INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X

Impact factor: 4.295

(Volume 4, Issue 4)

Available online at: www.ijariit.com

Data clustering algorithms: A second look

Yousef Alraba'nah
yrabanah@zu.edu.jo
Zarqa University, Zarqa, Jordan

Mohammed Al-refai
refai@zu.edu.jo
Zarqa University, Zarqa, Jordan

ABSTRACT

With the huge volume of digital data, clustering algorithms are providing efficient tools for data organizing and analyzing. Clustering algorithms are used in various domains such as bioinformatics, speech recognition, and information retrieval. Clustering is an automatic technique that divides a set of data objects into smaller groups such that the objects within a group are similar to each other and dissimilar to objects in other groups as much as possible. This paper reviews and discusses different clustering algorithms, their concepts, advantages, and limitations. A comparison among clustering algorithms will also be represented based on certain criteria.

Keywords—*Data clustering, Clustering analysis, Hierarchical, Partitioning, Density-Based and Grid-Based*

1. INTRODUCTION

The last two decades have seen the rapid increase in the volume of digital data. Recent development in the information systems including a huge number of web documents, digital libraries, repositories, emails, and articles have been growing dramatically and heightened the need for efficient and effective approaches to organizing and managing it. Such large data usually makes it difficult to browse and use traditional analytical techniques [1]. Developing systematic approaches for organizing the data into meaningful forms would be very helpful and provided an excellent chance to produce controllable data. Clustering is an important process that organizes a large collection of data into smaller groups, called clusters, of coherent and similar objects. The objects within a cluster seemed to be similar to each other as possible, while they should be dissimilar from objects within other clusters. In other words, the clustering aims to maximize the similarity among the objects of one cluster (intra-cluster), as well as minimize it with the objects of other clusters (inter-cluster). The result of the clustering process is a set of heterogeneous clusters with homogeneous content [2,3].

Splitting large data into a smaller number of groups makes the data objects accessing and browsing more efficient and relatively easy, and improve the searching performance [4]. From a machine learning perspectives, clustering is unsupervised learning where there are no class labels for the model training (unlabeled data), and we do not have prior knowledge about the exact number of clusters [5]. There is a number of clustering methods types including (and not limited to): Hierarchical, partitioning, density based, and grid-based clustering methods. In this paper, we will discuss the clustering process and list the main types of clustering methods. Moreover, we will highlight and compare the most commonly used clustering algorithms.

2. HIERARCHICAL CLUSTERING METHOD

In this clustering method, the clusters are represented as a hierarchy of a dendrogram or tree, where the root node comprises the entire data and each leaf node represents an object. The intermediate nodes represent clusters that group object-object, object-cluster, or cluster-cluster based on the levels of the hierarchy [6]. The hierarchical algorithms can be categorized based on the decomposition approach to agglomerative hierarchal clustering, and divisive hierarchal clustering [2]. The agglomerative approach, which is also known as a bottom-up approach, starts by considering each data object as a separate cluster, and gradually agglomerates the proximal clusters until all objects belong finally to the same cluster. For clusters merging process, the most popular methods are single linkage and complete linkage. The single linkage method considers the shortest distance between any pair of objects in two clusters, which is also known as the nearest neighbor method. On the contrary, the complete linkage method uses the farthest distance. Several methods also can be used such as all-pairs linkage, and centroid linkage [2,7].

On the other hand, the divisive approach, which is also known as a top-down approach, starts by considering entire data objects belong to a single cluster and successfully divides it into more smaller clusters until eventually, each cluster contains only one object [8]. Some of the well-known hierarchical algorithms include Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) [9], Clustering Using Representatives (CURE) [10], Robust Clustering Algorithm for Categorical attributes (Rock) [11], and Chameleon [12].

BIRCH algorithm consists of constructing an in-memory Clustering Feature tree (CF tree), where the tree is constructed while incrementally scanning data objects. The algorithm is then run to cluster the leaf nodes when a new data object is encountered, the tree is traversed starting from the root until leaf nodes, and successful CF comparing processes are done to determine whether the new object should be assigned to an existing cluster, or a new cluster should be formed. Birch algorithm has the advantage of scalability, where the algorithm can deal with very large datasets. The time complexity of BIRCH algorithm is O(n). However, the limitation of BIRCH comes from the fact that, the algorithm uses the concept of radius or diameter to control the cluster boundaries and consequently it may not work well when clusters are not spherical [9].

CURE algorithm was proposed to solve the problem of non-spherical clusters shapes. CURE algorithm works as follows: each cluster is represented by a fixed number of well-scattered objects and then shrunk them toward the cluster center by a certain function. In the next step, and based on their representative objects, the algorithm successively merges the closest pair of clusters together to form a new cluster. Having multiple representative objects for each cluster gives CURE the ability to work well with the non-spherical cluster shapes. In order to reduce the complexity, CURE uses random sampling and partitioning strategy to cluster the samples partially and gradually integrated them. The time complexity of CURE algorithm is O(n^2 log n) [10].

Rock algorithm was proposed as an improvement of CURE. The main idea behind ROCK was the link strategy which used as a measure to cluster objects based on their common neighbors. ROCK algorithm utilizes the random sampling strategy as in CURE. The advantage of ROCK algorithm over other algorithms is the ability to handle data with qualitative type. The time complexity of ROCK algorithm is O(n^2 log n) [11]. The chameleon algorithm uses the k-nearest neighbor graph to cluster objects, where an object has links only with its k-nearest neighbor and links with other objects are eliminated. The data objects divided firstly into smaller size clusters and then the clusters with small size are combined with larger size. The time complexity of the Chameleon algorithm is O(n^2) [12].

3. PARTITIONING CLUSTERING METHOD

Partition method splits data objects into a number of partitions (clusters) in a way such that each object is assigned to exactly one cluster, and each cluster must contain at least one object. Unlike hierarchical clustering algorithms, the partitioning clustering algorithms produce no overlapping clusters [13]. K-means [14] and K-medoids [15] are the two most famous partitioning algorithms.

K-means algorithm starts by randomly selecting K objects as representative of clusters (clusters centers). Afterward, all remaining objects are assigned to the cluster with the closest center. The cluster center is then updated by computing the mean of all objects belonging to that cluster. This process repeats until no change made on the clusters centers. The k-means algorithm is very simple and has good computational speed which therefore can be used to solve practical problems. The time complexity of the k-means algorithm is O(n k d), where k is the number of clusters, and d is the number of attributes [14, 16]. K-medoids algorithm was proposed as a variant of k-means. K-medoids algorithm is suitable for data objects with discrete values. Rather than using the mean of data objects as a representative of a cluster as in k-means, k-medoids uses an actual data object which is most centrally located as a representative to its cluster. The main drawback of k-means is the sensitivity to outliers (noisy objects) which come from the fact that a mean is affected directly by extreme noisy objects. Therefore, K-medoids is more robust to noises and outliers than k-means. Moreover, the k-medoids algorithm is more nature than k-means when regarding data objects types. Means computation requires data type to be continuous, while medoids always exist and do not require computations if the means are unavailable. The time complexity of the k-medoids algorithm is O(k(n-k $)^2$) [15].

4. DENSITY-BASED CLUSTERING METHOD

The core idea in this type of clustering is to construct clusters based on the density of data objects [2]. Density refers to the number of objects in a certain region of data objects. Data objects located in a region with high density are assumed to belong to the same cluster. A cluster is expanded in the direction of the density of data objects, and therefore density-based clustering algorithms can deal with clusters of arbitrary shapes. Moreover, such algorithms have no worry about outliers [7]. Some of the well-known algorithms of this type are Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [17] and Ordering Points To Identify the Clustering Structure (OPTICS) [18]. DBSCAN algorithm groups object into clusters based on dense regions. A region is said to be dense if a neighborhood of an object contains at least a certain a minimum number of neighboring objects. Data objects are divided logically into core object, border object, and noise object. Core objects are objects that have at least the required number of neighboring objects. Border objects are objects within the neighborhood of a core object but have neighboring objects fewer than the required. Noise objects are neither core or border objects whose neighboring objects are too far away. The algorithm selects an object and constructs a cluster if that object is a core object by absorbing all objects in its neighborhood. More than one core object can form a cluster if they are neighbors to each other. The time complexity of the DBSCAN algorithm is O(n log n) [17]. OPTICS algorithm was proposed as an extension of the DBSCAN algorithm. OPTICS uses the same concepts of DBSCAN. DBSCAN algorithm has a shortcoming in clustering data objects with varying density. OPTICS improves DBSCAN by ordering data objects linearly based on their spatial closeness, so that closest objects become neighbors in the ordering. The time complexity of OPTICS is similar to DBSCAN which is O(n log n) [18].

5. GRID-BASED CLUSTERING METHOD

Such method creates a grid structure by splitting the data objects into a number of cells. Clusters are constructed based on the grid structure, where the data objects are assigned to appropriate grid cells. A cell that contains more than a specified number of objects are treated as a dense cell. Afterward, the dense neighbor cells are combined together to form clusters.

The advantage of this type of algorithm is its fast processing time as the clustering operations are performed on the grid rather than the data objects directly [2, 7]. Famous algorithms belong to the grid-based clustering are statistical Information Grid (STING) algorithm [19] and CLustering In QUEst (CLIQUE) algorithm [20]. STING algorithm divides data objects into several layers of rectangular cells which in turn form a hierarchical grid structure. Cells in a higher layer are divided into a number of cells of the next lower layer. Data objects within layers are clustered based on the regions of relevant cells. Statistics information such as mean, maximum, minimum and distribution of values are used to cluster objects. STING algorithm is used in clustering of spatial data objects and has a short running time, where its time complexity is O(n).

Table 1: Clustering Algorithms Comparison

Method	Algorithm	Scalability	Cluster shape	Outliers sensitivity	Time complexity
Hierarchical	BIRCH	Active	Spherical	No	O(n)
	CURE	Active	Arbitrary	No	$O(n^2 \log n)$
	ROCK	Moderate	Arbitrary	No	$O(n^2 \log n)$
	Chameleon	Active	Arbitrary	No	$O(n^2)$
Partitioning	K-means	Moderate	Spherical	Yes	O(n k d)
	K-medoids	Passive	Spherical	No	$O(k(n-k)^2)$
Density-based	DBSCAN	Moderate	Arbitrary	No	O(n log n)
	OPTICS	Moderate	Arbitrary	No	O(n log n)
Grid-based	STING	Active	Arbitrary	No	O(n)
	CLIQUE	Active	Spherical	Yes	$O(n+d^2)$

However, the algorithm is sensitive to the granularity of the lowest layer of the structure. Thus, the clustering quality will be affected. CLIQUE algorithm is efficient in clustering high dimensional data objects. The algorithm starts by partitioning each dimension of data objects into grids structure and then finds regions with high density in single dimension space. The dense regions are used to generate clusters in the subspaces. Afterward, the dense subspaces are examined to generate clusters for higher dimensions. CLIQUE algorithm is scalable which scales linearly with the number of objects and has low time complexity $O(n+d^2)$.

6. SUMMARY AND COMPARISON

In this section, we compare the discussed algorithms. Different criteria used in the comparison including time complexity, scalability of the algorithms, and cluster shapes. Table-1 summarizes the comparison.

7. CONCLUSION

Clustering algorithms play a vital role in analyzing and processing large data objects. Many clustering algorithms were and being proposed. In this paper, we introduce and discuss the concepts and ideas of the most commonly used clustering algorithms, as well as we present a comparison of these algorithm based on several criteria. Although the comparison is presented, however, there is no algorithm that can globally be used to cluster objects, as algorithms are proposed and designed under certain assumptions and for specific applications.

8. ACKNOWLEDGMENT

This research is funded by the Deanship of Research in Zarqa University /Jordan.

9. REFERENCES

[1] Huang, Anna. "Similarity measures for text document clustering." Proceedings of the sixth New Zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand. 2008.

[2] Aggarwal, Charu C., and Chandan K. Reddy, eds. Data clustering: algorithms and applications. CRC Press, 2013.

[3] Murtagh, Fionn, and Pedro Contreras. "Algorithms for hierarchical clustering: an overview." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2.1 (2012): 86-97.

[4] Wang, Haixun, and Jian Pei. "Clustering by pattern similarity." Journal of Computer Science and Technology 23.4(2008):481-496.

[5] Kim, Wooyoung. "Parallel clustering algorithms: A survey." Parallel Algorithms, Spring (2009).

[6] Bouguettaya, Athman, et al. "Efficient agglomerative hierarchical clustering." Expert Systems with Applications 42.5 (2015): 2785-2797.

[7] Xu, Rui, and Donald Wunsch. "Survey of clustering algorithms." IEEE Transactions on neural networks 16.3 (2005): 645-678.

[8] Li, Shaoning, Wenjing Li, and Jia Qiu. "A novel divisive hierarchical clustering algorithm for geospatial analysis." ISPRS International Journal of Geo-Information 6.1 (2017): 30.

[9] Zhang, Tian, Raghu Ramakrishnan, and Miron Livny. "BIRCH: A new data clustering algorithm and its applications." Data Mining and Knowledge Discovery 1.2 (1997): 141-182.

[10] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "Cure: an efficient clustering algorithm for large databases." Information systems 26.1 (2001): 35-58.

[11] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "ROCK: A robust clustering algorithm for categorical attributes." Data Engineering, 1999. Proceedings., 15th International Conference on. IEEE, 1999.

[12] Karypis, George, Eui-Hong Han, and Vipin Kumar. "Chameleon: Hierarchical clustering using dynamic modeling." Computer 32.8 (1999): 68-75.

[13] Celebi, M. Emre, ed. Partitional clustering algorithms. Springer, 2014.

[14] MacQueen, James. "Some methods for classification and analysis of multivariate observations." Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. No. 14. 1967.

[15] Park, Hae-Sang, and Chi-Hyuck Jun. "A simple and fast algorithm for K-medoids clustering." Expert systems with applications 36.2 (2009): 3336-3341.

[16] Burkardt, John. "K-means clustering." Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics (2009).

[17] Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." Kdd. Vol. 96. No. 34. 1996.

[18] Ankerst, Michael, et al. "OPTICS: ordering points to identify the clustering structure." ACM Sigmod record. Vol. 28. No. 2. ACM, 1999.

[19] Wang, Wei, Jiong Yang, and Richard Muntz. "STING: A statistical information grid approach to spatial data mining." VLDB. Vol. 97. 1997.

[20] Agrawal, Rakesh, et al. Automatic subspace clustering of high dimensional data for data mining applications. Vol. 27. No. 2. ACM, 1998.