

Graph based Clustering Techniques

Vaishali Pawar¹ and Radhika Chandwadkar²

¹⁻²MVPS's KBTCE Computer Engineering department, Nashik, India

Email: vaishalispawar88@gmail.com, radhikapchandwadkar@gmail.com

Abstract—Clustering is widely used technique to find hidden, meaningful patterns in the given dataset, called as clusters. It is grouping of data into groups of similar objects. The objective is that the objects within a group should be similar to one another and different from the objects in other groups. Clustering plays an excellent role in data mining applications, biology, science, pattern recognition, social networking, machine learning, web analysis, CRM, marketing, medical diagnostics, and many others. A variety of clustering algorithms have emerged that meet these requirements and were successfully applied to many data mining and pattern recognition problems. Due to their great representational power, Graphs can handle huge databases with large attributes. Hence Graphs play a vital role in clustering applications.

Index Terms— Clustering, data mining, machine learning, pattern recognition, social networking.

I. INTRODUCTION

Clustering is widely used in many disciplines. The typical applications include pattern [1], speech [4], and text recognition [3]. Machine learning clustering algorithms were applied to image segmentation [3] and computer vision [5]. Clustering is mainly used in applications that handle large datasets with large attributes. Clustering is more distinct or better if the similarity within a cluster and the difference between clusters is better. Clustering unavoidably loses fine details of data. But it achieves simplification in data representation. Intense efforts have been contributed by [6], [7], [8] and many others in information retrieval and text mining. There are many more application-specific developments in the field that are beyond our scope. Graphs are structures formed by a set of vertices (also called as nodes) and a set of edges that are connections between pairs of vertices. Due to their great representational power graphs are more widely used in applications that involve complex data. Graph clustering is the task of grouping the vertices of the graph into clusters with the consideration the edge structure of the graph in such a way that there should be many edges within each cluster and relatively few between the clusters. Graph based clustering techniques are popular in large attribute data applications like in data mining, wireless sensor networks, distributed systems etc.

In this study we attempt to analyze theoretically the existing graph based clustering algorithms and their classification. The rest of this paper is organized as follows: Section II explains the overall process of clustering and classification of clustering techniques. Section III explains role of Graph theory in clustering algorithms. Section IV introduces the Hierarchical clustering and Partitional Clustering algorithms. Finally, Section V concludes the paper.

II. CLUSTER ANALYSIS

Cluster analysis groups data objects based on information found in the data, which describes the objects and

their relationships.

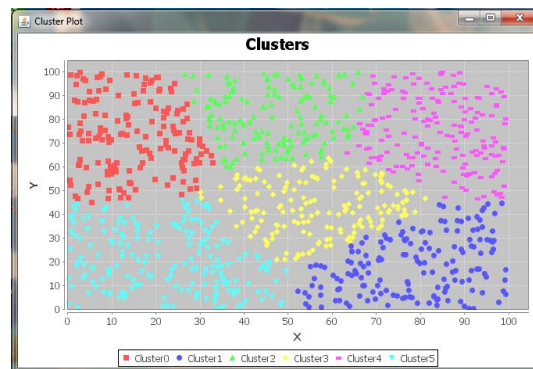


Fig. 1. Clustering the data

The Clustering process

Typical data clustering process involves the following steps [12].

- (1) Data representation
- (2) Definition of a data proximity function
- (3) Clustering or grouping,
- (4) Data abstraction and
- (5) Assessment of output

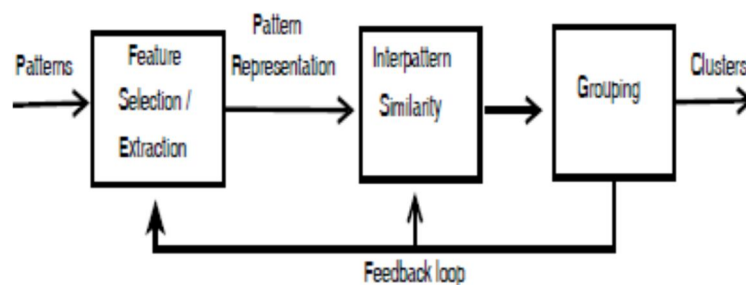


Fig. 2. Stages in clustering

Figure 2 shows typical steps in clustering. The grouping process output affects feature extraction and similarity computation function due to the feedback loop [10]. The number of available data patterns, their type, and scale of the features is available to the clustering algorithm. Pattern proximity function [11] measures the distance defined on pairs of patterns. A variety of distance measures are available. A simple distance measure like Euclidean distance can often be used to reflect similarity or dissimilarity between two patterns. The grouping step can be performed in a number of ways. The output clustering can be rigid or fuzzy. Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partitioned clustering algorithms identify the partition that optimizes a clustering criterion. Additional techniques for the grouping operation include probabilistic [16] and graph-theoretic [15] clustering methods. In the clustering context, a typical data abstraction is a compact description of each cluster, usually in terms of cluster prototypes or representative patterns such as the centroid [14].

A. Different ways of clustering

An entire collection of clusters is commonly referred to as a clustering, and in this section, we differentiate various types of clustering [14],[15]. Different approaches to clustering data can be described with the help of the hierarchy shown in Figure 3. At the top level, there is a distinction between hierarchical and partitional approaches.

Hierarchical methods produce a nested series of partitions, while partitioned methods produce only one. A partitional clustering is simply a division of the set of data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset. If we permit clusters to have sub clusters, then we obtain a hierarchical clustering, which is a set of nested clusters that are organized as a tree.

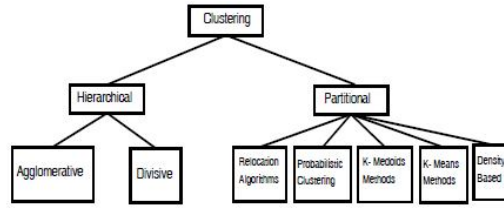


Fig. 3. Classification of clustering methods

III. GRAPH BASED CLUSTERING

A graph $G = (V, E, W)$ is a set of vertex set V , edge set E and set of non negative weights W . Let $X = \{X_1, \dots, X_n\}$ be a set of data points. $S = \{S_{ij}\}$ where $i, j = 1 \dots N$ be the similarity matrix in which each element indicates the similarity $S_{ij} \geq 0$ between two data points X_i and X_j . A nice way to represent the data is to construct a graph on which each vertex represents a data point and the edge weight carries the similarity of two vertices. The clustering problem in graph perspective is then formulated as partitioning the graph into subgraphs such that the edges in the same subgraph have high weights and the edges between different subgraphs have low weights. The partition of vertices of the graph into non-overlapping clusters should follow following conditions [15]. Let $C = \{C_1, C_N\}$ be the partition of V vertices such that

- (1) $C_i \neq \emptyset$ for $i \in \{1, \dots, N\}$
- (2) $C_i \cap C_j = \emptyset$ for $i, j \in \{1, \dots, N\}$ and $i \neq j$
- (3) $C_1 \cup C_2 \cup \dots \cup C_N = V$

From the theoretical aspect a graph can be partitioned into connected subgraphs. Keeping this in mind, the general methodology of graph-based clustering can be best described by following factors.

- (1) *Graph Modelling*: It deals with the problem of representing data using appropriate graph or modelling the real application as a graph.
- (2) *Distance Measure*: The graph clustering method should select appropriate distance measure that can improve the quality of a clustering.
- (3) *Algorithm*: The clustering algorithm should exactly or approximately optimize the quality measure.
- (4) *Evaluation*: Various metrics can be used to evaluate the performance of clustering by comparing with a baseline clustering.

IV. CLUSTERING ALGORITHMS

Based on the classification given in section II, roughly the algorithms can be categorized as Hierarchical and Partitional algorithms.

A. Hierarchical clustering algorithms

Hierarchical clustering builds a cluster hierarchy, which is a tree of clusters, called as a dendrogram[21]. Every cluster node contains child clusters. The sibling clusters partition the points covered by their common parent. Such an approach allows exploring data on different levels of granularity. Hierarchical clustering methods are classified as agglomerative and divisive methods [17].

1) Agglomerative and Divisive clustering:

An agglomerative clustering starts with one-point clusters and recursively merges two or more most appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. The process continues until an end criterion is achieved.

a) Single linkage Agglomerative clustering:

One of the simplest agglomerative hierarchical clustering methods is single linkage, also known as the nearest neighbor technique. The distance between groups is defined as the distance between the closest pair of objects, where only pairs consisting of one object from each group are considered. In the single linkage

method, $D(r, s)$ [12] is computed as

$$D(r, s) = \text{Min} (d (i, j))$$

where object i is in cluster r and object j is cluster s . The overall time complexity of single-link clustering is $O(n^2)$ [13].

A drawback of this method is that the clusters may be forced together due to single elements being close to each other, even though many of the elements in each cluster may be very distant to each other. This is also called as chaining phenomenon [13].

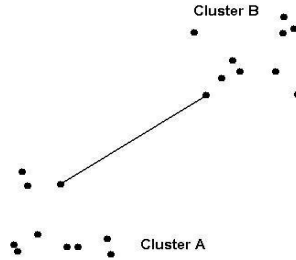


Fig. 4. Single linkage Agglomerative clustering

b) Complete linkage Agglomerative clustering:

The complete linkage is also known as farthest neighbor clustering method. The distance between groups is defined as the distance between the most distant pair of objects, one from each group.

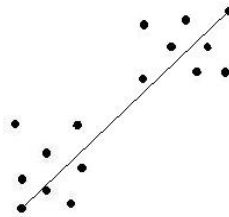


Fig. 5. Complete linkage Agglomerative clustering

In the complete linkage method, $D(r, s)$ is computed as

$$D(r, s) = \text{Max} (d (i, j))$$

where object i is in cluster r and object j is in cluster s and the maximum value of these distances is said to be the distance between clusters r and s .

c) Average linkage Agglomerative Clustering :

Here the distance between two clusters is defined as the average of distances between all pairs of objects, where each pair is made up of one object from each group [12]. The linkage function specifying the distance between two clusters is computed as the average distance between objects from the first cluster and objects from the second cluster. Based on recursive computation of cohesion, the time complexity of average-link clustering is $O(n^2 \log n)$ [13].

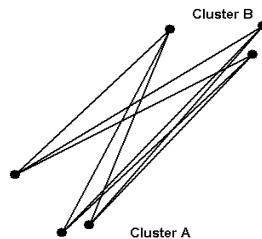


Fig. 6. Average linkage Agglomerative clustering

In the average linkage method, $D(r, s)$ is computed as

$$D(r, s) = \frac{T_{rs}}{(N_r * N_s)}$$

where T_{rs} is the sum of all pair wise distances between cluster r and cluster s . N_r and N_s are the sizes of the clusters r and s respectively.

B. Partitioned clustering algorithms

As the name suggests, partitioning clustering algorithms divide data into several subsets. As checking all possible subset systems is computationally expensive, certain greedy heuristics are used in the form of iterative optimization. Different relocation schemes that iteratively reassign points between the k clusters are applied. In traditional hierarchical methods, clusters are not revisited after being constructed. But in Partitioned algorithms, relocation algorithms gradually improve clusters [9]. This results in formation of high quality clusters. One approach to data partitioning is to take a conceptual point of view that identifies the cluster with a certain model whose unknown parameters have to be found. The concise cluster representation allows concise definition of intra-clusters measures. This ultimately gives rise to a global objective function [19]. The K-medoids and K-means methods are iterative variations of Partitioned algorithms. This approach starts with the definition of objective function depending on a partition. In iterative algorithms, pair-wise computations like linkage computing would be much expensive. But using unique cluster representative like medoids solves the problem. K-medoid is the most appropriate data point within a cluster that represents it. Representation by k-medoids has two advantages. First, it presents no limitations on attributes types. Second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster. So the algorithm is lesser sensitive to the presence of outer samples. In K-means cluster is represented by its centroid. Centroid is a mean (usually weighted average) of points within a cluster. These algorithms work effectively with numerical attributes of the dataset.

1) *K-Medoids Methods:* In K-medoids methods a cluster is represented by one of its points called as a medoid. It is an object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal [20]. The medoids have in-built resistance against outer samples as peripheral cluster samples do not affect them. After selection of initial medoids, clusters are defined as subsets of points close to respective medoids. The objective function is defined as the averaged distance or another dissimilarity measure between a point and its medoid. Two early versions of K-medoid methods are the algorithm PAM [18](Partitioning Around Medoids) and the algorithm CLARA [18](Clustering LARge Applications). PAM is iterative optimization algorithm. It combines relocation of points between perspective clusters with re-nomination of the medoids.

2) Compared to other partitioning methods such as PAM, CLARA can deal with much larger datasets [18]. Internally, this is achieved by considering sub-datasets of fixed size, so that the time and storage requirements become linear rather than quadratic. Each sub-dataset is partitioned into k clusters using the same algorithm as in the PAM function. Once k representative objects have been selected from the sub-dataset, each object of the entire dataset is assigned to the nearest medoid. The sum of the dissimilarities of the objects to their closest medoid is used as a measure of the quality of the clustering. The sub dataset, for which the sum is minimal, is retained. A further analysis is carried out on the final partition. Each sub-dataset is forced to contain the medoids obtained from the best sub dataset until then. Randomly drawn objects are added to this set until sample size has been reached.

3) The further step in progress is related to [2] who introduced the algorithm CLARANS (Clustering Large Applications based upon RANdomized Search) in the context of clustering in spatial databases. Reference [2] considers a graph whose nodes are the sets of k medoids and an edge connects two nodes if they differ by exactly one medoid. While CLARA [18] compares very few neighbours corresponding to a fixed small sample. CLARANS uses random search to generate neighbours by starting with an arbitrary node and randomly checking max neighbours. If a neighbour represents a better partition, the process continues with this new node. Otherwise a local minimum is found, and the algorithm restarts until numlocal local minima are found. The complexity of CLARANS is $O(n^2)$ in terms of number of points

2) *K-Means Methods:* The K-means is the most popular clustering algorithm used in scientific and computer applications. The mean C of its points, called as centroid, represents each cluster. The algorithm has good geometric and statistical sense for numerical attributes. But this does not work well with categorical attributes. The wide popularity of K-means algorithm is well deserved. The space requirements for K-means are modest because only the data points and centroids are stored. Specifically, the storage required is $O((m + K)n)$, where m is the number of points and n is the number of attributes[14]. The time requirements for K-means are also modest. It is basically linear in the number of data points. In particular, the time required is $O(I * K * m * n)$

n), where I is the number of iterations required for convergence. As mentioned, I is often small and can usually be safely bounded, as most changes typically occur in the first few iterations. Therefore, K-means is linear in m , the number of points and is efficient as well as simple provided that K , the number of clusters, is significantly less than m . The K-means algorithm also suffers from certain drawbacks. Its results strongly depend on the initial nomination of centroids. There is no fixed criterion to decide what is a good K to use [14]. Also the process is sensitive with respect to outliers. As the local optimum is known to be a far from the global one the algorithm lacks scalability. As a result of these disadvantages, resulting clusters may be unbalanced.

The bisecting K-means algorithm [11] is a straightforward extension of the basic K-means algorithm that is based on a simple idea. The theme is to obtain K clusters, split the set of all points into two clusters, select one of these clusters to split, and so on, until K clusters have been produced. There are a number of different ways to choose which cluster to split. We can choose the largest cluster at each step. Different choices result in different clusters.

V. CONCLUSION

In this paper we have seen an overview of some of the essential concepts and techniques of clustering and the significance of graph theory in cluster analysis. In general, many of the good measures of clustering are intertwined. The field of graph clustering seems to be revolving around fundamentally similar definitions, although some of the initial steps for the algorithms are quite far apart. The task of clustering is highly application specific. Most of the application areas need to cluster high dimensional data with complex relationship. Graph theory can be incorporated in such applications. The weighted graphs, directed graphs, minimum spanning tree, bipartite graphs, cliques etc. are useful concepts of graph theory that play a vital role in pattern representation and finding centroid / medoid of the graph clusters. Data transformation, information networks, biological and sociological networks, pattern recognition are the key fields which need clustering using graph theory.

REFERENCES

- [1] Kumar Dhiraj, Santanu Kumar Rath, "Comparison of SGA and RGA based Clustering Algorithm for Pattern Recognition," *Inter-national Journal of Recent Trends in Engineering*, Vol 1, No. 1, pp. 269-273, May 2009, 269-273
- [2] T. Raymond, H. Jiawei, "CLARANS: A Method for Clustering Objects for Spatial Data Mining," *IEEE Transactions on Knowledge And Data Engineering*, Vol. 14, pp. 1003-1016, 2002
- [3] Muhammad Muzzamil Luqman, Thierry Brouard, Jean-Yves Ramel and Josep Lladós, "A Content Spotting System For Line Drawing Graphic Document Images", *IEEE, International Conference on Pattern Recognition*, pp. 3420-3423, 2010
- [4] Masaki Naito, Li Dengb, Yoshinori Sagisakaa, "Speaker clustering for speech recognition using vocal tract parameters", *Speech Communication*, Volume 36, Issues 3-4, pp. 305-315, March 2002
- [5] A. Revathi, R. Ganapathy and Y. Venkataramani, "Text Independent Speaker Recognition and Speaker Independent Speech Recognition Using Iterative Clustering Approach" *International Journal of Computer Science Information Technology (IJCSIT)*, Vol 1, No 2, pp. 30-42, November 2009
- [6] Yoshinobu Hotta, Satoshi Naoi, Misako Suwa, "Improvement of numeral recognition using personal handwriting characteristics based on clustering", *Systems and Computers in Japan*, Volume 33, Issue 7, pp. 1041-113, June 2002
- [7] A. K. Jain, P. J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys*, Vol. 31, pp. 264-323, 1999
- [8] A. K. Jain, P. J. Flynn, "Image segmentation using clustering", *Advances in Image understanding: A festschrift for Azriel Rosenfeld*, IEEE Computer Society Press, 65-83, 1999
- [9] Cutting, Karger, Pedersen, Tukey, "Scatter/gather: a clusterbased approach to browsing large document collection", *Proceedings of the 15th ACM SIGIR Conference*, Vol. 8, pp. 318-329, 1992
- [10] Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning", *Proceedings of the 7th ACM SIGKDD*, pp. 269-274, 2001
- [11] M. Steinbach, G. Karypis, V. Kumar, "A comparison of document clustering techniques", *ACM SIGKDD, World Text Mining Conference*, 2000
- [12] Guojun Gan, Chaogun Ma, Jianhong Wu, "Data Clustering: Theory, Algorithms, and Applications", ASA-SIAM Series on Statistics and Applied Probability 20, 2007
- [13] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schtze, "Introduction to Information Retrieval", Cambridge University Press, 2008.
- [14] A. K. Jain, R. C. Dubes, "Algorithms for Clustering Data", Prentice-Hall advanced reference series, 1988
- [15] Zheng Chen, Heng Ji, "Graph-based Clustering for Computational Linguistics: A Survey", *Proceedings of TextGraphs-5 Workshop on Graph-based Methods for Natural Language Processing*, pp. 1-9, July 2010
- [16] V. L. Brailovsky, "A probabilistic approach to clustering", *Pattern Recognition Letters*, Volume 12, Issue 4, 193-

198,1991

- [17] C. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters", *IEEE Transactions on Computers*, Vol. C-20, pp. 68-86,1971
- [18] Kaufman L., Rousseeuw P.J., "*Finding Groups in Data: An Introduction to Cluster Analysis*", Wiley, New York, 1990
- [19] V. K. Panchal, Harish Kundra, Jagdeep Kaur, "Comparative Study of Particle Swarm Optimization based Unsupervised Clustering Techniques", *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.10, pp 132- 140 ,October 2009
- [20] B. Pardeshi, D. Toshniwal, "Improved k-medoids clustering based on cluster validity index and object density " *IEEE 2nd International Conference on Advance Computing(IACC)*,pp. 379-384, Feb. 2010
- [21] Nicola Fanizzi, Claudia dAmato, Floriana Esposito, "A Hierarchical Clustering Procedure for Semantically Annotated Resources" *Proceedings of AI*IA 2007: Artificial Intelligence and Human- Oriented Computing 10th Congress of the Italian Association for Artificial Intelligence*,Italy, Rome, pp. 266-277, September 2007.