d_4\}$	$\{d_1, d_3\}, \{d_1, d_4\}, \{d_3, d_4\}$	14
$\{d_2, d_3, d_4\}$	$\{d_2, d_3\}, \{d_2, d_4\}, \{d_3, d_4\}$	14

Figure 4 : Rank values of the candidate sets of initial centroids

As a result, a smaller rank(di,dj) value also represents a higher rank. The ranks of document pairs are Initial centroids better be well separated from each other in order to represent the whole data set. Thus, the document pairs with high ranks could be considered as good initial centroid candidates. For the selection of k initial centroids out of m candidates, there

are mCk possible combinations. Each combination is a k-subset of Sm, and we calculate the rank value of each combination comk as:

$$rank_{com_k} = \sum rank_{d_i,d_j}, \quad \text{for } d_i \in com_k \text{ and } d_j \in com_k$$

That means, the rank value of a combination is the sum of the rank values of the k C2pairs of initial centroid candidate doc-uments in the combination. In this example, there are 4 combinations available, and their rank values are shown in Figure 3.3.

Then, we choose the combination with the highest rank (i.e., the smallest rank value) as the set of initial centroids for the k-means algorithm. In this example, {d1,d2,d3} is chosen since its rank value is the smallest among four different combinations. The documents in this combination are considered to be well separated from each other, while each of the m is close enough to a group of documents, so they can serve as the initial centroids of the k-means algorithm.

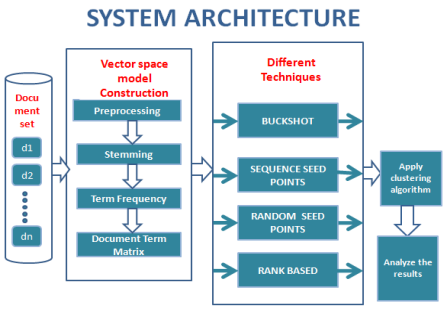

Figure 5 : System Architecture

[5] RESULTS

The implementation of this work is in Java. Figure 6 depicts the interface to the system. The user is allowed to provide the dataset, and select one of the initial centroid selection method. After pre-processing to find the root word stemming option is provided.

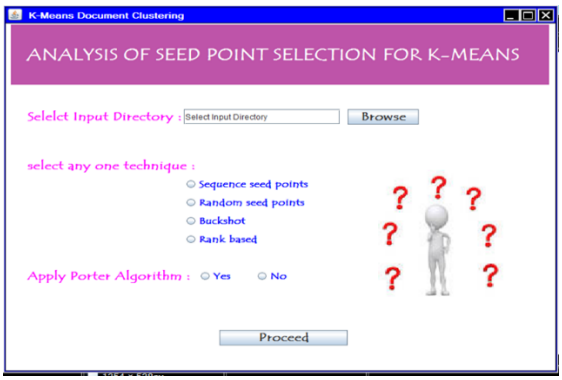

Figure 6 : Interface to Seed Point Selection method

The figure below shows the cluster accuracy obtained by entropy measure for sequence based seed selection

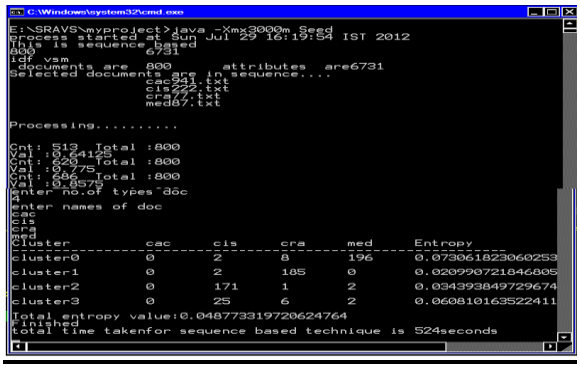

Figure 7: Seed selection based on Sequence method

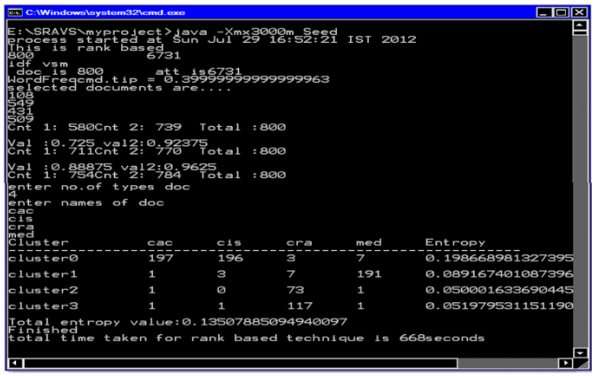

Figure 8 : Rank based approach of seed selection

Figure 9: Depicts the output directory of the resulting clustering.

Table 1 represents the cluster accuracy with initial centroids and simple kmeans algorithm. It can be observed that random and rank based methods exhibited consistent performance, but

random methods output is not the same with every run, it varies as it selects random centroids that differs from previous runs. It can also be observed that classic dataset is depicting low accuracy. The reason behind is we have less variation between different categories, as all documents content is on computer science related topic

	Sequencing	Random	Buckshot	Fractionation	Rank
Reu	0.478	0.469	0.438	0.428	0.479
DT	0.36125	0.281	0.266	0.321	0.331
Classic	0.048	0.107	0.092	0.093	0.135

Table 1 : Cluster Accuracy for different initial centroid methods

[6] CONCLUSION

In our experiments with Classic, Reuters&Dt data sets, the observation is sequence method the quality of clusters formed are good but on large data sets it may not be guaranteed and consumes more time. In literature this method is widely used as it is simple to code. The buckshot method takes less time and the quality of clusters is considerable. The results may vary with the sample selected. We have observed that Random takes less time but not consistent. The results vary with each run. Using Rank based method we are selecting most unrelated seeds, that leads to high quality clusters formation. But it takes more time. Hence we conclude that if quality of the clusters is the criteria, Rank & Sequence methods are better choice. In our future work we will address automatic k find method for a given dataset.

References

[1] Huang A., ‘Similarity Measures for Text Document Clustering”, Proceedings of New Zealand Computer Science Research Student Conference, 2008.

[2] Steinbach M., KarypisG. and Kumar V.A. “Comparison of Document Clustering Techniques”, in KDD Workshop on Text Mining, 2000.

[3] Schutze H. and Silverstein C., “Projections for Efficient Document Clustering”, ACM SIGIR Conference, 1997.

[4] Beil F., Ester M., and Xu X., “Frequent Term-Based Text Clustering”, Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 436–442, 2002.

[5] Cutting D.R., Karger J.O., Pedersen and Tukey J.W., “Scatter/Gather: A Cluster-Based Approach Browsing Large Document Collections”, Proceedings of ACM, SIGIR, pp. 318–329, 1992.

[6] Li A.Y. and Chung S.M., “Parallel Bisecting kMeans with Prediction Clustering Algorithm”, The Journal of Supercomputing, 39(1), pp. 19–37, 2007.

[7] Zamir O., et.al, “Fast and Intuitive Clustering of Web Documents”, KDD ’97, pp. 287-290, 1997.

[8] Larsen B. and Aone C., "Fast and Effective Text Mining using Linear-time Document Clustering", Proceedings of 5th ACM SIGKDD., International Conference, pp.125-149, 1999.

[9] Koller D. and Sahami M., "Hierarchically Classifying Documents using Very Few Words", Proceedings of 14th International Conference Machine Learning, pp. 170-178, 1997.

[10] Arthur D. and Vassilvitskii S. "kMeans++ Advantage of Careful Seeding", Symposium on Discrete Algorithms, 2007.

[11] Lloyd SP, "Least squares quantization in PCM", IEEE Transactions of Information Theory 28(2) pp. 129–137, 1982
.
[12] Raymond Ng. Han J., "An Efficient and Effective Clustering Methods for Spatial Data Mining", Proceedings of 20th International Conference Very Large Databases VLDB, pp. 144–155, 1994.

[13] Kim H. and Lee S., "An Intelligent Information System for Organizing Online Text Documents", Knowledge Information Systems 6, pp. 125-149, 2004.

[14] Zhao Y. and Karypis G., "Evaluation of Hierarchical Clustering Algorithms for Document Datasets", Proceedings of 7th International Conference on Information and Knowledge Management, pp. 515–524, 2002.

[15] MacQueen J., "Some Method for Classification and Analysis of Multivariate Observation", Proceedings 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297, 1967.

[16] Banerjee A., Merugu S. I. and Ghosh Jand, "Clustering with Bregaman Divergences", , Proceedings of 4th SIAM International Conference Data Mining (SDM), pp. 234–245, 2004.

[17] Jin R, Goswami A. and Agrawal G., "Fast and Exact Out-of-core and Distributed kMeans Clustering", Knowledge Information Systems, 10(1), pp. 17–40, 2006.

[18] Kaufman L. and Rousseeuw P., "Finding Groups in Data: An Introduction to Cluster Analysis", John Wiley and Sons, NY, 1990.

[19]Peng X. and Choi B., "Document Classifications Based On Word Semantic Hierarchies", Proceedings of International Conference on Artificial Intelligence and Application, pp. 362–367, 2005.

[20] Charu C. Aggarwal and ChengXiangZhai, "A Survey of Text Clustering Algorithms" www.charaggarwal.net, pp. 77-128, 2012.

[21] Y. Sri Lalitha and A. Govardhan, "Semantic Framework for Text Clustering with Neighbors", ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India - Volume II, Advances in Intelligent Systems and Computing Volume 249 © Springer International Publishing Switzerland, pp. 261-271, 2014.

[22] Guha S., Rastogi R., and Shim K., "CURE: An Efficient Clustering Algorithm for Large Databases", ACM SIGMOD Conference, 1998.

[23] Guha S., Rastogi R. and Shim K., "ROCK : a Robust Clustering Algorithm for Categorical Attributes", Information Systems, 25(5), pp. 345–366, 2000.

[24] Strehl. et.al."Impact of Similarity Measures on Web-Page Clustering". Workshop on AI for Web Search, 2000.

[25] ZhaoY., Karypis G., "Criterion Functions for Document Clustering: Experiment Analysis", Technical Report, Department of Computer Science, Minneapolis, pp. 1-49pp., 2001.

[26] Kogan J., "Data Driven Similarity Measures for kMeans Like Clustering Algorithms", Department of Mathematics and Statistics, Baltimore, 2003.

[27] Zhao Y. and Karypis G., "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering", Machine Learning, pp. 55, 311–331, 2004.

[28] Huang A., 'Similarity Measures for Text Document Clustering", Proceedings of New Zealand Computer Science Research Student Conference, 2008.

[29]http://www.daviddlewis.com/resources/testcollections/reuters21578/

[30]ftp://ftp.cs.cornell.edu/pub/smart/ (1.5MB RAR file)

[30] SK Althaf Hussain Basha, A.Govardhan, "MICR: Multiple Instance Cluster Regression for Student Academic Performance in Higher Education", International Journal of Computer Applications(IJCA), Volume 14–No.4,2011,pp.23-29, ISSN: 0975-8887

[31] SkAlthaf Hussain Basha, Ch. Prakash, D. Mounika, G. Maheetha, "An Approach for Multi Instance Clustering of Student Academic Performance in Education Domain", IIJDWM Journal, Volume 3,Issue 1,pp.1-9,Feb.2013,ISSN: 2249-7161.

[32] SK Althaf Hussain Basha, T Naveen Kumar , V. Anand , Donapati Srikanth, "Categorization of Academic Student Performance using Hybrid Techniques" International Conference on Advanced Computing Methodologies (ICACM-2013), Hyderabad, pp.325- 330,2013.