SENTENCE LEVEL TEXT CLUSTERING USING A FUZZY RELATIONAL CLUSTERING ALGORITHM

Sarika S. Musale[1], Jyoti Deshmukh[2]

[1,2]*Dept of Information Technology, Mumbai University, (India)*

ABSTRACT

Clustering is the process of grouping or aggregating of data items. Sentence clustering mainly used in variety of applications such as classify and categorization of documents, automatic summary generation, organizing the documents, etc. In text processing, sentence clustering plays a vital role this is used in text mining activities. Size of the clusters may change from one cluster to another. The traditional clustering algorithms have some problems in clustering the input dataset. The problems such as, instability of clusters, complexity and sensitivity. To overcome the drawbacks of these clustering algorithms, this paper proposes a algorithm called Fuzzy Relational Eigenvector Centrality-based Clustering Algorithm (FRECCA) is used for the clustering of sentences. In this algorithm single object may belong to more than one cluster.

Keywords: FRECCA, HFRECCA, Hierarchical Structure, Sentence Clustering.

I. OVERVIEW

We live in a world where vast amounts of data are collected daily. Analyzing such data is an important need. Data mining can meet this need by providing tools to discover knowledge from data. Data mining can be viewed as a result of the natural evolution of information technology.

Some of the most common data mining algorithms in use today [1]. We have broken the techniques into two sections as Classical Techniques: Statistics, Neighborhoods and Clustering Next Generation Techniques: Trees, Networks and Rules Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. The set of clusters resulting from a cluster analysis can be referred to as a clustering. In this context, different clustering methods may generate different clustering on the same data set. The partitioning is not performed by humans, but by the clustering algorithm. Hence, clustering is useful in that it can lead to the discovery of previously unknown groups within the data.

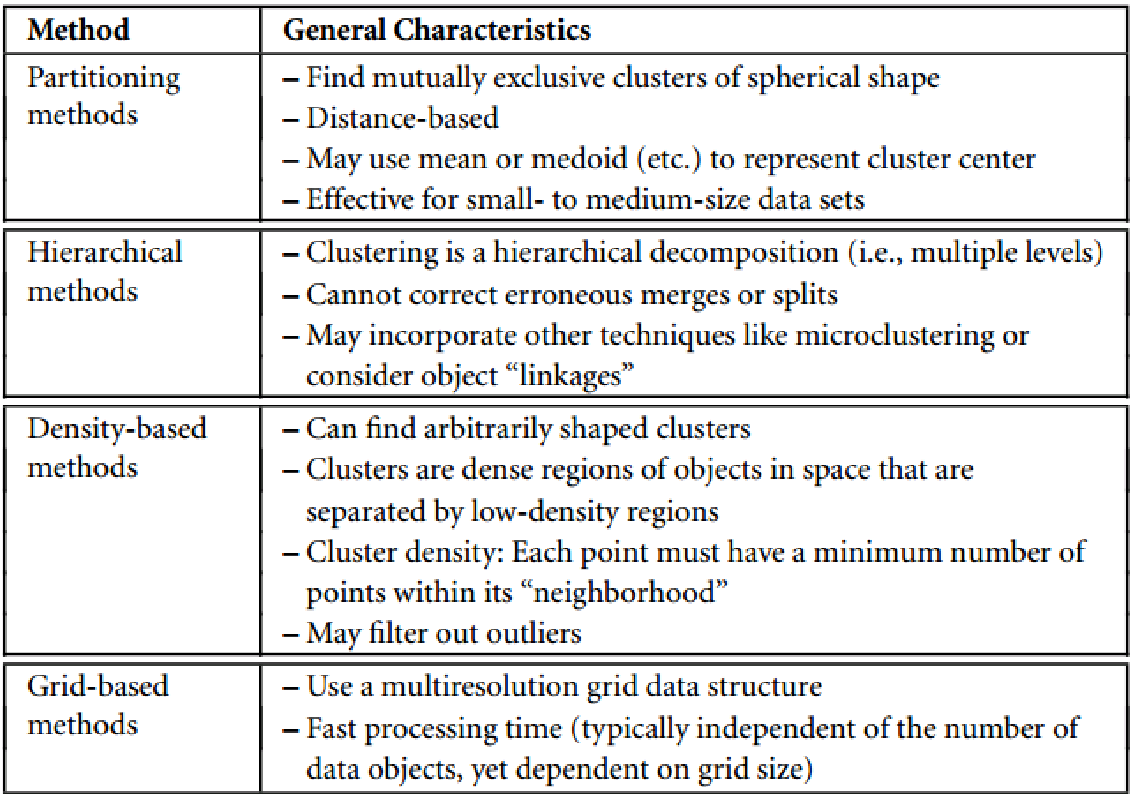
Cluster analysis has been widely used in many applications such as business intelligence, image pattern recognition, Web search, biology, and security[2]. In business intelligence, clustering can be used to organize a large number of customers into groups, where customers within a group share strong similar characteristics. This facilitates the development of business strategies for enhanced customer relationship management. Moreover, consider a consultant company with a large number of projects. To improve project management, clustering can be applied to partition projects into categories based on similarity so that project auditing and diagnosis can be conducted effectively.Clustering is a challenging research field. In this, you will learn about the requirements for clustering as a data mining tool, as well as aspects that can be used for comparing clustering methods.

II. REQUIREMENTS OF CLUSTERING IN DATA MINING

- Scalability: Many clustering algorithms work well on small data sets containing fewer than several hundred data objects.
- Ability to deal with different types of attributes: Many algorithms are designed to cluster numeric data. Recently, more and more applications need clustering techniques for complex data types such as graphs, sequences, images, and documents.
- Discovery of clusters with arbitrary shape: Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures.
- Ability to deal with noisy data: Most real-world data sets contain outliers and/or missing, unknown, or erroneous data. Clustering algorithms can be sensitive to such noise and may produce poor-quality clusters. Therefore, we need clustering methods that are robust to noise.
- Constraint-based clustering: Real-world applications may need to perform clustering under various kinds of constraints. A challenging task is to find data groups with good clustering behavior that satisfy specified constraints.
- Interpretability and usability: Users want clustering results to be interpretable, comprehensible, and usable. That is, clustering may need to be tied in with specific semantic interpretations and applications

III. REVIEW OF LITERATURE

There are many clustering algorithms in the literature. It is difficult to provide a crisp categorization of clustering methods because these categories may overlap so that a method may have features from several categories. Nevertheless, it is useful to present a relatively organized picture of clustering methods. In general, the major fundamental clustering methods can be classified into the following categories [2].



Method	General Characteristics
Partitioning methods	– Find mutually exclusive clusters of spherical shape – Distance-based – May use mean or medoid (etc.) to represent cluster center – Effective for small- to medium-size data sets
Hierarchical methods	– Clustering is a hierarchical decomposition (i.e., multiple levels) – Cannot correct erroneous merges or splits – May incorporate other techniques like microclustering or consider object "linkages"
Density-based methods	– Can find arbitrarily shaped clusters – Clusters are dense regions of objects in space that are separated by low-density regions – Cluster density: Each point must have a minimum number of points within its "neighborhood" – May filter out outliers
Grid-based methods	– Use a multiresolution grid data structure – Fast processing time (typically independent of the number of data objects, yet dependent on grid size)

Table 1 Clustering Methods

3.1 Partitioning Methods

The simplest and most fundamental version of cluster analysis is partitioning, which organizes the objects of a set into several exclusive groups or clusters. To keep the problem specification concise, we can assume that the number of clusters is given as background knowledge. The most well-known and commonly used partitioning methods—k-means and k-medoids.

k-means algorithm

A centroid-based partitioning technique uses the centroid of a cluster, Ci , to represent that cluster. Conceptually, the centroid of a cluster is its center point. The centroid can be defined in various ways such as by the mean or medoid of the objects (or points) assigned to the cluster[1].

```
Algorithm: k-means. The k-means algorithm for partitioning, where each cluster's center
               is represented by the mean value of the objects in the cluster.
Input:     k: the number of clusters,
            D: a data set containing n objects.
Output:    A set of k clusters.
Method:
           (1) Arbitrarily choose k objects from D as the initial cluster centers;
           (2) Repeat
           (3) (Re) assign each object to the cluster to which the object is the most similar,based on the mean
value of the objects in the cluster;
```

(4) Update the cluster means, that is, calculate the mean value of the objects for each cluster;

(5) Until no change;

IV. HIERARCHICAL METHODS

While partitioning methods meet the basic clustering requirement of organizing a set of objects into a number of exclusive groups, in some situations we may want to partition our data into groups at different levels such as in a hierarchy. A hierarchical clustering method works by grouping data objects into a hierarchy or "tree" of clusters. Representing data objects in the form of a hierarchy is useful for data summarization and visualization.

A hierarchical clustering method can be either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or topdown (splitting) fashion.

- **An agglomerative hierarchical clustering method** uses a bottom-up strategy. It typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied.
- **A divisive hierarchical clustering method** employs a top-down strategy. It starts by placing all objects in one cluster, which is the hierarchy's root. It then divides the root
  cluster into several smaller sub clusters, and recursively partitions those clusters into smaller ones. The partitioning process continues until each cluster at the lowest level is coherent enough

V. BIRCH: MULTIPHASE HIERARCHICAL CLUSTERING USING CLUSTERING FEATURE TREES

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is designed for clustering a large amount of numeric data by integrating hierarchical clustering.BIRCH uses the notions of clustering feature to summarize a cluster, and clustering feature tree (CF-tree) to represent a cluster hierarchy. These structures help the clustering method achieve good speed and scalability in large or even streaming databases, and also make it effective for incremental and dynamic clustering of incoming objects.

5.1 Density-Based Methods

Partitioning and hierarchical methods are designed to find spherical-shaped clusters. They have difficulty finding clusters of arbitrary shape such as the "S" shape and oval clusters. Given such data, they would likely inaccurately identify convex regions, where noise or outliers are included in the clusters. To find clusters of arbitrary shape, alternatively, we can model clusters as dense regions in the data space, separated by sparse regions. This is the main strategy behind density-based clustering methods, which can discover clusters of non spherical shape. The basic techniques of density-based clustering, three representative methods, namely, DBSCAN, OPTICS and DENCLUE [1].

5.2 DBSCAN: Density-Based Clustering Based on Connected Regions with High Density

"How can we find dense regions in density-based clustering?" The density of an object o can be measured by the number of objects close to o. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) finds core objects, that is, objects that have dense neighborhoods. It connects core objects and their neighborhoods to form dense regions as clusters. "How does DBSCAN quantify the neighborhood of an object?" A user-specified parameter e > 0 is used to specify the radius of a neighborhood we consider for every object. The e-neighborhood of an object o is the space within a radius e- centered at o.

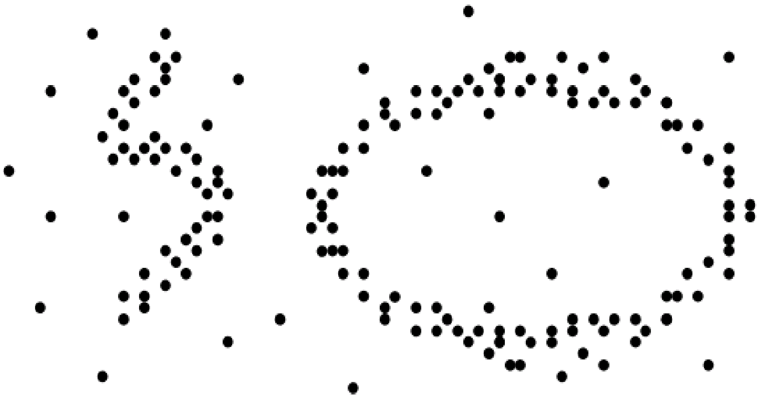


Fig.1 Clusters of arbitrary shape.

5.3 Evaluation of Clustering

Now we know what clustering is and know several popular clustering methods. We may ask, "When I try out a clustering method on a data set, how can I evaluate whether the clustering results are good?" In general, cluster evaluation assesses the feasibility of clustering analysis on a data set and the quality of the results generated by a clustering method. The major tasks of clustering evaluation include the following:

- Assessing clustering tendency. In this task, for a given data set, we assess whether a Non random structure exists in the data. Blindly applying a clustering method on a data set will return clusters; however, the clusters mined may be misleading. Clustering analysis on a data set is meaningful only when there is a nonrandom structure in the data.
- Determining the number of clusters in a data set. A few algorithms, such as k-means,require the number of clusters in a data set as the parameter. Moreover, the number of clusters can be regarded as an interesting and important summary statistic of a data set. Therefore, it is desirable to estimate this number even before a clustering algorithm is used to derive detailed clusters.
- Measuring clustering quality. After applying a clustering method on a data set, we

want to assess how good the resulting clusters are. A number of measures can be used. Some methods measure how well the clusters fit the data set, while others measure how well the clusters match the ground truth, if such truth is available. There are also measures that score clustering and thus can compare two sets of clustering results on the same data set.

5.4 Present Investigation

In many text processing activities, Sentence clustering plays an important role. For instance, various authors have argued that incorporating sentence clustering into extractive multi document summarization helps avoid problems of content overlap, leading to better coverage [3], [4], [5], [6]. On the other hand, sentence clustering can also be used within more general text mining tasks. For instance, regard as web mining [7].

5.5 Problem

Clustering is an extensively studied data mining problem in the text domains. The difficulty finds numerous applications in customer segmentation, classification, collaborative filtering, visualization, document organization, and indexing. In text mining, clustering the sentence is one of the processes and used within general text mining tasks. In hard clustering methods, a pattern belongs to a single cluster is necessary, which is difficult in sentence level clustering [8].**Solution**

Fuzzy clustering algorithms allow patterns to belong to all clusters with differing degrees of membership. This is important in domains such as sentence clustering, since a sentence is likely to be related to more than one theme or topic present within a document or set of documents. However, because most sentence similarity measures do not represent sentences in a common metric space, conventional fuzzy clustering approaches based on prototypes or mixtures of Gaussians are generally not applicable to sentence clustering. A novel fuzzy clustering algorithm that operates on relational input data; i.e., data in the form of a square matrix of pair wise similarities between data objects. The algorithm uses a graph representation of the data, and operates in an Expectation-Maximization framework in which the graph centrality of an object in the graph is interpreted as likelihood [8].

VI. IMPLEMENTATION

Irrespective of the specific task (e.g., summarization, text mining, etc.), most documents will contain interrelated topics or themes, and many sentences will be related to some degree to a number of these. The successfully being able to capture such fuzzy relationships will lead to an increase in the breadth and scope of problems to which sentence clustering can be applied. However, clustering text at the sentence level poses specific challenges not present when clustering larger segments of text, such as documents. We now highlight some important differences between clustering at these two levels, and examine some existing approaches to fuzzy clustering. Clustering text at the document level is well established in the Information Retrieval (IR) literature, where documents are typically represented as data points in a high dimensional vector space in which each dimension corresponds to a unique keyword, leading to a rectangular representation in which rows represent documents and columns represent attributes of those documents.

Fuzzy Relational Eigenvector Centrality-based Clustering Algorithm (FRECCA) is the proposed clustering algorithm. We first describe the use of Page Rank as a general graph centrality measure, and review the Gaussian mixture model approach. We then describe how Page Rank can be used within an Expectation-Maximization framework to construct a complete relational fuzzy clustering algorithm.

6.1 Graph-Based Centrality and PageRank

The basic idea behind the PageRank [9] algorithm is that the importance of a node within a graph can be determined by taking into account global information recursively computed from the entire graph, with connections to high-scoring nodes contributing more to the score of a node than connections to low-scoring nodes. It is this importance that can then be used as a measure of centrality. PageRank assigns to every node in a directed graph a numerical score between 0 and 1, known as its PageRank score (PR), and defined as

$$PR(V_i) = (1 - d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j)$$

Where $In(V_i)$ is the set of vertices that point to V_i, $Out(V_j)$ is the set of vertices pointed to by Vj, and d is a damping factor. Using the analogy of a random surfer, nodes visited more often will be those with many links coming in from other frequently visited nodes, and the role of d is to reserve some probability for jumping to any node in the graph, thereby preventing getting stuck in a disconnected part of the graph.

6.2 Mixture Models and the EM Algorithm

FRECCA is motivated by the mixture model approach, in which a density is modeled as a linear combination of C component densities p(x/m) in the form $\sum \pi_m p(\mathbf{x}|m)$, where the π_m are called mixing coefficients, and represent the prior probability of data point x having been generated from component m of the mixtu

6.3 Fuzzy Relational Clustering

Unlike Gaussian mixture models, which use a likelihood function parameterized by the means of the mixture components, the FRECCA algorithm uses the Page Rank score of an object within a cluster as a measure of its centrality to that cluster. These Page Rank values are then treated as likelihoods. Since there is no parameterized likelihood function as such, the only parameters that need to be determined are the cluster membership values and mixing coefficients. The algorithm uses Expectation Maximization to optimize these parameters. We assume in the following that the similarities between objects are stored in a similarity matrix S={sij}, where sij is the similarity between objects i and j.

Initialization. We assume here that cluster membership values are initialized randomly, and normalized such that cluster membership for an object sums to unity over all clusters. Mixing coefficients are initialized such that priors for all clusters are equal.

Expectation step. The E-step calculates the PageRank value for each object in each cluster. PageRank values for each cluster are calculated.

Maximization step. Since there is no parameterized likelihood function, the maximization step involves only the single step of updating the mixing coefficients based on membership values calculated in the Expectation Step.

VII. RESULTS AND DISCUSSIONS

7.1 Duplicate Clusters

The number of initial clusters must be specified as input to the algorithm. If this number is too high, then duplicate clusters (i.e., clusters with identical membership values across all objects) will be found. While it might appear at first sight that duplicate clusters can simply be removed after the algorithm has converged. The solution is to perform a check for duplicate clusters at the completion of each Maximization step. If duplicate clusters are found, membership values are renormalized, and the algorithm is allowed to proceed until a stage at which convergence has been achieved and no duplicate clusters exist.

7.2 Effect of Damping Factor, d2

The damping factor d that appears in the Page Rank calculation affects the fuzziness of the clustering, but generally does not affect the number of clusters, provided that the value is above approximately 0.8. In general, the higher the value of d, the harder is the clustering, with cluster membership values being close to either zero or one. We used a value of 0.85.

7.3 Thresholding of Similarity Values

Depending on the domain, the graph representing the relation between objects may be heavily or sparsely connected. In the case of sentence clustering, we have found that many of the similarities sij between sentences are very small. The clustering performance of the algorithm can be improved by thresholding these similarity values such that all values below the threshold are converted to zero.

7.4 Hard Clustering

The algorithm outputs cluster membership values p_i^m,which represent the degree of membership of object i to cluster m. If hard clustering is required, this can be trivially achieved by assigning a sentence to the cluster m for which membership is highest; i.e.$\arg\max_{m \in C}\{p_i^m\}$.

7.5 Future Scope

In fuzzy clustering algorithm that operates on relational input data; i.e., data in the form of a square matrix of pair-wise similarities between data objects. However, the major disadvantage of the Fuzzy Relational Eigenvector Centrality based Clustering Algorithm (FRECCA) is its time complexity. The FRECCA lies in its ability to identify fuzzy clusters, and if the objective is to perform only hard clustering.

Hierarchical fuzzy relational clustering algorithm that operates on relational input data; i.e., data in the form of a square matrix of pair –wise similarities between data objects. The algorithm uses a graph representation of the data, and operates in a Fuzzification Degree framework .Results of applying the algorithm is capable of identifying overlapping clusters of semantically related sentences, and that it is therefore of potential use in a variety of text mining tasks.

VIII. CONCLUSION

Cluster analysis divides data into meaningful or useful groups (clusters).If meaningful clusters are the goal, and then the resulting clusters should capture the “natural” structure of the data. For example, cluster analysis has been used to group related documents for browsing, to find genes and proteins that have similar functionality. However, in other cases, cluster analysis is only a useful starting point for other purposes, e.g., data compression or efficiently finding the nearest neighbors of points.

In comparison with hard clustering methods, in which a pattern belongs to a single cluster, fuzzy clustering algorithms allow patterns to belong to all clusters with differing degrees of membership. This is important in domains such as sentence clustering, since a sentence is likely to be related to more than one theme or topic present within a document or set of documents. However, because most sentence similarity measures do not represent sentences in a common metric space, conventional fuzzy clustering approaches based on prototypes or mixtures of Gaussians are generally not applicable to sentence clustering.

Acknowledgement

I sincerely feel that the credit of special topic report could not be narrowed down to only one individual. This work is an integrated effort of all those concern with it, through whose able cooperation and effective guidance I could achieve its completion.

I express my gratitude to Principal **V.C.Kotak** and Head of Department, for their generous appreciation, persistent and encouragement. I am gladly thankful to my guide **Ms. Swati Deshpande** and my co-guide **Ms.Jyoti Deshmukh** for being kind enough to spare their valuable time, provide insights and guidance on topic Sentence Level Text Clustering using a Fuzzy Relational Clustering Algorithm

I owe a special acknowledgment to them for giving me a lot of their time during the period of preparing this project. I could never had done it without their support, technical advice and suggestions, thorough reading of all the work. The blessing, help and guidance given by them time to time shall carry me a long way in the journey of life on which I am about to embark.

I would like to thank the Review Committee for their invaluable suggestions and feedback without which our work would have been very difficult. I am also thankful to the management and laboratory staff of SAKEC for their valuable support. Last but not least, I sincerely thank to those from teaching and non-teaching staff from SAKEC who were somehow attached with my endeavour.

REFERENCES

[1] *Excerpted from the book Building Data Mining Applications* for CRM by Alex Berson, Stephen Smith, and Kurt Thearling.

[2] *Data Mining: Concepts and Techniques* - Jiawei Han & Micheline Kamber

[3] V. Hatzivassiloglou, J.L. Klavans, M.L. Holcombe, R.Barzilay, M. Kan, and K.R. McKeown, “SIMFINDER: A Flexible Clustering Tool for Summarization,” Proc. NAACL Workshop Automatic Summarization, pp. 41-49, 2001.

[4] H. Zha, “Generic Summarization and Keyphrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering,” Proc. 25th Ann. Int’l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 113-120, 2002.

[5] D.R. Radev, H. Jing, M. Stys, and D. Tam, “Centroid-Based Summarization of Multiple Documents,” Information processing and Management: An Int’l J., vol. 40, pp. 919-938, 2004.

[6] R.M. Aliguyev, “A New Sentence Similarity Measure and Sentence Based Extractive Technique for AutomaticText Summarization,” Expert Systems with Applications, vol. 36, pp. 7764- 7772, 2009.

[7] R. Kosala and H. Blockeel, “Web Mining Research: A Survey,” ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 1-15, 2000.

[8]Clustering Sentence-Level Text Using a Novel Fuzzy Relational Clustering Algorithm Andrew Skabar, Member, IEEE, and Khaled Abdalgader

[9]S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," Computer Networks and ISDN Systems, vol. 30, pp. 107-117, 1998.

[10] Cluster-Centric Fuzzy Modeling Witold Pedrycz, Fellow, IEEE, and Hesam Izakian, Student Member, IEEE