Multi Angle Analysis of The Existing Clustering Algorithms

Jinzhen Ping [1], Qian Wang [2], Lili Yu [3], XueFang Wu [4]

Department of Software Engineering Shijiazhuang Information Engineering Vocational College

41472796@qq.com, 76192918@qq.com, 25734476@qq.com, 30081406@qq.com

Keywords: data mining; clustering; algorithm

Abstract. Data mining clustering is a broad research field. It is used to partition the data set of clusters. Different clustering methods use different similarity definition and technology. Several popular clustering algorithms are analyzed from three different perspectives: the clustering criterion, clustering algorithm and frame representation. Furthermore, some new construction algorithm, mixed or generalization of some algorithm were introduced. As a result of the analysis of several points of view, it can be covered and distinguished from most existing algorithms. It is based on self tuning algorithm and clustering benchmark.

Introduction

Clustering is an important data-mining technique used to find data segmentation and pattern information. Clustering technique is widely used in applications of financial data classification, spatial data processing, satellite photo analysis, and medical figure auto-detection etc. The problem of clustering is to partition the data set into segments (called clusters) so that intra-cluster data are similar and inter-cluster data are dissimilar.

Criteria

***Distance-Based clustering*.** The basic idea of distance-based clustering is that a cluster is the data points close to each other. The distance between two data points is easy to define in Euclidean space. The widely used distance definitions include Euclidean distance, and Manhattan distance. However, there are several choices for similarity definition between two sets of data points, as follows:

$$\mathrm{Similarity}_{\mathrm{rep}}(C_i, C_j) = \mathrm{distance}(\mathrm{rep}_i, \mathrm{rep}_j) \quad (1)$$

$$\mathrm{Similarity}_{\mathrm{avg}}(C_i, C_j) = \sum \mathrm{distance}(v_i, v_j) \quad (2)$$

$$\mathrm{Similarity}_{\mathrm{max}}(C_i, C_j) = \max\{\mathrm{distance}(v_i, v_j) \;\; v_i \in C_i, \;\; v_j \in C_j\} \quad (3)$$

$$\mathrm{Similarity}_{\mathrm{min}}(C_i, C_j) = \min\{\mathrm{distance}(v_i, v_j) \;\; v_i \in C_i, \;\; v_j \in C_j\} \quad (4)$$

In (1), repi and repj are representatives of Ci and Cj, respectively. The representative of a data set is usually the mean, such as in k-means. Single representative methods usually employ Definition (1). It is obvious that the complexity of (2), (3), and (4) are all O (|Ci|*|Cj|), which are inefficient for large data sets. Although they are more global definitions, they are usually not directly applied on similarity definition for sub-clusters or clusters. The only exception is BIRCH, in which CF-vector and CF-tree are employed to accelerate the computation. Some trade-off approaches are taken, as it will be discussed in Section III, in which the detailed analysis of single representative methods is also given. The advantage of distance-based clustering is that distance is easy for computing and understanding. And distance-based clustering algorithms usually need parameters of K, which is the number of final clusters user wants, or the minimum distance to distinguish two clusters. However, the disadvantage of them is also distinct that they are noise-sensitive. Although some techniques are introduced in some of them, they result in other serious problems. CURE uses representative shrinking techniques to reduce the impact of noises. This shortcoming counteracts the advantage of multi-representatives that the algorithm can identify arbitrary-shaped clusters. BIRCH, which is the first clustering algorithm considering noises, introduces a new parameter T, which is substantially a parameter related to density.

Furthermore, it is hard for user to understand this parameter unless the page storage ability of CF-tree is known (Page size/entry size/T is an approximation of density in that page). In addition, it may cause loss of small clusters and long-shaped clusters. Since lack of space, the detailed discussion is omitted here.

Density-Based Clustering. Other than distance-based clustering methods, density-based clustering stands for that clusters are dense areas. Therefore, the similarity definition of data points is based on whether they belong to connected dense regions. The data points belonging to the connected dense region belong to the same cluster. Based on the different computation of density, density-based clustering can be further classified into Nearest-Neighbor (called NN in the rest of this paper) methods and cell-based methods. The difference between them is that the former define density based on data set, and the latter define it based on data space. No matter which kind a density-based clustering algorithm belongs to, it always needs a parameter of minimum-density threshold, which is the key to define dense region.

NN Methods. NN methods only treat points, which have more than k neighbors in hyper-sphere whose radius is å, as data points in clusters. Since the neighbors of each point should be counted, the index structures which support region query, such as R*-tree, or X-tree, are always employed. Because of the curse of dimensionality, these methods don't have good scalability for dimensionality. Furthermore, NN methods will result in frequent I/O when the data sets are very large. However, for most multi-dimensional data sets, these methods are efficient. In short, the shortcoming of this kind of methods is the shortcoming of the index structures they based-on.

Traditional NN methods, such as DBSCAN and its descendants, need parameters of density threshold and Recently, OPTICS, whose basic idea is the same as DBSCAN, focuses on automatically identification of cluster structures. Since the novel techniques in OPTICS do not belong to the topic of this sub-section, we will discuss them in Section V.

Cell-Based Methods. Cell-based methods count density information based on the units. STING, Wave Cluster, DBCLASD, CLIQUE, and OptiGrid all fall into this category. Cell-based methods have the shortcoming that cells are only approximation of dense areas. Some methods introduce techniques to solve this problem, as will be introduced in Section III Density-based clustering methods all meet problem when data sets contain clusters or sub-clusters whose granularity is smaller than the granularity of units for computing density. A well-known example is the dumbbell-shaped clusters, as shown in our experimental result. However, for density-based clustering methods, it is easy to remove noises, if the parameters are properly set. That is to say, it is robust to noises.

Linkage-Based Clustering. Other than distance-based or density-based clustering, linkage-based clustering can be applied to arbitrary metric spaces. Furthermore, since in high-dimensional space, the distance information and density information is not sufficient for clustering, linkage-based clustering is often employed.

Linkage-based methods are based on graph or hyper-graph model. They usually map the data set into a graph/hyper-graph, then cluster the data points based on the edge/hyper-edge information, so that the highly connected data points are assigned to the same cluster. The difference between graph model and hyper-graph model is that the former reflects the similarity of pair of nodes, while the latter usually reflects the co-occurrence information. ROCK and CHAMELEON use graph model, while ARHP, PDDP, STIRR, and CACTUS use hyper-graph model. Although the developers of CACTUS didn't state that it is a hyper-graph-model-based algorithm, it belongs to that kind. The quality of linkage-based clustering result depends on the definition of link or hyper-edge. Since it is impossible to handle a complete graph, the graph/hyper-graph model always eliminates the edges/hyper-edges whose weight is low, so that the graph/hyper-graph is sparse. However, to gain the efficiency, it may reduce the accuracy.

The algorithms fall in this category use different frameworks. ROCK and CHAMELEON are hierarchical clustering methods, while ARHP is divisive method, and STIRR uses dynamical system model. Furthermore, since the co-occurrence problem is similar to association rule mining problem, ARHP and CACTUS both borrow Apriority algorithm to find the clusters. Another algorithm

employ Apriority-like algorithm is CLIQUE.

Cluster Representation

The purpose of clustering is to identify the data clusters, which are the summary of the similar data. Each algorithm should represent the clusters and sub -clusters in some forms. Although labeling each data point with a cluster identity is a straightforward idea, most methods don't employ this approach. This may be because that: (1) The summary, which should be easily understandable, is more than (data-point, cluster-id) pairs; (2) It is time- and space-expensive to label all the data points in the process of clustering; (3) Some methods employ accurate compact cluster representatives, which make the time-consuming process of labeling unnecessary. We classify the cluster representation techniques into four kinds, as discussed in the following:

Representative Points. Most distance-based clustering methods use some points to represent clusters. These points are called representative points. The representatives may be data points, or some other points that do not exist in database, such as means of some sets of data points. The data representation techniques falling into this category can be further classified into three classes:

Single Representative. The simplest approach is to use one point as the representative of each cluster. Each data point is assigned to the cluster whose representative is the closest one. The representative point may be the mean of the cluster, like k-means methods do, or the data point in the database, which is the closest point to the center, like k-methods methods do.

The shortcoming of single representative approach is obvious: (1) only sphere clusters can be identified; and (2) large clusters with small cluster beside will be split, while some data points in the large cluster will be assigned to the small cluster. Therefore, this approach will fail when processing data sets with arbitrary shaped clusters or clusters with great difference.

All data Points. Using all the data points in a cluster to represent it is another straightforward approach. However, it is time-expensive since: (1) the data sets are always large so that the label information cannot fit in memory, which leads to frequent disk access, and (2) while computing information intra- and inter- clusters, it will access all data points. Furthermore, the label information is hard to understand. Therefore, no popular algorithms take this approach.

Algorithm Framework

In the above two sections, we discussed the clustering criteria and cluster representation, which are the two most important factors for clustering effectiveness. In this section, the algorithm framework will be discussed. The algorithm framework determines the time complexity of the algorithms, and the needed parameters. Furthermore, algorithm framework also affects the techniques of preprocessing. These are the focuses in the following three subsections.

Optimization Methods. Optimization methods usually try to optimize a certain measure. Traditional optimization methods are also known as partitioning methods. The most famous ones include k-means (including its variance k-modes, k-prototypes), and k-medoids (including PAM, CLARA, CLARANS, etc.). Some new built algorithms also fall into this category, including STIRR.

Other than k-means, k-medoids methods use data points to represent a cluster. Since noises or outliers less influence the medoids, they are more robust than k-means. However, the cost of k-medoids algorithms is also expensive. PAM, CLARA, and CLARANS are three most famous k-medoids algorithms. PAM is the first k-medoids method. CLARA and CLARANS both use sampling technique, in which CLARA use fixed samples, while CLARANS don't. Furthermore, CLARANS exploits randomized search. Therefore, CLARANS is more scalable than PAM and CLARA.

Other than k-means or k-medoids, some new built optimization algorithms don't use representatives, such as STIRR. STIRR is designed to handle categorical data, so that means or medoids is difficult to define. It maps the data set into a hyper-graph and then employs dynamical system techniques to find basins, which are fix-points of the system. Therefore, it can be viewed as

the process of finding an optimum of the system configuration.

Agglomerate Method. Agglomerate algorithms treat data points or data set partitions as sub-clusters in the beginning. Then they merge the sub-clusters iteratively until the final clusters are gotten. BIRCH, CURE, ISAAC, ROCK, STING, CHAMELEON, all fall into this category.

The agglomerate methods have the shortcoming that the time complexity is at least O (n2). Therefore, several techniques are employed to accelerate the processing. Since the number of the merge operations depends on the number of initial objects, some preprocessing techniques are used to reduce the object to be processed. Sampling and partitioning are two widely used preprocessing techniques. The developers of CURE proved that a small sample could guarantee the quality of clustering, while CURE, STING, CHAMELEON all use partitioning before merging the sub-clusters. Another technique used to accelerate the processing is indexing. Nearly all agglomerate algorithms exploit special index structure. BIRCH uses CF-tree, CURE uses k-d-tree and heap, ROCK uses two-level heap, STING uses quad-tree-like index, and CHAMELEON uses k-d-tree and heap-based priority queue.

Divisive Methods. Divisive methods belong to hierarchical methods as agglomerate methods do. Divisive methods begin with a large cluster, which contains all the data points, and then partition the cluster based on the dissimilarity recursively, until some stop condition is reached. ARHP, PDDP, and OptiGrid fall into this category. ARHP uses hyper-graph model. The whole data set is mapped to a hyper-graph by using association rule discovery techniques first. Then, the sub-graphs satisfy that the fitness is larger than a threshold is partitioned out.

Conclusion

In this paper, we try to analyze the existing popular clustering algorithms both theoretically and experimentally from three different viewpoints: clustering criteria, cluster representation, and algorithm framework, so that most algorithms can be covered, and distinguished. This work can be the basis of: (1) Clustering algorithm advantage/disadvantage analysis; (2) Clustering algorithm selection for data mining users; (3) Clustering algorithm auto-selection for different data sets; (4) Self-tuning clustering algorithm development; (5) Clustering benchmark construction. The analysis shows that most current algorithms have its shortcomings while being effective or efficient for some special characteristic data sets.

References

[1] Fasulo, D. An analysis of recent work on clustering algorithms. Technical Report, Department of Computer Science andEngineering, University of Washington, 2014.

[2] Sprenger, T.C., Brunella, R., Gross, M.H. H-BLOB: a hierarchical visual clustering method using implicit surfaces. TechnicalReport No.341, Computer Science Department, ETH Zürich, 2012.

[3] Han Jiawei, Kamber Micheline. Data Mining: Concepts and Techniques[M] . San Fransisco : Magan Kaufmann Publishers, 2006.

[4] Wilkinson B, Allen M. Parallel Programming : Techniques and Applications Using Netw or ked Workstatio ns and Parallel Computers[M] . New Jersey : Prentice Hall, 2014

[5] Brualdi RA. Introductory Combinator ics[M] .New Jersey : Pr entice Hall, 2012.