International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 3 Issue 11 November 2016

A Comparative study of file Clustering Algorithms

Diwakar Kumar[1] and Prof. Ketan Singh[2]
Malhotra Technical Research Institute, Bhopal[1, 2]
diwakar5883@gmail.com[1]

Abstract: *Web is a big source of text data and it increasing exponentially every day. To find useful text data from large text data is difficult. A File or Text clustering is unsupervised method for segregating text data in to significant clusters. The use of File clustering is to retrieve information from the large database. There are various method of File clustering but have some deficiency like big volume, high dimensionality and complex semantic. The aim of this survey paper is to compare differetypes of file clustering method with their merit and demerit.*

Keywords: *k-medoids, vector-space model, k-means, File clustering, DBSCAN, Grid-Based Methods, and Partitioning based algorithms.*

1. INTRODUCTION

A file clustering also called text clustering is method for segregating text information in to considerable clusters. Cluster analysis is crucial human commotion [1].

Most earlier studies of data mining have resolute on structured data object, such as relational, transactional, and data warehouse [4], However, in veracity, a large piece of the available information is stored in file databases (or text databases), which consist of large collections of files or text data from various sources, such as reports, research files, books, digital libraries, e-mail, and Web pages content. Text databases are quickly growing due to the increasing amount of information presented in electronic form, such as e-publications, various categories of electronic files, electronic-mail (e-mail), and the World Wide Web (www) [14]. Nowadays most of the data in government, industry, business word, and other institutions are stored electronically, in the form of file databases (Text database).

Data stored in most Files are partially structured data in that they are neither completely structured nor completely unstructured. For example, a file may contain a few structured fields, such as title name, authors name , publication date of file, category, and so on, but also hold some largely unstructured text data, such as abstract and contents[4],. There has been a deal of studies on the implementation and modelling partially structured Data in current database study, Moreover, information retrieval methods, such as text indexing techniques, have been developed to handle unstructured files. Traditional information retrieval method becomes insufficient for the increasingly vast amounts of text data. Classically, only a small fraction of the many available files (text) will be applicable to a given individual user. Without knowing what could be in the record or files, it is very difficult to formulate effectual queries for analyzing and extracting useful information from the data base. Users require tools to compare different records or files, rank the relevance and importance of the files, or find out patterns and trends from multiple files. Thus, text mining has become so popular and essential theme in data mining

2. PRELIMINARIES

Clustering is a very challenging field of research in which their potential relevance poses their own special requirements.

2.1 The following are requirements of clustering in data mining:

2.1.1) Scalability:

A lot of clustering algorithms work well on small data sets containing smaller amount than several hundred data objects (items), however, a huge database may hold millions of objects (items) [11]. Clustering on a sample of a given large data set may lead to influenced results.

2.1.2) Ability to deal with different kind of attributes:

Many algorithms are designed to cluster interval-based (numerical) data (text data). However, applications may require clustering other types of data, such as binary, ordinal data and categorical (nominal), or mixtures of these data types.

2.1.3) Discovery of clusters which have arbitrary shape:

Many clustering algorithms determine clusters based on Manhattan distance measures or Euclidean. Algorithms which are based on such distance measures be liable to find sphere-shaped clusters with similar size and density. However, a cluster could be of any type shape. It is important to develop algorithms that can identify clusters of arbitrary shape.

2.1.4) Minimum requirements for area knowledge to determine input parameters:

Many clustering algorithms need users to input certain parameters in cluster cram (such as the number of desired clusters). The clustering results can be fairly sensitive to input parameters. Parameters are commonly difficult to determine, especially for data sets which containing high-dimensional objects or data. This not only burdens to users, but it also makes the quality of clustering difficult to control.

2.1.5) Ability to deal with noisy data:

Most real-world databases contain outliers or unknown or missing, or erroneous data. a number of clustering algorithms are perceptive to such data and may lead to clusters of poor quality.

2.1.6) Insensitivity and Incremental clustering to the order of input records:

Some clustering algorithms cannot integrate newly inserted records (i.e., database updates) into existing clustering structures and, instead, must find out a new clustering from scratch. Some types of clustering algorithms are sensitive to the order of input data. That is, given a set of data object (items), such an algorithm may return dramatically a different type of clustering's depending on the inculcate of presentation of the input objects. It is most important to develop incremental clustering algorithms and algorithms that are insensitive to the order of input.

2.1.7) High dimensionality:

A database or a data warehouse can contain several proportions or attributes. Many clustering algorithms are better at handling low-dimensional data. Human eyes are better at judging the superiority of clustering for up to three scope. Finding clusters of data objects (items) in high dimensional space is very difficult, especially considering that such data can be highly skewed and sparse.

2.2 File or text mining process: file clustering process is divided into six basic stages [16]

2.2.1) Text: Billions of files must be handled in an efficient way, No clear picture of what files or records suit the application. Text Characteristics: it involve different Text Data, Data Mining, Pattern Discovery Attribute Selection, Text Transformation, Text Data Pre-processing, Interpretation, Evaluation characteristics like several input modes, Dependency, Ambiguity, High dimensionality (sparse input)

2.2.2) Text Pre-processing: it has series of sub stages like Text cleanup, Tokenization, Parts Of Speech tagging etc.

2.2.3) Process Text Transformation (Attribute Generation): it has two main sub stages Text Representation and Feature Selection

2.2.4) Selection of Attribute: it involve Reduce dimensionality, Remove irrelevant attributes

2.2.5) Data mining/Pattern Discovery: involve Structured Database, Application-dependent, Classic Data Mining techniques.

2.2.6) Interpretation /Evaluation: Terminate if Results well-suited for application at hand. Iterate if Results not satisfactory but significant.

3. MATHEMATICAL EPRESENTATION OF TEXT DATA

Files are represented using the vector-space model (VSM) [6]. In this model, each file or records, d, is considered to be a vector, d, in the term of space (set of file "words"). In its simplest form, each file is represented by the (TF) vector, dtf = (tf1, tf2, …, tfn),where tfi is the frequency of the ith term in the file. (Normally very common words are stripped out completely and different forms of a word are compact to one canonical form). In addition, we use the version of this model that weights each term based on its inverse file frequency (IDF) in the file collection. (This discounts frequent words with slight discriminating power.) Finally, in order to account for records or files of different lengths, and each file vector is normalized so that it is of unit length.

The similarity between two files must be measured in some way if a clustering algorithm is to be used. There are a lot of possible measures for calculating the similarity between files, but the most commonly used method is the cosine measure, which is can be defined as

$$\text{cosine}(d1, d2) = (d1 * d2) / \|d1\| \|d2\|,$$

Where* indicates the vector dot product & ||d|| is the length of vector d.

Given a set, of files S and their corresponding vector representations, now we define the centred vector c to be

$$c = \frac{1}{|s|}\sum_{des} d$$

which is nothing more than the vector obtained by averaging the weights of the various terms present in the files of S. Analogously to files, the comparison between two centroidvectors and between a file and a centred vector are computed using the cosine measurei.e.,

$$\text{cosine}(d, c) = (d * c) / \|d\| \|c\| = (d * c) / \|c\|$$

$$\text{cosine}(c1, c2) = (c1 * c2) / \|c1\| \|c2\|$$

For K-means clustering algorithm [3], the cosine measure is used to calculate which file centroidis closest to a given file or records. While a mean is sometimes used as the centred for K-means clustering, we follow the ordinary practice of using the mean. The mean is easier to calculate than the midpoint and has a number of fine mathematical properties. For example, computing the dot product between a file and a cluster centred is equivalent to calculating the average similarity between that file and the files that comprise the cluster centroid represents Mathematically,

$$d1 * c = \frac{1}{|s|}\sum_{des} d\,1 * d = \frac{1}{|s|}\sum_{des} \text{cosine}(d_1, d)$$

Also the square of the length of the centred vector is just the average pair shrewd similarity between all points in the cluster. In the following section, we will use this average pair wise similarity property as the basis for one of the measures for quantifying the goodness of a clustering algorithm.

$$\frac{1}{|s|}2\sum_{\substack{des \\ d` \epsilon s}} \text{cosine}(d\sim, d) = \frac{1}{|s|}\sum_{des} d * \frac{1}{|s|}\sum_{des} d = c * c = \|c\|_2$$

4. COMPARISON PARAMETERS OF FILE CLUSTERING ALGORITHMS

The aim of file clustering is the grouping of several types of text data into similar classes[7]. Grouping is based on similarity between text data. Text data which have high similarity grouped into one cluster and data which have dissimilarity groped into another cluster. Therefore to compare text clustering algorithms we use some criteria:

4.1) Higher scalability: Clustering algorithm is not only used in the sample records sets, but also a large scale of the reality text data sets which should have a good outcome.

4.2) High dimensional data: The text data sets expressed by Vector Space Model typically have thousands or immobile web Dimensional, so algorithm for text clustering is able to grip high dimensional data.

4.3) Find out the arbitrary shape of clusters: File clustering algorithm can find the arbitrary shape of class.

4.4) The sequence of input data is not sensitive: The sequence of the input text data has no effect on the results of the final clustering.

4.5) Better ability to deal with noise data: The vast majority of databases contain isolated points, the unidentified data and so on, if the algorithm is susceptible to such data, it will reduce the quality of clustering results.

5. DIFFERENTTYPES OF FILE (TEXT) CLUSTERING ALGORITHMS

Different File clustering algorithms exist in the literature. In general, the most important clustering algorithms can be classified into the following Categories:

5.1. Partitioning Based Algorithms

Given a database of n items or data, a partitioning method constructs k partitions of the data or items, where each individual partition represents a cluster and k ? n. That is, it classifies the items into k cluster, which together satisfy the following necessities: (1) each cluster must contain at least one item, and (2) each item must belong to exactly one cluster. Notice that the second necessity can be relaxed in some fuzzy partitioning techniques.

Given k, the quantity of partitions to construct, a partitioning method creates primary partitioning. It then uses an iterative relocation method that attempts to improve the partitioning by moving items from one cluster to another. The general criterion of a good partitioning is that item in the same cluster are “close” or correlated to each other, whereas objects of different clusters are “far apart” or much unrelated. There are various kinds of other criteria for judging the quality of partitions.

The most familiar and commonly used partitioning algorithms are k-means, k-medics, and their variations.

5.1.1k-means method

The k-means method[3][15]for partitioning, where each cluster’s centre is represented by the correspond to value of the objects in the cluster.

Input:

k: the number of clusters,

D: a data set contains n items.

Output: A set of k clusters.

Method:

(1) Arbitrarily choose k item from D as the initial cluster centres;

(2) Repeat

(3) (Re) assign each item to the cluster to which the item is the most similar, based on the mean value of the item in the cluster.

(4) Revise the cluster means, i.e., calculate the mean value of the item for each cluster;

(5) Until no change;

The k-means algorithm can be useful only when the mean of a cluster is defined.

The k-means algorithms are not appropriate for discovering clusters with non convex shapes or clusters of very dissimilar size.

5.1.2 k-Medoids method

PAM (Partitioning Around Medoids), a k-medoids algorithm [11] for partitioning based on medoid or central objects.

Input:

k: the number of clusters,

D: a data set containing n item.

Output: A set of k clusters.

Method:

(1) Arbitrarily choose k item in D as the initial representative item or seeds;

(2) Repeat

(3) Assign each remaining item to the cluster with the nearest representative item;

(4) Arbitrarily select a non representative item, o random;

(5) Calculate the total cost, S, of swapping representative item, oj, with o random;

(6) If S < 0 then exchange oj with o arbitrary to form the new set of k representative item;

(7) Until no change;

The k-medoids algorithms [11] are more robust than k-means algorithms in the occurrence of noise and outliers, because a medoid is fewer influenced by outliers or other tremendous values than a mean. However, its processing is more expensive than the k-algorithms method. Both algorithms require the user to indicate k, the number of clusters.

Hierarchical algorithm and divisive Hierarchical algorithm, depending on whether the hierarchical breakdown in a bottom-up(integration) or top-down (split) fashion. The superiority of a pure hierarchical clustering process suffers from its incapability to perform adjustment once a merge or split decision has been executed. That is, if a merge or split decision later turns out to have been a poor choice, the technique cannot be backtrack and accurate it. Recent studies have emphasized the integration of hierarchical agglomeration with iterative relocation methods.

In general, there are two kind of hierarchical clustering algorithm [7]:

5.2.1 Agglomerative clustering algorithm:

This bottom-up strategy starts by insertion each item in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the items are in a single large cluster or until certain termination circumstances are satisfied. Mainly hierarchical clustering methods belong to this class[7]. They differ only in their definition of inter cluster similarity.

Assume a similarity function that finds the similarity of two cas (instances): sim(x, y), Cosine similarity of file vectors. Among the current clusters, find out, two clusters, ci and cj, that are most related. Replace ci and cj with a single cluster ci ? cj

Use maximum similarity of pairs:

$$sim(c_i,c_j) = \max_{x\in c_i, y\in c_j} sim(x,y)$$

$$sim(c_i,c_j) = \max_{x\in c_i, y\in c_j} sim(x,y)$$

Use minimum similarity of pairs:

$$sim(c_i,c_j) = \min_{x\in c_i, y\in c_j} sim(x,y)$$

$$sim(c_i,c_j) = \min_{x\in c_i, y\in c_j} sim(x,y)$$

$$sim(c_i,c_j) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right|-1\right)} \sum_{\vec{x}\in(c_i\cup c_j)} \sum_{\vec{y}\in(c_i\cup c_j):\vec{y}\neq\vec{x}} sim(\vec{x},\vec{y})$$

$$sim(c_i, c_j) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right| - 1\right)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j): \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y})$$

Use average similarity between all pairs within the combined cluster to measure the similarity of two clusters.

5.2.2 Divisive clustering algorithm:

This top down scheme does the reverse of agglomerative hierarchical clustering algorithm by starting with all objects (items) in one cluster. It subdivides the cluster into slighter

5.2 Hierarchical Algorithm

A hierarchical clustering algorithm works by grouping data item into a tree of clusters. [2] Hierarchical clustering methods can be further classified as agglomerative and slighter parts, until each object (item) forms a cluster on its own or until it satisfies certain extinction conditions, such as a desired number of clusters is obtain or the distance of each cluster is within a certain threshold.

5.3. Density-Based algorithms

A Density-based algorithm clusters objects based on the notion of density. It either grows clusters according to the compactness of neighbourhood objects (such as in DBSCAN) or according to some density utility (such as in DENCLUE). OPTICS is a solidity based technique that generates an augmented ordering of the clustering structure of the data. DENCLUE clusters objects based on a set of compactness distribution functions.

5.3.1)DBSCAN algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based clustering algorithm. The algorithm grows regions with satisfactorily high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. It also describes a cluster as a maximal set of density-connected points.

The basic ideas of density-based clustering absorb a number of new definitions. We instinctively present these definitions, and then follow up with an example.

The neighbourhood within a radius ε of a specified object is called the ε-neighbourhood of the object.

If the ε-neighbourhood of an object contains at least a minimum number, Mints, of objects, then the object is called a core object.

Given a set of objects (items), D, we say that an object p is unswervingly density-reachable from object q if p is inside the e-neighbourhood of q, and q is a core object.

An object p is solidity-reachable from object q with respect to e and Mints in a set of objects, D, if there is a sequence of objects p1,… , pan, where p1 = q and pan = p such thatpi+1 is directly density-reachable from pi with respect to e and MinPts, for 1 ? i ? n,pi ε D.

An object p is solidity-connected to object q with respect to e and MinPts in a set of objects, D, if there is an object o ε D such that both p and q are density-reachable from o with respect to and MinPts. Density reachability is the transitive closure of express density reachability, and this relationship is asymmetric. Only core objects (items) are mutually density reachable. Density connectivity is a symmetric relation.

5.4. Grid-Based Methods

A grid-based method first quantizes the entity space into a finite number of cells that form a grid structure, after that performs clustering on the grid structure. STING isan example of a grid-based method based on numerical information stored in grid cells. Wave Cluster and CLIQUE are two clustering algorithms that are both grid based and density-based.

Table.1comparison of different clustering algorithms

Name of algorithm	**Data sets size**	**Number of clusters**	**Data sets types which are used**	**adapt to dynamic data**	**Initial conditions**	**Condition of termination**
Density based Algorithm	Small and huge datasets	Small and large clusters	Random and ideal datasets	yes	yes	precise
Grid based algorithm	Small and huge datasets	Small and large clusters	Random and ideal datasets	yes	yes	precise
Partitioning Based algorithms	Small and huge datasets	Small and large clusters	Random and ideal datasets	yes	yes	precise
Hierarchical clustering	Small and huge datasets	Small and large clusters	Random and ideal datasets	no	no	Not precise

6. CONCLUSION

In this survey we had focused on various file clustering algorithms, and there feature selection methods, limitations, terms, advantages and available recent innovation in feature selection. We expect, that the interested readers will have broad overview of this field and several preliminary point for further details.

REFERENCES

[1] Zhao Y and Karypis G. "Hierarchical Clustering Algorithms for Document Datasets"[A]. Data Mining and Knowledge Discovery [C].10(2), pp.141-168, 2005.

[2] Hongbin, Gao, Haizhen Yang, Xiaobin Zhang. "an Improved Document Clustering Algorithm"[J] Computer Applications, 2008,27(9):30-32.

[3] Xiaojun Wang, Jianwu Yang, Xiaoou Chen. "an Improved K-means Document Clustering algorithms"[J] Computer Engineering, 2003, 29(2): 102-104.

[4] Salton G, Wong A, Yang C."A vector space model for automatic indexing" [J] .Communications of the ACM, 1975, 18(11) : 613- 620.

[5] Wache H, Vogele T, Vissers U, "Ontology-Based Integration of Information-A Survey of Existing Approaches"[C]. In: Proc of the IJCAE01 Workshop: Ontologies and Information Sharing. Seattle, WA, 2001:108-117.

[6] Kim H J, Lee S G. "A semi-supersvised document clustering technique for information and organization" [A].In: Proc of the Ninth International Conference on Information and Knowledge Management[C]. McLean, Virginia, 2002.159-168.

[7] Karypis G, Zhao Y. "Evalution of hierarchical clustering algorithms for document datasets" [A]. In: Proc of the International Conference on Information and knowledge Management[C]. New York, 2002. 512-524.

[8] Bing. Liu "Web Data Mining" [M] Tsinghua University Press, 2009.

[9] T. Kohonen. "Self-Organizing Maps". Series in Information Sciences, 30, Springer, Heidelberg, Second Edition. 1995.

[10] M. Steinbach, G. Karypis, and V. Kumar. "A comparison of document clustering techniques". Technical Report 00-034, University of Minnesota, 2000.
[11] Inderjit Dhillon, Jacob Kogan, and Charles Nicholas. "Feature selection and document clustering". In Michael W. Berry, editor, Survey of Text Mining, pages 73{100. Springer, 2003.
[12] Arindam Banerjee and Sugato Basu. "Topic models over text streams: A study of batch and online unsupervised learning". In SDM. SIAM, 2007.
[13] Ste_en Staab and Andreas Hotho. "Ontology-based text document clustering. In Intelligent Informa-tion Processing and Web Mining, Proceedings of the International "IIS: IIPWM'03 Conference held in Zakopane, pages 451{452, 2003.
[14] Daphe Koller and Mehran Sahami, "Hierarchically classifying documents using very few words", Proceedings of the 14th International Conference on Machine Learning (ML), Nashville, Tennessee, July 1997, Pages 170-178.
[15] Paul Bradley and Usama Fayyad, "Refining Initial Points for K-Means Clustering", Proceedings of the Fifteenth International Conference on Machine Learning ICML98, Pages 91-99. Morgan Kaufmann, San Francisco, 1998.
[16] O. Zamir and O. Etzioni, "Web document clustering: A feasibility demonstration", in Proceeding of 19th International ACM SIGIR Conference on Research and Development in Informational Retrieval, June1998.