ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 7, Issue 6, June 2018

PRIME CLUSTERING USING HYBRID FUZZY-GA-DBSCAN IN TEXT DATA MINING

[1]Geeta

Geet027@gmail.com

Student of IEC collage of engg. Greater Noida(UP)

[2]Ashish Kr. Chaturvadi

Ashishchakarvati.cs@ieccollage.com

Department of computer science & Technology

Abstract

As the discovery of information from text data becomes more and more important there is a necessity to develop clustering algorithms designed for such a task. One of the most, successful approaches to clustering is the density based methods. However due to the very high dimensionality of the data, these algorithms are not directly applicable. In this paper we demonstrate the need to suitably exploit the already developed feature reduction techniques, in order to maximize the clustering performance of density based methods.

Keywords: Clustering, Text Mining, GA, SVM, K-mean, C-mean and DB-scan.

1. Introduction

Cluster analysis as an important means of text information mining, is widely used in log analysis, statistics of public opinion and other areas .In recent years, the means of text and word frequency word bag clustering emerge in endlessly, these algorithms extracted a lot of valuable information available from the vast amounts of data, creating a huge amount of wealth. However, with the rapid development of computer and Internet industry, many companies store data reaches research level, such background puts forward higher requirements for data mining algorithms. Some traditional text clustering methods cannot adapt to the massive and high dimension data mining tasks, because of the bottleneck of performance, so it is necessary to design some efficient and agile clustering methods[1]. Experts at home and abroad, according to the characteristic of text mining, many classical clustering scheme applied to the text field, among them, the k-means algorithm proposed by accuracy provides an efficient random clustering, but the method of obtaining the center of mass in iteration makes the clustering result very vulnerable to outliers. And The DBSCAN algorithm proposed by Arlia Massimo and Domenica accurately describes the density of clustering objects, and excludes the interference of noise points, but its O() time complexity often makes it helpless in the face of massive data clustering task[2]. In this paper, we combine the advantages of these two clustering schemes, and use the concept of relative distance to effectively reduce the time complexity of the distance calculation[3]. At the same time, this paper introduces a heuristic rule, which greatly reduces the computation of the distance of the whole object. This algorithm can not only efficiently get the clustering result in a short time, but also can ensure the correctness of the conclusion. Under the conditions of meeting the user's requirements, minimizing the cost of distance computation and the effect of noise points[4].

1.1. Types of Clustering

Broadly speaking, clustering can be divided into two subgroups:

- **Hard Clustering:** In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering**: In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each costumer is assigned a probability to be in either of 10 clusters of the retail store.

1.2. Text Categorization

Text categorization aims in classifying documents into predetermined fixed categories [12]. Transforming documents into an appropriate representation for the learning algorithm and the classification task is the first step in text categorization. Each distinct word w_i in documents which occurs for certain number of times is corresponded to a feature. The word considered as features if it appears in the training set at least 3 times and it is not stop-word (like "and", "or", etc.). This model of representation leads to thousands of dimension features spaces which needs feature subset selection to improve generalization accuracy and to avoid over fitting. Text classification has 5 properties [1]. Firstly, it has high dimensional feature space. If each word in the training documents considered as feature space, then there will be more than 50,000 attributes in a few thousand training example. The second property is that document has sparse vectors. If each document only contains a small quantity of distinct word, this means that document vector are very sparse. Third, text has heterogeneous use of terms and fourth it also has high level of redundancy. Between each document, there are still possibilities of its document vectors to overlap each other. In this case, the word in particular document are may be contained in other documents identified as another distinct category. Last property is frequency distribution of words and Zipf's law.

According to Zipf's law, there is small number of words that occurs very frequently whereas most word occurs infrequently. Moreover, Zipf's law says that if one ranks words by their term frequency, the r-th most frequent word appears roughly 1/r times the term frequency of the most frequent words.

2. Literature Review

Document clustering means grouping a set of documents into different clusters so that two objectives are satisfied: First, similar documents should be grouped in the same cluster, and second, dissimilar documents should be placed in different clusters. The influence of the Internet is continuously increasing due to its impact on the peoples' lifestyles, working policies, communications and what not. Due to the easy use of different web services, such as the blogs, social networking sites, service providing sites and news agencies, a huge amount of data are being added in the World Wide Web (Web). However, most of these data are unstructured. Hence, when a user needs to find an item, s/he needs the support of a search engine. Upon getting a request, a search engine needs to return similar documents related to the search topic. Here document clustering plays a great role to improve the results and the complexity of such search as because it clusters similar documents in one group. Document clustering is, hence, a much well studied and have got attraction of many researchers in the information retrieval field [6]. In the literature, many algorithms are found to cluster text documents. K-means [2]-[5], [7] clustering is one of the most popular clustering algorithms. It clusters a given set of documents into K different groups. It is very simple and computationally efficient. However, in K-means clustering, user needs to specify how many clusters are expected (K's value) in advance, which may sometimes be difficult to set. Moreover, the Kmeans clustering algorithm starts with an initial (usually random) clustering, on which the performance largely depends [7]. In recent years, a number of proposals are found that use genetic algorithm [6]. Genetic algorithm is an optimization technique. It follows the evolution principles through randomized natural selection [8]. It is known to achieve very good (near optimal) solution, especially when the space is large and multi-modal [9]. Though the performances of genetic algorithms in document clustering are found to be better than the other available methods [6], it may converge to local optimal values. This is known as premature convergence phenomenon (PCP) [10]. To avoid PCP, a Double Layered Genetic algorithm for document Clustering (DLGC) [11] is proposed. However, it needs very high computation if the number of generations used in the first layer becomes high. Moreover, it also requires specifying the number of clusters in advance. We propose a two-phase genetic algorithm-based evolutionary approach for text document clustering. Instead of applying genetic algorithm on the whole dataset, we partition the dataset into some groups and apply genetic algorithm to each of the partitions separately. Finally, we apply another genetic algorithm phase on the outcomes of the earlier ones. This allows to get rid of the local minima. Unlike most of the available methods, specifying the expected number of clusters in advance is also not required in our proposal. Experimental results also demonstrate the superior performance of the proposed approach as compared to the other available approaches.

3. Method

3.1. K-Means

The K-Means is a partitioning method of clustering where n documents are partitioned into k partitions / clusters in such a way that each cluster has at least one document and each document belongs to only one cluster. The second condition is sometimes relaxed if we know that a document can belong to

more than one topic or subject. This is one of the simplest methods and creates clusters which are spherical in shape. This algorithm works well for a small corpus. The time complexity of this algorithm is linear in the number of documents. K-means is based on the concept that a center point can represent a cluster. In particular, for K-means we use the notion of a centroid, which is the mean or median point of a group of points (in this case documents). Note that a centroid almost never corresponds to an actual data point. In the K-Means algorithm the input is the number of clusters k, the corpus containing the documents to be clustered and k initial arbitrary documents. The output contains k clusters of documents.

The algorithm works as follows:

1. Select arbitrarily K documents as the initial centroids.

2. Assign each document to the closest centroid using some similarity function.

3. Re-compute the centroid (mean) of each cluster.

4. Repeat steps 2 and 3 until the centroids do not change.

3.2. GA (Genetic algorithm)

In our algorithm, first we initialize the population. Then we call GA function and DDE function one by one for making new population. In GA function, Crossover and Mutation operators are applied. After applying the operators, if new chromosomes are better, then these chromosomes are added to the new population otherwise old solutions are added into the new population. When DDE is called, first we find out global best chromosome from the old population, then apply Mutation operator on that best solution. After then, we apply Crossover between old population and global best chromosome and make temporary population. Then we compare old population chromosomes with temporary population chromosomes and make new population with better chromosomes. The steps of algorithms are given below. In the algorithm (pc) GA is the Crossover probability for Genetic Algorithm, (pc)DDE is the Crossover probability for Discrete Differential Algorithm, (pm)GA is the Mutation probability for Genetic Algorithm, (pm)DDE is the Mutation probability for Discrete Differential Algorithm, *iter*ation is the no. of iterations, and *np* is the population size.

Algorithm

1. for i ←1 to np (population initialization) do

a) Randomly select k documents from n documents and consider them k centers of k clusters. These k centers are one chromosome of initial population.

b) For every (n-k) documents do i. find distances from k centers by n×n matrix.

ii. Cluster document with any center according to minimum distance.

c) Find out the fitness value of this chromosome by fitness formula.

2. Consider initial population as the oldpop.

3. for i ←1 to iter do

a) if i is odd then

i. call GA_function

b) else

i. call DDE_function

4. Find out best chromosome from the last population

This chromosome has best clusters of documents.

GA_function

1. for j=1 to np/2 (Make new population by

Crossover) do

a) Select two chromosomes from oldop randomly.

b) if crossover probability (pc)GA is satisfied then

i. apply Crossover between chromosomes and find out two offspring.

ii. Calculate fitness values of two offspring by fitness formula.

iii. Select best two among old chromosomes and two offspring.

iv. Add them into newpop.

c) Else

i. add the chromosomes into newpop.

2. for every chromosome in newpop do

a) if mutation probability (pm)GA is satisfied then

i) Apply Mutation upon chromosome and find out offspring.

ii) Calculate fitness value of offspring by fitness formula.

iii) Select best one between old chromosome and offspring.

iv) Update newpop by best one.

3. Update oldpop by newpop and empty newpop.

3.3. Text classification using Support vector machine

Text-based classification is a technique which may be used to identify different types of data from the applications' point of view. Different researches are going on to identify ways of finding out the classes of data from a set of input data. In the present paper, a text-based classifier has been implemented and this classifier model can be used to classify input text into one of two categories, as defined by the user. The classifier is first trained with an initial dataset using the principle of supervised learning. After the training process is complete, the classifier makes use of the trained data in order to classify any new input text that may be provided. The proposed model also offers an incremental approach to text classification as it dynamically trains the classifier from a new set of data provided by the users.

The proposed method consists of three connected sections, viz., Initialize, Train and Classify. Corresponding to each section, the algorithm is stated below:

3.3.1 Algorithm: Initialize

BEGIN

Read file containing the data instances for training

Read file containing the corresponding labels

Convert all data instances to lowercase

Remove all punctuation marks and other non-alphanumeric characters from the data instance

Split each data instance to its constituent words and remove stop words

END

3.3.2 Algorithm: Train

BEGIN

LOOP through each data instance and label pair in training data

FOREACH word w in the data instance

IF w is not in WordList hash table THEN

Add w to WordList

IF the corresponding label is category 1 THEN

Set category 1 value of w to 1 and category 2 value to 0

ELSE

Set category 1 value of w to 0 and category 2 value to 1

END IF

ELSE

IF the corresponding label is category 1 THEN

Add 1 to the category 1 value of w

ELSE

Add 1 to the category 2 value of w

END IF

END IF

END FOR

END LOOP

END

3.3.3 Algorithm: Classify

BEGIN

LOOP while user wants to classify data

Read data entered by the user

Preprocess the data as in the training process

Compute the sum of the category 1 and category 2 values of the words in the data instance

Let these sums be sum1 and sum2 respectively

IF sum1 > sum2 THEN

Classify data instance as belonging to category 1 by assigning the appropriate label

ELSE IF sum2 > sum1 THEN

Classify data instance as belonging to category 2 by assigning the appropriate label

END IF

Add information about the newly classified data instance and its label to the hash table

Write the newly classified data instance and its corresponding label to disk

END LOOP

3.4. Fuzzy c-means (FCM)

Probabilistic fuzzy cluster analysis **Error! Reference source not found.Error! Reference source not found.** relaxes the requirement: $u_{ij} \in \{0,1\}$, which now becomes: $u_{ij} \in [0,1]$. However $\sum_{i=1}^{c} u_{ij} = 1, \forall j \in \{1,\dots,n\}$ still holds. FCM optimizes the following objective function:

$$J_f = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 . \qquad (1)$$

Parameter *m*, *m*>1, is called the fuzzyfier or the weighting exponent. The actual value of *m* determines the 'fuzziness' of the classification. It has been shown **Error! Reference source not found.** that for the case *m*=1, J_f becomes identical to J_h and thus FCM becomes identical to hard c-means.

The transformation from the hard c-means to the FCM is very straightforward; we must just change the equation for calculating memberships **Error! Reference source not found.** with:

$$u_{ij} = \frac{1}{\sum_{l=1}^{c}\left(\frac{d_{ij}^2}{d_{lj}^2}\right)^{\frac{1}{m-1}}} = \frac{d_{ij}^{\frac{-2}{m-1}}}{\sum_{l=1}^{c} d_{lj}^{\frac{-2}{m-1}}},$$

(Error! No text of specified style in document.)

and function for re-computing clusters centers**Error! Reference source not found.** with:

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m}. \qquad (3)$$

Equation (Error! **No text of specified style in document.**) clearly shows the relative character of the probabilistic membership degree. It depends not only on the distance of the object $\mathbf{x}_j$ to the cluster$\mathbf{c}_i$, but also on the distances between this object and other clusters

3.5. Proposed DB-scan

Clustering is the procedure of categorizing the nodes into altered collections through dividing sets of data into a series of subsets called clusters. Clusters can be discrete as a high density regions or low density area. Density is measured as the number of nodes in the "neighborhood". Clustering is a technique to achieve high data density. Density Based clustering is an efficient clustering method that determines the adequate number of clusters as well as provides the appropriate position for any cluster to initiate[4]. In this paper, Density based Spatial Clustering of Applications with Noise (DBSCAN) [3] algorithm is used for cluster formation. It is a density based algorithm which discovers clusters with arbitrary shape. The input parameters required for this algorithm is, Eps- radius of the cluster and MinPts - minimum nodes required inside the cluster. The basic idea behind this DBSCAN [3] algorithm is as follows, Eps –neighborhood of a Node: The Eps neighborhood of a node p, denoted by NEps (p) is defined by

$$\text{NEps (p)} = \{p \in DB \mid \text{dist (p, q)} \le \text{Eps}\} \qquad (4)$$

There are two kinds of nodes in the cluster, the node which is inside the cluster are called core nodes, and nodes on the border of the cluster are called border nodes which shows in Figure 1.

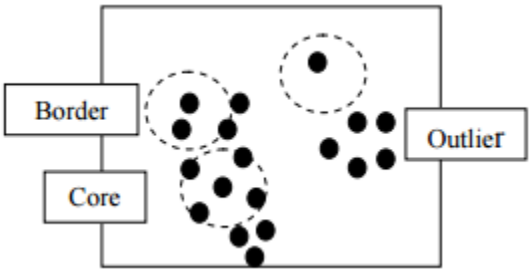

Figure 1: Core and Border nodes in Clusters formation

Directly density-reachable: A node p is directly density reachable from a node q w.r.t. Eps, MinPts if

1) p∈NEps (q) and
2)|NEps(q)|≥MinPts (5)

Density-reachable: A node p is density-reachable from a node q wrt. Eps and MinPts if there is a chain of nodes p1... pn, p1 = q, pn = p such that pi+1 is

directly density-reachable from pi as shown in Figure 2.

Density-connected: A node p is density-connected to a node q wrt. Eps and MinPts if there is a node o such that both p and q are density-reachable from o w.r.t. Eps and MinPts as shown in Figure 3

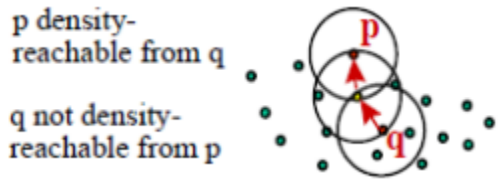

Figure 2: Density-reachable in DBSCAN

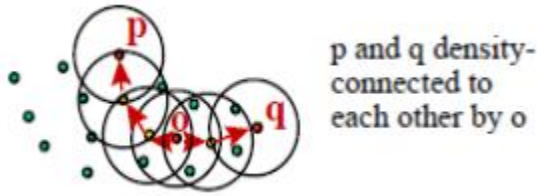

Figure 3: Density-connectivity in DBSCAN

Cluster: Let D be a database of nodes. A Cluster C w.r.t. Eps and MinPts is a non-empty subset of D satisfying the following conditions:

1) ∀p , q: if p ∈ C and q is density-reachable from p w.r.t. Eps and MinPts, then q Є C. (Maximality)

2) ∀p , q ∈ C: p is density-connected to q wrt. Eps and MinPts (Connectivity)

Noise: The noise node is defined as the set of points in the database DB not belonging to any cluster C, i.e. noise = {p ∈ DB | ∀ : p ∉ C}.

For the duration of the Cluster Formation stage, the Density based Spatial Clustering of uses with Noise (DBSCAN) algorithm which starts with an arbitrary selects a preliminary node p that has not been visited and it retrieves all neighbor nodes density accessible from starting node p with respect to Eps and MinPts. If the no. of neighbors is greater than or equal to MinPts then the cluster is formed. And also it discovers the Core nodes and Border nodes. Neither Core nodes nor Border node in the area will be marked as Noise node.

The Core node initiates a clustering procedure and nearby nodes will be added into the queue for the further expansion. Each nodes in a queue will be popped out and find the Eps neighbor node for the popped out node. When the new node is a core node, all its neighbor nodes will be assigned with the current cluster id. Then its unprocessed neighbor nodes will be pushed into queue for further processing. This process will be repeated until there are no nodes in the queue for the further processing.

DBSCAN could merge two clusters into one cluster, if two clusters of different density are "close" to each other. Let the distance between two sets of nodes C1 and C2 be defined as dist (C1, C2) = min {dist (p, q) | p∈ C1, q∈ C2}. Consequently, a recursive call of DBSCAN may be necessary for the detected clusters with a higher value for MinPts. The following illustrate the pseudo code of DBSCAN.

It stands for Density Based Spatial Clustering Algorithm with Noise. Though this is a density based algorithm it has been found to give very good results in text clustering also. DBSCAN requires two parameters: epsilon (eps) and minimum points (minPts). In the textual data eps would be the value of cosine distance (between 0 and 1) and minPts generally works well for 4 (at least four documents are similar to the document under consideration). For the following algorithm, a point is a document.

1. It starts with an arbitrary starting point that has not been visited. It then finds all the neighbor points within distance eps of the starting point.

2. If the number of neighbors is greater than or equal to minPts, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited.

3. The algorithm then repeats the evaluation process for all the neighbors' recursively.

4. If the number of neighbors is less than minPts, the point is marked as noise.

5. If a cluster is fully expanded (all points within reach are visited) then the algorithm proceeds to iterate through the remaining unvisited points in the dataset. This algorithm has an advantage that it does not require to know the number of clusters in the data a priori and it can also detect noise. Moreover the clusters unlike K-Means are of arbitrary shape.

4. Result

Owing to the rapid development of information technology and the rising popularity of the Internet, obtaining information is becoming easy, but turning diverse data into useful information often takes a long time. Obtaining useful information has become a very important research topic. Data mining responds to this demand. Density-based data clustering techniques are data mining techniques with high clustering accuracy that can remove noise from datasets, but the repeated diffusion process that they

involve takes much time, resulting in poor efficiency. This work proposed the FGKM-DBSCAN data clustering technique, which retains the following advantages of the density-based clustering method. The proposed two-phase filtering technique, which comprises data points filtering method for the original circles and their smaller inner circles and dynamic polygon boundary expansion method. This scheme greatly reduces the time cost as well as improves the noise filtering rate. The experimental results of this paper demonstrate that the FGKM-DBSCAN algorithm is superior to other clustering algorithms such as K-mean, GA and C-mean. It has a high noise filter rate, and a significantly better time cost. In sum, the FGKM-DBSCAN algorithm is a high-efficiency and high-quality data clustering technique.

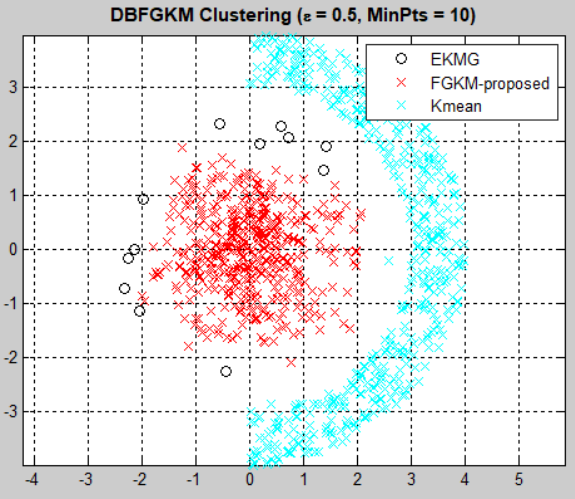

Figure 4: Cluster formation using K-mean, EKMG and prosed FGKM clustering

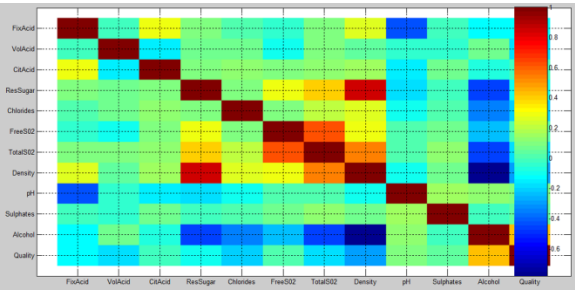

Figure 5: classification of clustering using SVM

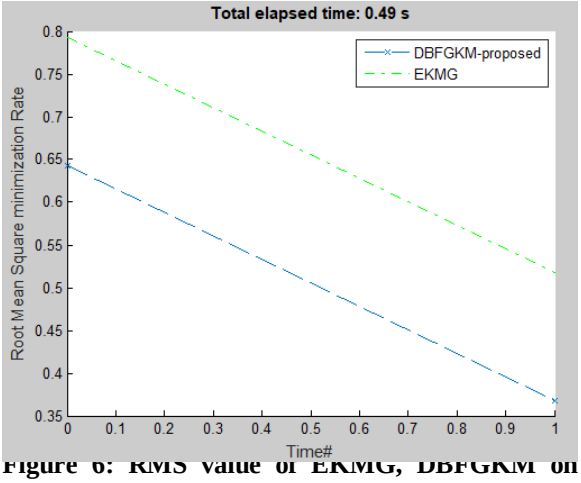

Figure 6: RMS value of EKMG, DBFGKM on Time

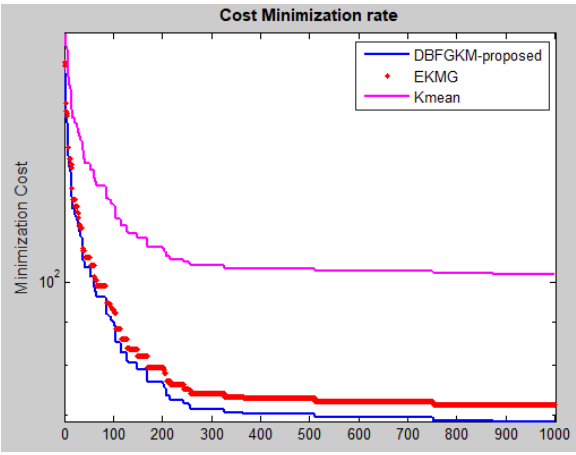

Figure 7: cost minimization of K-mean, EKMG and proposed-DBFGKM

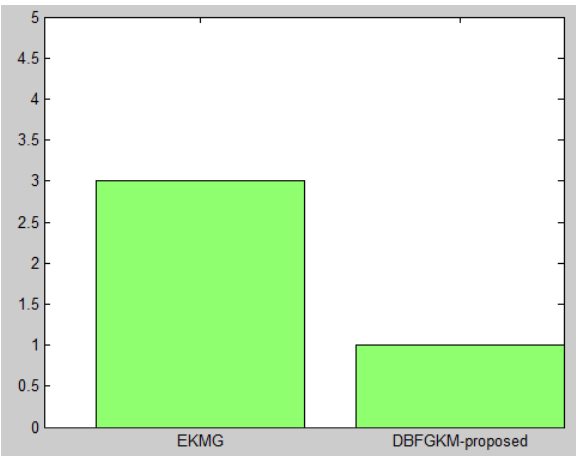

Figure 8: Error graph of two different data that is EKMG and DBFGKM-proposed

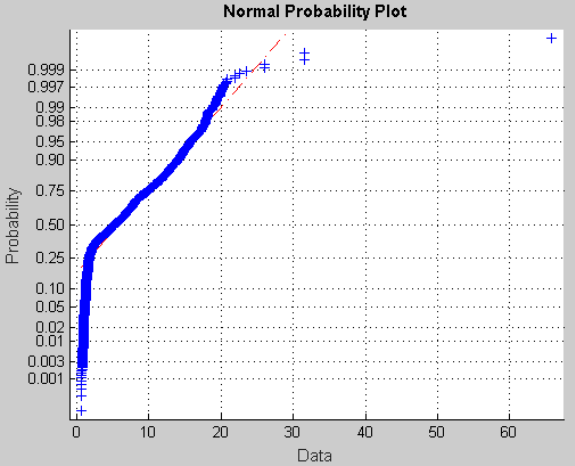

Figure 9: Normal probability plot

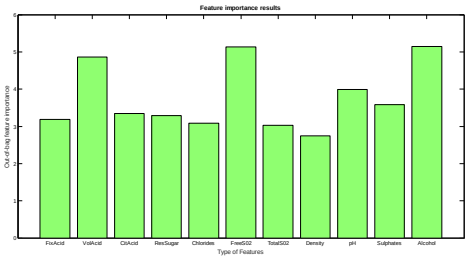

Figure 10: Features extraction of overall data

5. Applications of Proposed algorithm

Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

6. Conclusion

Clustering is the process to divide a data sets into the predetermine class or cluster, it is widely used in the field of information technology to increases the process of information retrieval as well as it is used in field of natural language processing. This paper introduces the comparison between different text clustering algorithm based on some parameters which can be used to develop the new clustering algorithm which can efficiently cluster the data well as can work on the arbitrary data very well.

Future work

There is a lot of scope in this field of Text Clustering. Methods like the fuzzy clustering are becoming popular as these methods apply the concept of fuzziness i.e. a three valued logic like true, false and maybe for a document to belong to a cluster. There are other hierarchical methods also where still there is scope for research. There are clustering methods related to neural networks also.

References

[1]Joachims, T. A Statistical Learning Model of Text Classification for Support Vector Machines In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 2001, p. 128-136

[2] Jain, A.K., Murty, M.N., and Flynn P.J. 1999. Data Clustering: A Review. ACM Computing Surveys, Vol. 31, No. 3.

[3] Jain, A.K. 2008. Data Clustering: 50 years beyond K-means. In 19th International Conference on Pattern Recognition. Tampa, FL.

[4] Jain, A.K., and Dubes, R.C. 1988. Algorithms for Clustering Data.Prentice-Hall Advanced Reference Series. Prentice Hall, NJ.

[5] Duda, R.O., Hart, P.E., and Stork, D.G. 2000. Pattern Classification, 2nd Edition. Wiley-Interscience.

[6] Song, W., and Park, S.C. 2009. Genetic Algorithm for text clustering based on latent semantic indexing, Computers and Mathematics with applications, vol. 57, pp. 1901-1907.

[7] Cha, S.M., and Kwon, K.H. 2001. A new migration method of the multipopulation genetic algorithms.The Korea Institute of Information Scientists and Engineers.

[8] Maulik, U. and Bandyopadhyay, S. 2000. Genetic algorithm-based clustering technique.Pattern Recognition, vol 33, no. 9, pp 1455-1465.

[9] Srinivas, M. and Patnaik, L.M. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE transactions on System, Man and Cybernatics, vol. 24, no. 4, pp. 656-667

[10] Andre, J., Siarry, P., and Dongon, T. 2001. An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization. Advances in Engineering Software, vol. 32, no. 1, pp. 49- 60

[11] Choi, L.C., Lee, J.S., and Park, S.C. 2011. Double layered genetic algorithm for document clustering. Communications in Computer and Information Science, vol. 257, pp. 212-218.

[12]Joachims, T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features, 1997.