Time Improving Policy of Text Clustering Algorithm by Reducing Computational Overheads

Mamta Gupta
Department of Computer Science
M.E. Student of SVITS Indore, India.

Anand Rajavat
Department of Computer Science
Prof. SVITS Indore, India.

ABSTRACT
Since the amount of text data stored in computer repositories is growing every day, we need more than ever a reliable way to assemble or classify text documents. Clustering can provide a means of introducing some form of organization to the data, which can also serve to highlight significant patterns and trends .Document clustering is used in many fields such as data mining and information retrieval. This paper presents the results of an experimental study of some common document clustering techniques. In particular, we compare the two main approaches of document clustering, agglomerative hierarchical clustering Modified BIRCH and Partitional clustering algorithm K-means. As a result of comparing both algorithms we attempt to establish appropriate clustering technique to generate qualitative clustering of real world document.

Keywords
Document clustering, BIRCH, K-means, Support Vector Model, Matrix Representation.

1. INTRODUCTION
Cluster analysis refers to a family of procedures which are fundamentally concerned with automatically arranging data into meaningful groups. The grouping process, generally referred to as clustering, attempts to divide a set of data objects so that those assigned to the same group share common characteristics, while those assigned to different groups are conceptually unrelated. In certain situations, this form of data analysis may be used to verify whether or not a dataset contains patterns that are assumed to exist according to a particular hypothesis. For other purposes, the identification of clusters represents the initial phase in a larger application, where it may be used as a means of summarizing or compressing data. However, cluster analysis is increasingly being employed as an important tool in knowledge discovery tasks. In this context, it forms an integral part of exploratory data analysis, where users may be unfamiliar with the exact contents of the data and may wish to introduce some form of organization, or identify important trends [6]

Data clustering concerns how to group a set of objects based on their similarity of attributes and their proximity in vector space. With partitioned clustering the algorithm creates a set of data non-overlapping subsets (clusters) such that each data object is in exactly one subset. These approaches require selecting a value for the desired number of clusters to be generated. A few popular heuristic clustering methods are k-means and a variant of k-means-bisecting k-means, k-medoids, PAM , CLARA , CLARANS etc. With hierarchical clustering the algorithm creates a nested set of clusters that are organized as a tree. Such hierarchical algorithms can be agglomerative or divisive approaches. Agglomerative algorithms, also called the bottom-up algorithms, initially treat each object as a separate cluster and successively merge the couple of clusters that are close to one another to create new clusters until all of the clusters are merged into one. Divisive algorithms, also called the top-down algorithms, proceed with all of the objects in the same cluster and in each successive iteration a cluster is split up using a flat clustering algorithm recursively until each object is in its own singleton cluster. The popular hierarchical methods are BIRCH, ROCK, Chamelon and UPGM. An experimental study of hierarchical and partition clustering algorithms was done by and proved that bisecting kmeans technique works better than the standard kmeans approach and the hierarchical approaches. Density-based clustering methods group the data objects with arbitrary shapes. Clustering is done according to a density (number of objects), (i.e.) density-based connectivity. The popular density-based methods are DBSCAN and its extension, OPTICS [3]. Grid-based clustering methods use multiresolution grid structure to cluster the data objects. The benefit of this method is its speed in processing time. Some examples include STING, Wave Cluster. Model-based methods use a model for each cluster and determine the fit of the data to the given model. It is also used to automatically determine the number of clusters. Expectation-Maximization, COBWEB and SOM (Self-Organizing Map) are typical examples of model-based methods. Frequent pattern-based clustering uses patterns which are extracted from subsets of dimensions, to group the data objects. Constraint-based clustering methods perform clustering based on the user-specified or application-specific constraints. It imposes user's constraints on clustering such as user's requirement or explains properties of the required clustering results.[7]Among all these methods, this paper is aimed to explore two methods –K-means which is partitioning based clustering method and Hierarchal based clustering methods. We compare them by using some criterion function.

2. TEXT PREPROCESSING
Before introducing the clustering algorithms themselves, it is necessary to examine the Preliminary phase of the cluster analysis process, which is concerned with producing a machine-interpretable representation from a document collection. As we shall see, the nature of the preprocessing techniques that are applied in this context can greatly influence the outcome of any subsequent clustering procedure.

2.1 Document Parsing
Given a new collection of unstructured text documents {d1, . . . , dn}, the first task is to apply a once-off parsing process, where the set of raw documents is transformed into a data model which can be subsequently analysed by a machine learning algorithm. This task generally involves applying a chain of procedures to each document:

Tokenisation: This initial procedure transforms the content of a document into a sequence of terms, representing words or phrases, which will subsequently be used to characterise the document. In some cases it may be useful to preserve information regarding the relative ordering of terms, depending upon the choice of data model.

Stemming: To reduce the number of unique terms, it is generally useful to stem terms to their roots. For the English language, the standard procedure is to apply the Porter suffix stripping algorithm to eliminate common morphological and inflectional endings (e.g. "programming"!"program"). A variety of open and commercial techniques are available for stemming documents written in other languages.

Stop-word removal: It will often be the case that it is not necessary to include all terms from the original corpus vocabulary in the data model. Notably, in text mining tasks it is extremely common to remove basic functional words (e.g. "the", "if") which occur so frequently in documents that they have no discriminating power and can be considered to be noise.

2.2 Vector Space Model

The choice of a suitable model to express document-term relations is fundamental to the success of text mining tasks. The vector space model, also referred to as the "bag of words" approach, has been the dominant method for representing documents in information retrieval, text classification and clustering problems. In this model, each document dj is represented by a vector xj = {f1, . . . , fm} in a m-dimensional term space, where m is the total number of unique terms across all documents in the corpus and fi indicates the frequency of occurrence of the i-th term in dj . Once an entire corpus of n documents has been transformed to a corresponding set of feature vectors {x1. . . xn}, the documents can be clustered based on the similarities or dissimilarities between vectors. For convenience, the complete model is usually stored as a single term-document matrix

A = [x1 x2 . . . xn] 2 IRm×n

where the entry Aij indicates the frequency of the i-th term in the document vector xj .

When employing this model, a significant amount of information about the original documents is lost, including information describing the ordering and context of terms. Some authors have suggested adapting the bag of words approach to use features constructed from phrases or n-grams, which consist of sequences of n consecutive characters extracted from text strings. However, the former can greatly exacerbate problems related to sparsity, while the latter reduces the interpretability of the model. Other radically different approaches for modelling text have been proposed, but none of these have been widely adopted. While the lack of spatial information and the sparse, high-dimensional nature of the term space do represent significant drawbacks, the vector space model remains the most popular choice due to its simplicity and the fact that traditional clustering algorithms working on numerical feature vectors can be directly applied to this representation. The underlying assumption here is that, if a pair of vectors is close to one another in the high-dimensional term space, the corresponding documents will share similar concepts.

2.3 Term Weighting

When working with text data, it is common to employ an additional preprocessing step, which involves replacing the original raw term frequency values with weighted frequencies calculated according to some normalisation function. This function is typically composed of two components: term frequency (tf) and inverse document frequency (idf). The former has the effect of increasing the impact of terms that occur frequently in a single document, while the latter seeks to reduce the influence of terms occurring in many documents, which may not be helpful in discriminating between the underlying classes in the data.

A wide variety of tf-idf weighting schemes were Among these, the ltc variant is most frequently employed in the document clustering literature, which applies logarithmic normalisation to both term frequency and document frequency values. Formally, the weighted frequency value for the i-th term in the document dj is defined as:

tfidf(i, j) = ltf (i, j) ·(logn/dfi) where

ltf (i, j) = 1 + log fi if fi > 0

0 otherwise.

where fi is the number of occurrences of the i-th term in dj and dfi is the total number of documents in the dataset which contain that term.

2.4 Similarity Measures

The choice of a suitable measure for quantifying the strength of association between pairs of documents is essential to the successful discovery of accurate groupings. For some clustering algorithms, a single pairwise similarity or dissimilarity matrix can be constructed as part of the preprocessing phase to avoid unnecessary computations during the subsequent clustering process. In other cases, the measure will be employed to pairs of documents during the execution of the algorithm itself. While a wide range of techniques for assessing similarity have been proposed in different application fields, we consider here three metrics that are interesting by the perspective of researchers working with text datasets.

Cosine similarity

Although Euclidean distance is useful in many domains, it has frequently been shown that it does not work well for high-dimensional data, due to the importance it places on absent values. As an alternative, the most commonly used method to compute the similarity between two documents when employing the vector-space model has been to measure the cosine of the angle between their corresponding vectors:

cos(xi, xj) = <xi,xj>/||xi|| · ||xj ||

Note that the numerator is the dot product between the two vectors, while ||xi|| in the denominator indicates the length of xi. This normalisation ensures that pairs of documents which differ in length, but have term frequencies in equal proportions, are considered to be identical. In that case, the value of Eqn. is one, while a value of zero indicates that a pair of documents does not share any common terms. For algorithms that make use of dissimilarity values, cosine distance may be computed by simply using:

dcos(xi, xj) = 1.0 − cos(xi, xj)

3. PARTITIONING BASED CLUSTERING

A partitioning method creates k partitions, called clusters, from given set of n data objects. Initially, each data objects are assigned to some of the partitions. An iterative relocation technique is used to improve the partitioning by moving objects from one group to another. Here, each partition is represented by either a centroid or a medoid. A centroid is an average of all data objects in a partition, while the medoid is the most representative point of a cluster [7]. The fundamental requirements of the partitioning based methods are each cluster must contain at least one data object, and each data objects must belong to exactly one cluster. In this category of clustering, various methods have been developed. A distance measure is one of the feature space used to identify similarity or dissimilarity of patterns between data objects [8].

k-means

K-means is one of the simplest unsupervised learning algorithms to group similar data objects [11].K-means forms clusters for *n* objects based on the attributes into *k* partitions where *k<n.* The algorithm starts by partitioning the input points into *k* initial sets, either at random or using heuristic data. It then calculates the mean point or centroid of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for new clusters, and the algorithm is repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters. The centroids should be placed in a cunning way as different centroid location provides different results.

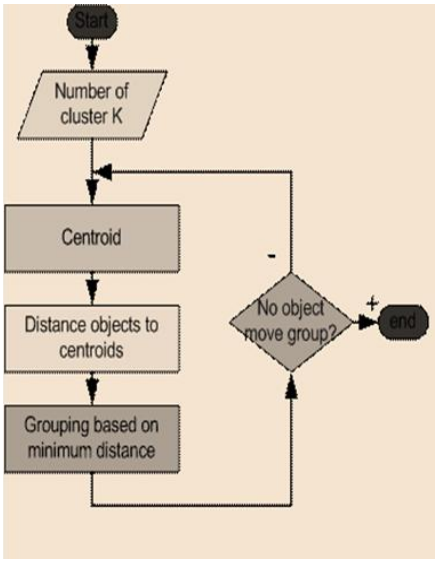

Fig 1-work flow of k-means algorithm

K-Means Clustering Algorithm

The main goal using K-means algorithm is to minimize the objective function, shown below

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2,$$

where *||xi(j) – cj||2* is a distance measure between a datapoint *xi(j)* and cluster center *cj* , showing the distance between n data points to their respective cluster centroids[12].This above equation clearly specifies that clusters are formed by minimizing the distance between the centroid and the data point. The

algorithm begins with assigning k centroids choosen randomly in a plane. All the points in the data set are assigned to a centroid that is nearest to it forming clusters. Once this initial arrangement is done, the next step will be to recalculate the centroid in each cluster by finding the center of the cluster from first step. This centroid is the point that is equidistant from all the points in that cluster. The next step is to again assign each point in the data set to the centroid in each cluster by finding the minimum distance between each point and every cluster and choosing the one with the minimum distance. Once again new centroid for every cluster is calculated. This looping is repeated until k centroids do not change their location. The diagrammatic representation of Kmeans algorithm is shown below followed by the steps in the process.

4. HIERARICHAL BASED CLUSTERING

Hierarchical clustering approaches attempt to create a hierarchical decomposition of the given document collection thus achieving a hierarchical structure as described in section Hierarchical methods are usually classified into Agglomerative and Divisive methods depending on how the hierarchy is constructed.

Agglomerative Methods

Agglomerative methods start with an initial clustering of the term space, where all documents are considered representing a separate cluster. The closest clusters using a given inter-cluster similarity measure are then merged continuously until only 1 cluster or a predefined number of clusters remain. Some of these methods are more suitable for discovering clusters of nonconvex forms than partition-based algorithms. Agglomerative methods normally produce hard (hierarchical) clusters. Agglomerative algorithms are usually classified according to the inter-cluster similarity measure they use. The most popular of these are single-link, complete-link and group average.

Divisive Methods

Divisive clustering algorithms start with a single cluster containing all documents. It then continuously divides clusters until all documents are contained in their own cluster or a predefined number of clusters are found. They thus work directly opposite of the agglomerative methods discussed above. These methods are usually significantly faster than the agglomerative methods, but have the drawback that “splits” or divisions cannot be undone to undo erroneous decisions – see appendix C for an example of this. Due to the divisive nature, these algorithms almost always produce a hard (hierarchical) clustering.

BIRCH

BIRCH (balanced iterative reducing and clustering using hierarchies) is an unsupervised data mining algorithm used to perform hierarchical clustering over particularly large data-sets. An advantage of BIRCH is its ability to incrementally and dynamically cluster incoming, multi-dimensional metric data points in an attempt to produce the best quality clustering for a given set of resources (memory and time constraints). In most cases, BIRCH only requires a single scan of the database.

5. MODIFIED BIRCH ALGORITHM

First of all, finding mean of all points of first document then second and then third.

We are using jaccard distance as similarity distance between mean and all points.

p1=1-((Pm-t1)/(pm+t1))

Repeat this process for all points.

Algo BIRCH (data I, B, T)

 # Phase 1

1. A= {{}} Starts with CF tree with a vacant leaf node.)

2. for each of the point p in I:

3. D= a leaf node in N that is closest to p means similar to p.

If value of d is small than others then

4. Append p to m.

5. Compute the diameter D of m.

6. If D> T:

7. Split (m) # may have need of splitting of ancestors of m

 # phase 2

8. Apply another clustering algorithm to cluster the leaves of A.

#phase 3

9. Repeat the process till all the point's p in I get located.

#phase 4

10. Final Result as one big cluster obtained.

6. RESULTS

Before applying algorithms on text data, we have first convert in text file format in which we have any sentence as one id and give some title to it and then applying clustering algorithm on them. Here we are different sizes of files and by comparing them we establish right clustering approach.

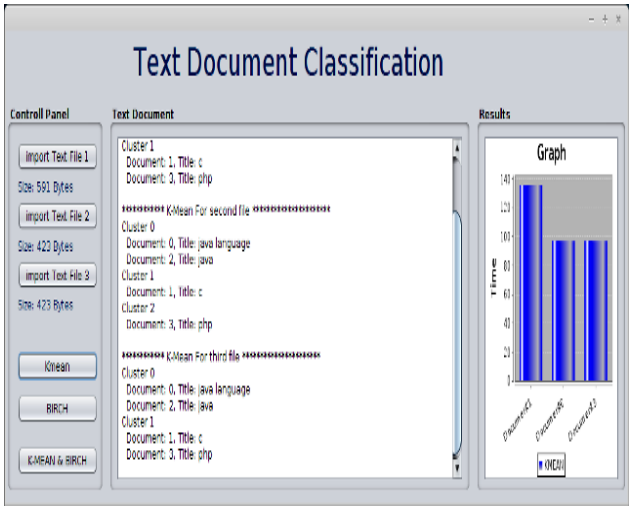

Fig 2: Applying k-means on text data

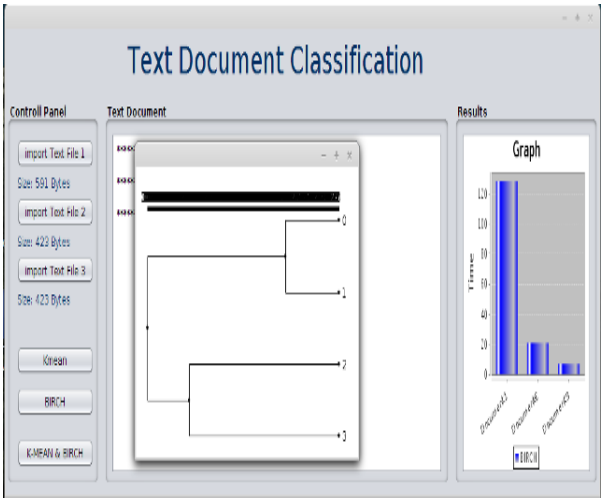

Fig 3: Applying Modified BIRCH on text data

On comparing both results of algorithms approaches we found that BIRCH takes less time for clustering the documents.

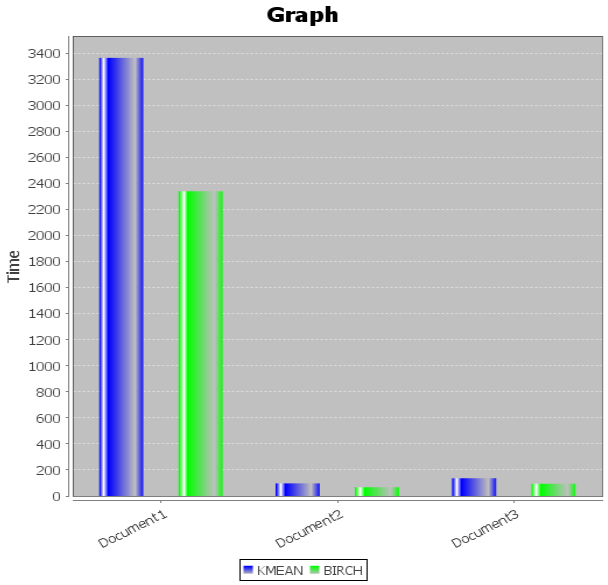

Fig 4: Comparison of K-means and Modified BIRCH

Table 1: Finding the time of k-means and Birch algorithm

DOCUMENT NAME	DOCUMENT SIZE	K-means Algorithm (time in millisecond)	BIRCH Algorithm (time in millisecond)
Document 1	423 bytes	160	20
Document 2	529 bytes	180	34
Document 3	14629 bytes	3400	44

7. CONCLUSION

In this paper, we will like to evaluate the performance of K-means, partition and modified BIRCH Agglomerative hierarchical approach by comparing them. Along with comparing clustering approaches we find that which is appropriate clustering algorithm to produce high feature quality clustering of real world documents. And also establishing a new algorithm which we try to make more efficient than existing clustering approaches which we already discusses in this paper. In the future we shall introduce improvement policies of accuracy, efficiency and memory in our algorithm.

8. REFERENCES

[1] Shi Zhong,"A k-means algorithm to improve the Efficiency Using Normal Distribution Data Points",(IJCSE) International Journal on Computer Science and Engineering, 2010.

[2] Xufei Wang, Jiliang Tang and Huan Liu, "Document Clustering via Matrix Multiplication" 2011 11th IEEE InternationalConference On Data Mining.

[3] Book: Information Retrieval, Algorithms and heuristics by David A.Grossman and Ophir Frieder.Published by Springer International.

[4] Anil K. Jain, "Pattern Recognition Letters", Journal Elsevier, Pattern Recognition Letters 31 (2010) 651–666.

[5] Atika Mustafa, Ali Akbar, and Ahmer Sultan"Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization", International Journal of Multimedia and Ubiquitous Engineering Vol. 4, No. 2, April, 2009.

[6] L. Wanner, "Introduction to Clustering Techniques",International Union of Local Authorities, July, 2004.

[7] T. Velmurugan,and T. Santhanam, "A Survey of Partition based Clustering Algorithms in DataMining: An Experimental Approach" An experimental approach.

[8] Porter, M.F.: An algorithm for suffix stripping.Program, Vol. 14, No. 3, 1980

[9] Na Wang; Pengyuan Wang; Baowei Zhang; , "An improved TF-IDF weights function based on information theory," Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On , vol.3, no., pp.439- 441, 12-13 June 2010.

[10] Lee, D.L.; Huei Chuang; Seamons, K.; , "Document ranking and the vector-space model,", IEEE , vol.14, no.2, pp.67-75, Mar/Apr 1997.

[11] Shobha S. Raskar, D. M. Thakore "Text Mining and Clustering Analysis", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011.

[12] Mrs .S.C.Punitha and Dr.M.Punithavalli, "A Comparative Study to Find A Suitable Method for Text Document Clustering", International Journal of Computer Science & Technology(IJCSIT)Vol 3,No 6 December 2011.