A Brief Survey on Document Clustering Techniques using MATLAB

Rachitha Sony.Krotha[#1], Suneetha Merugula [#2]
Department of Information Technology,GMRIT Rajam, A.P, India.

Abstract *- Document clustering is a more specific technique for unsupervised document organization, it is generally considered to be a centralized process. Clustering methods can be used to automatically group the retrieved documents into a list of meaningful categories. This paper gives an overview of some of the mostly used document clustering techniques and introduces the matlab tool which provides us many functions that helps in the clustering of the documents. In particular we concentrate on the most commonly used clustering techniques Agglomerative hierarchical clustering and K-means that are commonly used for document clustering and related matlab functions available in the matlab toolbox.*

Keywords: *clustering, hierarchial clustering, K-means , Matlab toolbox*

1. INTRODUCTION

Document clustering[1] is considered as a centralized process has been in use in a number of different areas of text mining and information retrieval. Clustering can be considered the most important unsupervised learning problem so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. The main requirements that a clustering algorithm should satisfy are:

- scalability;
- dealing with different types of attributes;
- discovering clusters with arbitrary shape;
- minimal requirements for domain knowledge to determine input parameters;
- ability to deal with noise and outliers;
- insensitivity to order of input records;
- high dimensionality;
- interpretability and usability.

MATLAB or Matrix laboratory is a programming environment for algorithm development, data analysis, visualization, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran. MATLAB is the language of technical computing.. MATLAB is widely used in academic and research institutions as well as industrial enterprises.

The MATLAB application is built around the MATLAB language, and most use of MATLAB involves typing MATLAB code into the Command Window or executing text files containing MATLAB code and functions. MATLAB can create and manipulate arrays of 1 (vectors), 2 (matrices), or more dimensions. In the MATLAB vernacular, a vector refers to a one dimensional ($1\times N$ or $N\times 1$) matrix, commonly referred to as an array in other programming languages. A matrix generally refers to a 2-dimensional array, i.e. an m×n array where m and n are greater than 1. Arrays with more than two dimensions are referred to as multidimensional arrays. Arrays are a fundamental type and many standard functions natively support array operations allowing work on arrays without explicit loops. Therefore the MATLAB language is also an example of array programming language.

II.CLUSTERING TECHNIQUES

A. Hierarchial Clustering

Hierarchical techniques produce a nested sequence of partitions. The result of a hierarchical clustering algorithm can be graphically displayed as tree, called a dendogram. This tree graphically displays the merging process and the intermediate clusters. For document clustering, this dendogram provides a taxonomy, or hierarchical index. There are two basic approaches to generating a hierarchical clustering:

a) **Agglomerative**: Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.

b) **Divisive**: Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

B. K-Means Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple

and easy way to classify a given data set through a certain number of clusters.This algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

C.Basic K-means Algorithm for finding *K* clusters.

1. Select *K* points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change.

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a *d*-dimensional real vector, *k*-means clustering aims to partition the *n* observations into *k* sets ($k \leq n$) $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \boldsymbol{\mu}_i\|^2$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

III MATLAB TOOLS FOR CLUSTERING

A.Hierarchial clustering in Matlab

The Statistics Toolbox[4][5] in matlab provides a function clusterdata which supports agglomerative clustering and performs all of the necessary steps. It incorporates the pdist, linkage, and cluster functions, which can be used separately for more detailed analysis. The dendrogram function plots the cluster tree. Here a complete description is explained followed by an example.

Syntax

```
Z = linkage(X)
Z = linkage(X,method)
Z = linkage(X,method,metric)
Z = linkage(X,method,pdist_inputs)
Z = linkage(X,method,metric,'savememory',value)
Z = linkage(Y)
Z = linkage(Y,method)
```

Description

Z = linkage(X) returns a matrix Z that encodes a tree of hierarchical clusters of the rows of the real matrix X.

Z = linkage(X,method) creates the tree using the specified method, where method describes how to measure the distance between clusters.

Z = linkage(X,method,metric) performs clustering using the distance measure metric to compute distances between the rows of X.

Z = linkage(X,method,pdist_inputs) passes parameters to the pdist function, which is the function that computes the distance between rows of X.

Z = Linkage (X,method,metric,' savememory', value) uses a memory-saving algorithm when value is 'true', and uses the standard algorithm when value is 'false'.

Z = linkage(Y) uses a vector representation Y of a distance matrix. Y can be a distance matrix as computed by pdist, or a more general dissimilarity matrix conforming to the output format of pdist.

Z = linkage(Y,method) creates the tree using the specified method, where method describes how to measure the distance between clusters. The following cluster clearly explain the process.

An example: Hierachial Clustering in MATLAB
In the following example four clusters are computed of the Fisher iris data using Ward linkage and ignoring species information, and see how the cluster assignments correspond to the three species.

```
load fisheriris

Z = linkage(meas,'ward','euclidean');

c = cluster(Z,'maxclust',4);
crosstab(c,species)
firstfive = Z(1:5,:) % first 5 rows of Z
dendrogram(Z)
ans =
    0   25    1
    0   24   14
```

```
     0     1    35
    50     0     0
firstfive =
  102.0000  143.0000         0
    8.0000   40.0000    0.1000
    1.0000   18.0000    0.1000
   10.0000   35.0000    0.1000
  129.0000  133.0000    0.1000
```

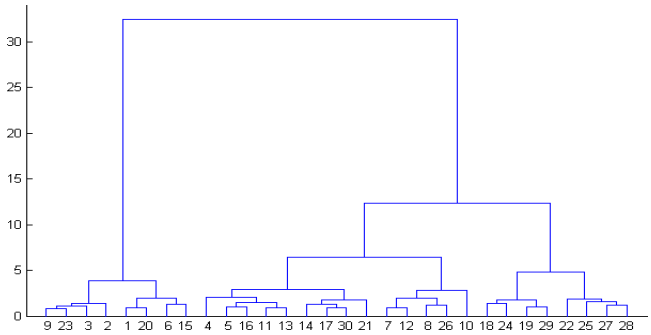

Fig 1: Hierarchial Clustering

Create a hierarchical cluster tree for a data with 20000 observations using Ward's linkage. If you set savememory to 'off', you can get an out-of-memory error if your machine doesn't have enough memory to hold the distance matrix. Cluster the data into four groups and plot the result. X = rand(20000,3);

Z=linkage(X,'ward','euclidean','savememory','on');

c = cluster(Z,'maxclust',4);
scatter3(X(:,1),X(:,2),X(:,3),10,c)

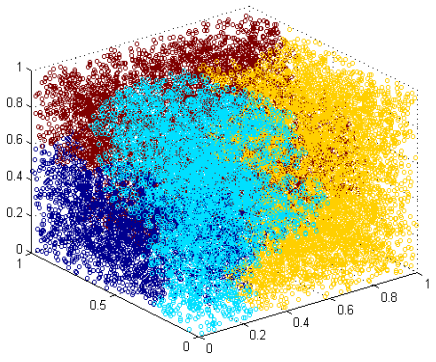

Fig 2: Scatter Plot Of Hierachial CLustering

B. K- Means clustering in Matlab

The Statistics Toolbox in matlab provides a function K-Means to cluster the data. The following is the description of the function K-Means with an example.

Syntax

IDX = kmeans(X,k)
[IDX,C] = kmeans(X,k)
[IDX,C,sumd] = kmeans(X,k)
[IDX,C,sumd,D] = kmeans(X,k)
[...] = kmeans(...,*param1*,*val1*,*param2*,*val2*,...)

Description

IDX = kmeans(X,k) partitions the points in the n-by-p data matrix X into k clusters. This iterative partitioning minimizes the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. Rows of X correspond to points, columns correspond to variables. kmeans returns an n-by-1 vector IDX containing the cluster indices of each point. By default, kmeans uses squared Euclidean distances. When X is a vector, kmeans treats it as an *n*-by-1 data matrix, regardless of its orientation.

[IDX,C] = kmeans(X,k) returns the k cluster centroid locations in k-by-p matrix C.

[IDX,C,sumd] = kmeans(X,k) returns the within-cluster sums of point-to-centroid distances in the 1-by-k vector sumd.

[IDX,C,sumd,D] = kmeans(X,k) returns distances from each point to every centroid in the n-by-k matrix D.

[...]=kmeans(...,*param1*,*val1*,*param2*,*val2*,...) enables you to specify optional parameter/value pairs to control the iterative algorithm used by kmeans.

An Example: K-Means Clustering in MATLAB
Here two clusters are formed from separated random data:

```
X = [randn(100,2)+ones(100,2);...
     randn(100,2)-ones(100,2)];
opts = statset('Display','final');

[idx,ctrs] = kmeans(X,2,...
                    'Distance','city',...
                    'Replicates',5,...
                    'Options',opts);
5 iterations, total sum of distances = 284.671
4 iterations, total sum of distances = 284.671
4 iterations, total sum of distances = 284.671
3 iterations, total sum of distances = 284.671
3 iterations, total sum of distances = 284.671

plot(X(idx==1,1),X(idx==1,2),'r.','MarkerSize',12)
hold on
```

```
plot(X(idx==2,1),X(idx==2,2),'b.','MarkerSize',12)
plot(ctrs(:,1),ctrs(:,2),'kx',...
    'MarkerSize',12,'LineWidth',2)
plot(ctrs(:,1),ctrs(:,2),'ko',...
    'MarkerSize',12,'LineWidth',2)
legend('Cluster1','Cluster ','Centroids',...
'Location','NW')
```

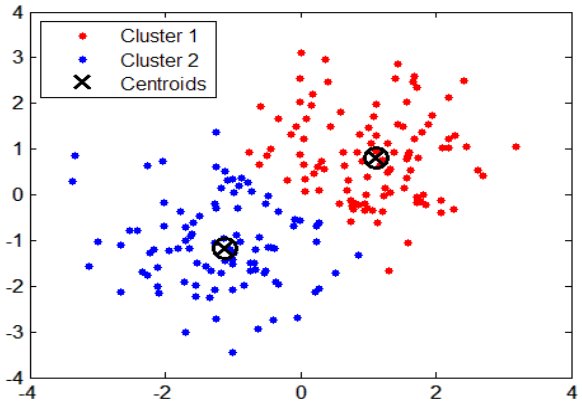

Fig 3: Graph K-Means

IV. CONCLUSION

In this article a brief introduction about the broad field of document clustering is discussed . Even though it is impossible to describe all the algorithms in detail, We tried to discuss the clustering methods in particular the most commonly used, hierarchial clustering and K-Means clustering, their properties and applications using MATLAB. The idea discussed and provided references should give the interested reader a rough preview of MATLAB and its tools that are helpful in further studies.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incompleted without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success. We would like to express our deep sense of gratitude and sincere thanks to our college management for providing me an opportunity with all required facilities in completion of the work.

REFERENCES

[1] Jiawei Han and Micheline Kamber., "Data Mining Concepts and Techniques", Elsevier Pubications.
[2] Rajan chatamvelli "Data Mining Methods", Narosa publishing house.
[3] Manu Konchady, "Text Mining Application Programming", Cengage Learning
[4] Stephen J.Chapman, "MATLAB Programming for Engineers", Thomson Learning third edition.
[5] *Statistics Toolbox User's Guide*. (September 2009), Available at:
[6] 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.
[7] Benjamin C. M. Fung, Ke Wang, and Martin Ester, Hierarchical Document clustering.
[8] Moses Charikar, Chandra Chekuri, Tomas Feder, and Rajeev Motwani, *Incremental Clustering and Dynamic Information Retrieval*, STOC 1997, Pages 626-635, 1997.
[9] Javed Aslam, Katya Pelekhov, and Daniela Rus, *A Practical Clustering Algorithm for Static and Dynamic Information Organization*, Proceedings of the 1998 ACM CIKM International Conference on Information and Knowledge Management, Bethesda, Maryland, USA, Pages 208-217, November 3-7, 1998.
[10] 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.
[11] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, (1998*), ROCK: A Robust Clustering Algorithm forCategorical Attributes*, In Proceedings of the 15th International Conference on Data Engineering, 1999.
[12] 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.
[13] Charu C. Aggarwal, Stephen C. Gates and Philip S. Yu, On the merits of building categorization systems by supervised clustering, Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 352 – 356, 1999