ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Document Deliver Using Clustering Techniques and Finding Best Cluster

Jyoti, Neha kaushik, Rekha

Abstract- Clustering is an important tool in data mining and knowledge discovery. This enables the users to comprehend a large amount of data. We develop DSSC (Document Similarity soft clustering), a soft-clustering algorithm based on the similarity function given. DSSC is similar to many other soft clustering algorithms like fuzzy C-means. That is, it starts out with a carefully selected set of initial clusters, and uses an iterative approach to improve the clusters. At the end, DSSC produces a set of clusters with each document belonging to several potential clusters. This approach only requires a similarity function to be defined properly, and does not rely on any underlying probability assumptions. This allows DSSC to overcome problems of standard soft clustering algorithms mentioned above, without paying any price in efficiency (in fact, we perform K-means based algorithm in many cases.

I. INTRODUCTION

Cluster Analysis:

Grouping similar customers and products is a fundamental marketing activity. It is used, prominently, in market segmentation. As companies cannot connect with all their customers, they have to divide markets into groups of consumers, customers, or clients (called segments) with similar needs and wants. Cluster analysis is a convenient method for identifying homogenous groups of objects called clusters. Objects (or cases, observations) in a specific cluster share many characteristics, but are very dissimilar to objects not belonging to that cluster. The first step is to decide on the characteristics that you will use to segment your customers. In other words, you have to decide which clustering variables will be included in the analysis. The objective of cluster analysis is to identify groups of objects (in these case, customers) that are very similar with regard to their price consciousness and brand loyalty and assign them into clusters. After having decided on the clustering variables (brand loyalty and price consciousness), we need to decide on the clustering procedure to form our groups of objects. This step is crucial for the analysis, as different procedures require different decisions prior to analysis.

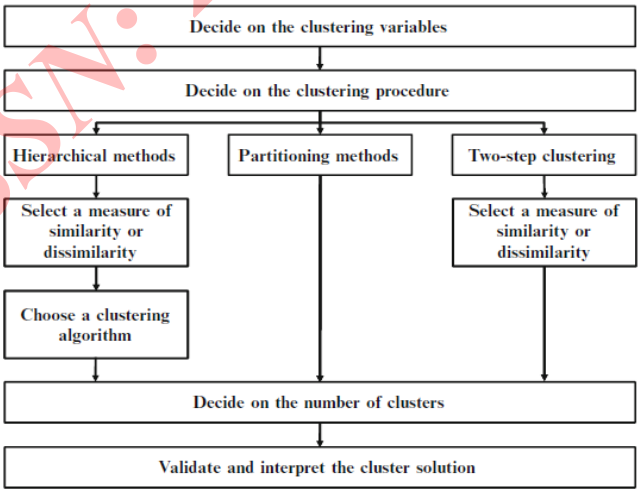


Fig 1.Steps in a Cluster Analysis

Hierarchical Clustering Algorithms

The hierarchical clustering is generally classified into two types of approach such as agglomerative approach and divisive approach. Agglomerative approach is the clustering technique in which bottom up strategy is used to cluster the objects.

It merges the atomic clusters into larger and larger until all the objects are merged into single cluster. Divisive approach is

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

the clustering technique in which top down strategy is used to cluster the objects. In this method the larger clusters are divided into smaller clusters until each object forms cluster of its own. Figure shows simple example of hierarchical clustering.

Hierarchical clustering proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. The result of the algorithm is a tree of clusters, called dendrogram, which shows how the clusters are related. By cutting the dendrogram at a desired level, a clustering of the data items into disjoint groups is obtained.

Hierarchical clustering procedures are characterized by the tree-like structure established in the course of the analysis. Most hierarchical techniques fall into a category called agglomerative clustering.

In this category, clusters are consecutively formed from objects. Initially, this type of procedure starts with each object representing an individual cluster.

These clusters are then sequentially merged according to their similarity. First, the two most similar clusters (i.e., those with the smallest distance between them) are merged to form a new cluster at the bottom of the hierarchy. In the next step, another pair of clusters is merged and linked to a higher level of the hierarchy, and so on.

This allows a hierarchy of clusters to be established from the bottom up. In Fig. we show how agglomerative clustering assigns additional objects to clusters as the cluster size increases. There are two type of hierarchical clustering:

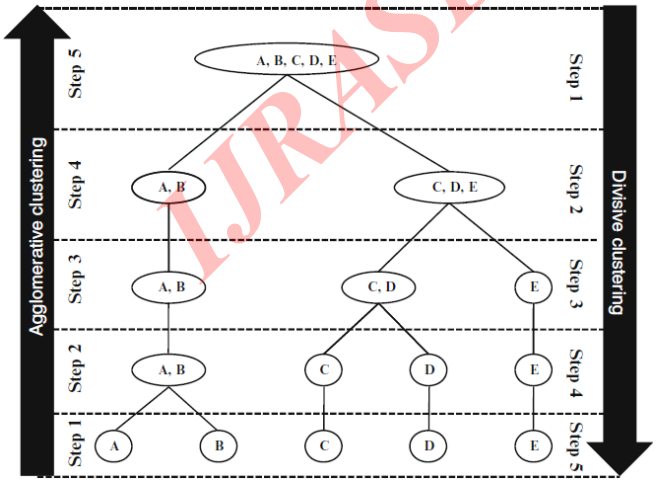


Fig 2. Agglomerative and Divisive Clustering

1. Agglomerative Clustering Algorithm

Compute the proximity matrix

Let each data point be a cluster

Repeat

Merge the two closest clusters

Update the proximity matrix

Until only a single cluster remains.

Key operation is the computation of the proximity of two clusters.

(In this paper we use Agglomerative Clustering Algorithm. This allows a hierarchy of clusters to be established from the bottom up.)

2. Divisive Hierarchical Clustering Algorithm

Compute a minimum spanning tree for the proximity graph.

Repeat

Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity).

until only singleton clusters remain

II. IMPLIMENTATION

"Rapid Miner is the world-wide leading open-source data mining solution due to the combination of its leading-edge technologies and its functional range. Applications of Rapid Miner cover a wide range of real-world data mining tasks." Real-world knowledge discovery processes typically consist of complex data preprocessing, machine learning, evaluation, and visualization steps. Hence a data mining platform should allow complex nested operator chains or trees, provide transparent data handling, comfortable parameter handling and optimization, be flexible, extendable and easy-to-use. Rapid Miner (formerly Yale) is an environment for machine learning and data mining processes.

ISSN: 2321-9653

Different Ways of Using Rapid Miner:

Rapid Miner can be started off-line, if the process configuration is provided as XML file. Alternatively, the GUI of Rapid Miner can be used to design the XML description of the operator tree, to interactively control and inspect running processes, and to continuously monitor the visualization of the process results. Break points can be used to check intermediate results and the data flow between operators. Of course you can also use Rapid Miner from your program. Clear interfaces define an easy way of applying single operators, operator chains, or complete operator trees on your input data. A command line version and a Java API allow invoking of Rapid Miner from your programs without using the GUI. Since Rapid Miner is entirely written in Java, it runs on any major platform/operating system

Multi-Layered Data View Concept

Rapid Miner's most important characteristic is the ability to nest operator chains and build complex operator trees. In order to support this characteristic the Rapid Miner data core acts like a data base management system and provide a multi-layered data view concept on a central data table which underlies all views. This multi-layered view concept is also an efficient way to store different views on the same data table. This is especially important for automatic data preprocessing tasks like feature generation or selection. No matter whether a data set is stored in memory, in a file, or in a database, Rapid Miner internally uses a special type of data table to represent it.

Transparent Data Handling

Rapid Miner supports flexible process (re)arrangements which allow the search for the best learning scheme and preprocessing for the data and learning task at hand. Rapid Miner achieves a transparent data handling by supporting several types of data sources and hiding internal data transformations and partitioning from the user. Due to the modular operator concept often only one operator has to be replaced to evaluate its performance while the rest of the process design remains the same. This is an important feature for both scientific research and the optimization of real-world applications.

Tool Used In Implementation

Tanagra was written as an aid to education and research on data mining by Ricco Rakotomalala On the main page of the Tanagra site, Rakotomalala outlines his intentions for the software. He intended Tanagra to be a free, open-source, user-friendly piece of software for students and researchers to mine their data. Tanagra simplifies this paradigm by restricting the graph to be a tree. This means that there can only be one parent to each node, and therefore only one data source for each operation. This is a limitation in Tanagra that will be further discussed later in this report. This report attempts to outline a number of the functionalities of Tanagra, as well as their shortcomings, and conclude with a final recommendation and general evaluation of the suitability of Tanagra for the usage of our fictional company tab

Compatibility with other formats

While Tanagra cannot directly read or write other, more complex formats, there is a conversion tool available on the Tanagra website to convert from ARFF files, the format used by Weka.

Method for Importing Data

Importing data in Tanagra is a relatively simple operation and is performed at the very beginning of a project due to the use of the stream diagram model. An import data dialog allows a user to select a text file on the local hard drive, and import it.

Visualization

Visualization of Data**:**

Tanagra allows a number of different options for visualizing data. A user may view the data directly in a table. Various scatter-plots and graphs can also be produced, but not with particular ease.

ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

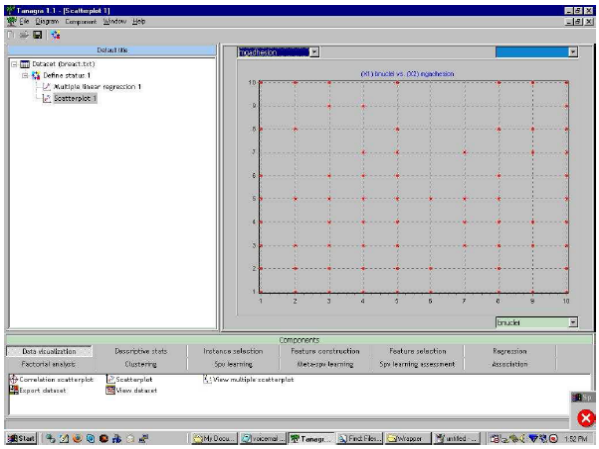


Fig 3. Visualization of Tanagra

It is sometimes difficult to tell which visualizations can be applied to a given data set. Most require few or no parameters, but must be applied as a sub-node not to the data set itself, but instead to a defined "Define Status" component that specifies the desired variables to view or analyses.

Visualization of Analysis Results:

Results of analyses in Tanagra are primarily visualized in a two dimensional graph. Color can be used to differentiate between a class attribute, or clustering or classification result. For data with high dimensionality, two dimensions can be selected for display.

Similarity Measure:

Before clustering, a similarity/distance measure must be determined. The measure reflects the degree of closeness or separation of the target objects and should correspond to the characteristics that are believed to distinguish the clusters embedded in the data. In many cases, these characteristics are dependent on the data or the problem context at hand, and there is no measure that is universally best for all kinds of clustering problems. Moreover, choosing an appropriate similarity measure is also crucial for cluster analysis, especially for a particular type of clustering algorithms.

Cosine Similarity

The cosine measure has been one of the most popular similarity measures due to its sensitivity to text vector pattern. The cosine measure computes the cosine of the angle between two feature vectors and is used frequently in text mining where vectors are very large but sparse. When documents are represented as term vectors, the similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, the so-called cosine similarity. Cosine similarity is one of the most popular similarity measure applied to text documents, such as in numerous information retrieval applications and clustering too. An important property of the cosine similarity is its independence of document length. For example, combining two identical copies of a document d` to get a new pseudo document d`, the cosine similarity between d and d` is 1, which means that these two documents are regarded to be identical. In other words, documents with the same composition but different totals will be treated identically.

Finding the similarity between documents using cosine similarity measure formula:

$$Sim_1(S_i, S_j) = \sum_{n-gram} \frac{2^* | S_i \cap S_j |}{| S_i | + | S_j |} \quad \text{.....}$$

Cosine Similarity measure by rapid miner tool:

pdf1	pdf2	similarity
1	2	0.028654
1	3	0.016676
1	4	0.09054
1	5	0.235681
1	6	0.068092
1	7	0.032895
1	8	0.052474
1	9	0.991174
1	10	0.024173
.	.	
.	.	
1	45	
.	.	
.	.	
n	m	

e.g.
1 to 2.....45
2 to 3.....45
.
.
34 to 45.....

Similarity Matrix Generation:

Firstly I have generated the similarity between 45 pdf dataset from cosine similarity measure formula in Rapid Miner tool. Then this matrix made into n x n format it means that 45 x 45 because I have 45 pdf data set and generate the matrix in n x n

ISSN: 2321-9653

from. This Matrix shows that the similarity between each document to every document. This is the pre step for applying algorithms, generate the clusters and validate them.

PDF`S	1	2	3	4	5	6	7	8	9	10	...	45
1	0	0.028654	0.016676	0.09054	0.235681	0.068092	0.032895	0.052474	0.991174	0.024173		
2	0.028654	0	0.028216	0.04278	0.177049	0.072692	0.075905	0.016362	0.025062	0.022709		
3	0.016676	0.028216	0	0.017701	0.041832	0.046038	0.115757	0.015843	0.012578	0.019553		
4	0.09054	0.04278	0.017701	0	0.094821	0.016497	0.05069	0.030335	0.088979	0.016308		
5	0.235681	0.177049	0.041832	0.094821	0	0.187466	0.10884	0.032802	0.221996	0.017759		
6	0.068092	0.072692	0.046038	0.016497	0.187466	0	0.085151	0.068344	0.056466	0.028574		
7	0.032895	0.075905	0.115757	0.05069	0.10884	0.085151	0	0.058304	0.028415	0.042577		
8	0.052474	0.016362	0.015843	0.030335	0.032802	0.068344	0.058304	0	0.053167	0.034712		
9	0.991174	0.025062	0.012578	0.088979	0.221996	0.056466	0.028415	0.053167	0	0.02683		
10	0.024173	0.022709	0.019553	0.016308	0.017759	0.028574	0.042577	0.034712	0.02683	0		
.												
.												
.												
45												

Table: Matrix Generation of 45 PDF`S dataset.

Validate the Clusters:

DSSC aims at providing soft clustering on a set of documents based on a given similarity measure. It has the following goals:

Enable soft clustering: documents can be clustered into multiple clusters.

Efficient: DSSC should be able to run faster than traditional hard clustering algorithms.

Cluster discovery: the algorithm should be able to find clusters that hard clustering algorithms cannot find.

Handle outliers: the algorithm should be robust against outliers.

DSSC requires a similarity measure between documents: that is, given two documents x and y, there is a function $0 \le f(x, y) \le 1$ which returns how similar x and y are. It also requires a number k, which denotes the number of clusters that the user is expecting. Note that DSSC can decide to produce a different number of clusters, depending on the input documents. DSSC produces a set of clusters at the end. Each cluster c is denoted by a set of documents called cluster centroids. The centroids serve two purposes: to define the set of documents most representative of the cluster, and to determine the degree of membership between c and each document. A measure m(c, x) is defined to represent how similar a document x is to cluster DSSC can be broadly divided into four steps: a pre-processing step to clean up and transform the data; an initial cluster generation step to initialize clusters and remove outliers; an iterative step to build clusters; and a post-processing step to present the results. Each step is described below:

1. Pre-processing**:**

In this step each document is transformed into a structure that will be used by the similarity function f(). One such representation is a vector, with each dimension denoting the presence/absence of a certain word in that document. In addition, we remove all the stop words (like articles, propositions and auxiliaries verbs) that are not helpful in Clustering.

2. Initial cluster generation**:**

At this step the input is analyzed, initial clusters are produced and outliers are removed. The first thing for DSSC to do is to decide what constitute as "similar" documents. Essentially, we need to find a threshold value λ such that two documents are considered similar if and only if $f(x, y) \ge \lambda$ Since DSSC is designed to adapt to different similarity measures f, it is not reasonable for the user to supply a value for λ. As a result, DSSC determines the appropriate value of λ based on the input documents.

3. Iterative step:

Cluster Cohesion: Measures how closely related are objects in a cluster.

$$WSS = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$

WSS stands for Within Sum of Squared Error.

Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters

$$BSS = \sum |C_i| (m - m_i)^2$$

– Where $|C_i|$ is the size of cluster i

BSS stands for Between Sum of Squared Error.

Example: Squared Error

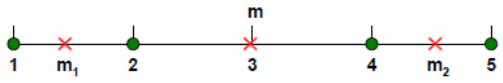
ISSN: 2321-9653

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

–Cohesion is measured by the within cluster sum of squares (SSE)

In this step, clusters are refined. Since DSSC uses cluster centroids as representative of each cluster, this step examines each cluster and decides whether the centroids should change.

- Example: SSE
  - BSS + WSS = constant



K=1 cluster:

$$WSS = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$
$$BSS = 4 \times (3-3)^2 = 0$$
$$Total = 10 + 0 = 10$$

K=2 clusters:

$$WSS = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$
$$BSS = 2 \times (3-1.5)^2 + 2 \times (4.5-3)^2 = 9$$
$$Total = 1 + 9 = 10$$

Different Aspects of Cluster Validation

Determining the clustering tendency of a set of data, i.e. distinguishing whether non-random structure actually exists in the data.

Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.

Evaluating how well the results of a cluster analysis fit the data without reference to external information.

- Use only the data.

Comparing the results of two different sets of cluster analyses to determine which is better.

Determining the 'correct' number of clusters. For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Final Comment on Cluster Validity:

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

III. EXPERIMENTAL & RESULTS

To evaluate the ability of our multilevel refinement algorithms to further improve the quality of a clustering, we used them to refine the solutions produced by the hierarchical clustering algorithm described that is based on the generalized group average model. We used five data sets to perform these comparisons. Two of these data sets consist of points in two dimensions and were synthetically generated, one was obtained from Reuter's newswire and the other two were obtained from the TREC collection of documents. For each one of the data sets we constructed an $n \times n$ similarity matrix, using techniques that are appropriate for the nature of each particular data set. Details on how the similarity matrices were constructed are presented along with the experimental results in the following sections. From each one of these five similarity matrices, a sparse graph representation was obtained by using the *k*-nearest neighbor graph approach . In all the experiments presented and we selected *k* to be equal to 10, and we studied the effectiveness of our refinement algorithms for different values of *k*. The hierarchical clustering algorithm that was used to obtained the clustering solutions as well as the clustering objective functions that are used by our refinement algorithms, require that we specify the value of the parameter that models the degree of inter-connectivity between the items in a cluster. In addition to the hierarchical clustering algorithm presented that operates on the sparse *k*-nearest neighbor graph, we also compare the quality of the clustering produced by our algorithms against two other algorithms.

IV. AIM AND OBJECTIVE:

Main objective of my thesis is to delivery of document using hierarchical clustering technique.

Increase document retrieval efficiency

Compare the set of cluster and validate the cluster using BSS/WSS technique

Solve outliers problem

Ranking the document according to their relevancy

And deliver to user.

V. CONCLUSIONS AND DIRECTIONS OF FUTURE RESEARCH

ISSN: 2321-9653

For the explosion of information in the World Wide Web, this thesis proposed a new method of summarization via soft clustering algorithm. It used Google search engine to extract relevant documents, and mixed query sentence into document set which segmented from multi-documents set, then this paper created efficient hierarchical clustering to cluster all the documents. Also, there are a lot of rooms for improvement. For example, readability is an important aspect in the performance of multi-document summarization. In future work, we will consider new soft cluster algorithm to more improve the efficiency of clustering.

REFERENCES

[1] King-Ip Lin, Ravikumar Kondadadi. A Similarity-Based Soft Clustering Algorithm for Documents. 7th International Conference on Database Systems for Advanced Applications (DASFAA '01). 2001.

[2] Dragomir R.Radev et al. Centroid-based summarization of multiple documents. Information Processing and Management: an International Journal, Vol. 40, pp. 919-38, 2004.

[3] Bharati M Ramager, "Data Mining techniques and Applications", "International Journal of Computer Science and Engineering Vol. 8", December 2009.

[4] Accessible from Sonali Agarwal, Neera Singh, Dr. G.N. Pandey, "Implementation of Data Mining and Data Warehouse in E-Governance", "International Journal of Computer Applications (IJCA) (0975-8887), Vol.9- No.4," November 2010.

[5] Calinski, T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat Theory Methods 3(1):1–27

[6] Ng, R. and Han, J.(1994). Efficient and Effective Clustering Methods for Spatial Data Mining. In Proceeding's of the 20th VLDB Conference, Santiago, Chile.

[7] D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Turkey. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the ACM SIGIR, 1992.

[8] Willet, Peter "Parallel Database Processing, Text Retrieval and Cluster Analyses" Pitman Publishing, London, 1990.

[9] JiaweiHan, MichelineKamber Data Mining Concepts and Techniques [M] Beijing: Mechanical Industry Press, 2005 185-218

[10] Jain, A.K., Murty M.N., and Flynn P.J. (1999): Data Clustering.