Clustering Techniques: A Brief Survey of Different Clustering Algorithms

Deepti Sisodia
Technocrates Institute of Technology, Bhopal, India

Lokesh Singh
Technocrates Institute of Technology, Bhopal, India

Sheetal Sisodia
Samrat Ashoka Technological Institute, Vidisha, India

Khushboo saxena
Technocrates Institute of Technology, Bhopal, India

Abstract - Partitioning a set of objects into homogeneous clusters is a fundamental operation in data mining. The operation is needed in a number of data mining tasks. Clustering or data grouping is the key technique of the data mining. It is an unsupervised learning task where one seeks to identify a finite set of categories termed clusters to describe the data . The grouping of data into clusters is based on the principle of maximizing the intra class similarity and minimizing the inter class similarity. The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? This paper deal with the study of various clustering algorithms of data mining and it focus on the clustering basics, requirement, classification, problem and application area of the clustering algorithms.

I. INTRODUCTION

Clustering[1,2] is an unsupervised learning task where one seeks to identify a finite set of categories termed clusters to describe the data .Unlike classification that analyses class-labeled instances, clustering has no training stage, and is usually used when the classes are not known in advance. A similarity metric is defined between items of data, and then similar items are grouped together to form clusters. Often, the attributes providing the best clustering should be identified as well. The grouping of data into clusters is based on the principle of maximizing the intra class similarity and minimizing the inter class similarity. Properties about the clusters can be analyzed to determine cluster profiles, which distinguish one cluster from another.

A good clustering method[1,2] will produce high quality clusters with high intra-class similarity - Similar to one another within the same cluster low inter-class similarity - Dissimilar to the objects in other clusters The quality of a clustering result depends on both the similarity measure used by the method and its implementation. The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

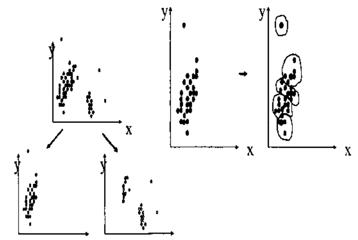

Figure.1:Data partitioning & clustering

II. Classification of Clustering Algorithms

Clustering algorithms can be broadly classified[4] into three categories, in the following subsections together with specific algorithms:

2.1 Partitioning
2.2 Hierarchical
2.3 Density-based

In short, partitioning algorithms attempt to determine k clusters that optimize a certain, often distance-based criterion function. Hierarchical algorithms create a hierarchical decomposition of the database that can be presented as a dendrogram. Density-based algorithms search for dense regions in the data space that are separated from one another by low density noise regions.

2.1. Partitioning Clustering Algorithms

Partitioning clustering attempts to decompose a set of N objects into k clusters such that the partitions optimize a certain criterion function. Each cluster is represented by the centre of gravity (or centroid) of the cluster, e.g. k-means, or by the closest instance to the gravity centre (or medoid), e.g. k-medoids. Typically, k seeds are randomly selected and then a relocation scheme iteratively reassigns points between clusters to optimize the clustering criterion. The minimization of the square-error criterion - sum of squared Euclidean distances of points from their closest cluster centroid, is the most commonly used. A serious drawback of partitioning algorithms is that there are a number of possible solutions. In particular, the number of all possible partitions P (n, k) that can be derived by partitioning n patterns into k clusters is :

$$P(n,k) = \frac{1}{k!}\sum_{i=1}^{k}(-1)^{k-1}\binom{k}{i}(i)^{n}$$

2.1.1 K-Means

K-means is perhaps the most popular clustering method in metric spaces. Initially k cluster centroids[3,7] are selected at random; k-means then reassigns all the points to their nearest centroids and recomputed centroids of the newly assembled groups. The iterative relocation continues until the criterion function, e.g. square-error converges. Despite its wide popularity, k-means is very sensitive to noise and outliers since a small number of such data can substantially influence the centroids. Other weaknesses are sensitivity to initialization, entrapments into local optima, poor cluster descriptors, inability to deal with clusters of arbitrary shape, size and density, reliance on user to specify the number of clusters.

Finally, this algorithm aims at minimizing an objective function; in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\| x_i^{(j)} - c_j \right\|^2 ,$$

where || xi(j) – cj ||2 is a chosen distance measure between a data point xi(j)and the cluster centre Cj, is an indicator of the distance of the n data points from their respective cluster centres.

The algorithm steps are

- Choose the number of clusters, *k*.
- Randomly generate *k* clusters and determine the cluster centers, or directly generate *k* random points as cluster centers.
- Assign each point to the nearest cluster center.
- Recompute the new cluster centers.
- Repeat the two previous steps until some convergence criterion is met (usually that the assignment hasn't changed).

2.1.2 Fuzzy k-means clustering

In fuzzy clustering, each point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the center of cluster. For each point x we have a coefficient giving the degree of being in the kth cluster uk(x). Usually, the sum of those coefficients is defined to be

$$\forall x \quad \sum_{k=1}^{\text{num. clusters}} u_k(x) \ = 1.$$

With fuzzy k-means, the centroid of a cluster is the mean of all points, weighted by their degree of belonging to the cluster:

$$\mathrm{center}_k = \frac{\sum_x u_k(x)^m x}{\sum_x u_k(x)^m}.$$

The degree of belonging is related to the inverse of the distance to the cluster center:

$$u_k(x) = \frac{1}{d(\mathrm{center}_k, x)},$$

then the coefficients are normalized and fuzzyfied with a real parameter $m > 1$ so that their sum is 1. So

$$u_k(x) = \frac{1}{\sum_j \left(\frac{d(\mathrm{center}_k, x)}{d(\mathrm{center}_j, x)}\right)^{2/(m-1)}}.$$

For m equal to 2, this is equivalent to normalising the coefficient linearly to make their sum 1. When m is close to 1, then cluster center closest to the point is given much more weight than the others, and the algorithm is similar to k-means.
The fuzzy k-means algorithm is very similar to the k-means algorithm:

- Choose a number of clusters.
- Assign randomly to each point coefficients for being in the clusters.
- Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than ε, the given sensitivity threshold) :

Compute the centroid for each cluster, using the formula above.
For each point, compute its coefficients of being in the clusters, using the formula above.
The algorithm minimizes intra-cluster variance as well, but has the same problems as k-means, the minimum is a local minimum, and the results depend on the initial choice of weights. The Expectation-maximization algorithm is a more statistically formalized method which includes some of these ideas: partial membership in classes. It has better convergence properties and is in general preferred to fuzzy-k-means.

2.1.3 K-Medoids

Unlike k-means, in the k-medoids or Partitioning Around Medoids (PAM)[1,2]method a cluster is represented by its medoid that is the most centrally located object (pattern) in the cluster . Medoids are more resistant to outliers and noise compared to centroids. PAM begins by selecting randomly an object as medoid for each of the k clusters. Then, each of the non-selected objects is grouped with the medoid to which it is the most similar. PAM then iteratively replaces one of the medoids by one of the non-medoids objects yielding the greatest improvement in the cost function. Clearly, PAM is an expensive algorithm as regards finding the medoids, as it compares each medoid with the entire dataset at each iteration of the algorithm.

2.1.4 Clustering Large Applications based on Randomized Search -CLARANS

CLARANS combines the sampling techniques with PAM. The clustering process can be presented as searching a graph where every node is a potential solution, that is, a set of k-medoids. The clustering obtained after replacing a medoid is called the neighbor of the current clustering. CLARANS [13]selects a node and compares it to a user-defined number of neighbors searching for a local minimum. If a better neighbor is found having lower square error, CLARANS moves to the neighbor's node and the process starts again; otherwise the current clustering is a local optimum. If the local optimum is found, CLARANS starts with a new randomly selected node in search for a new local optimum.

The advantages and disadvantages of partitioning clustering methods are:

Advantages

1. Relatively scalable and simple.
2. Suitable for datasets with compact spherical clusters that are well-separated

I. Disadvantages

1. Severe effectiveness degradation in high dimensional spaces as almost all pairs of points are about as far away as average; the concept of distance between points in high dimensional spaces is ill-defined
2. Poor cluster descriptors
3. Reliance on the user to specify the number of clusters in advance
4. High sensitivity to initialization phase, noise and outliers
5. Frequent entrapments into local optima
6. Inability to deal with non-convex clusters of varying size and density.

2.2. Hierarchical Algorithms

Unlike partitioning methods that create a single partition, hierarchical algorithms[11] produce a nested sequence (or dendrogram) of clusters, with a single all-inclusive cluster at the top and singleton clusters of individual points at the bottom. The hierarchy can be formed in top-down (divisive) or bottom-up (agglomerative) fashion and need not necessarily be extended to the extremes. The merging or splitting stops once the desired number of clusters has been formed. Typically, each iteration involves merging or splitting a pair of clusters based on a certain criterion, often measuring the proximity between clusters. Hierarchical techniques suffer from the fact that previously taken steps (merge or split), possibly erroneous, are irreversible. Some representative examples are:

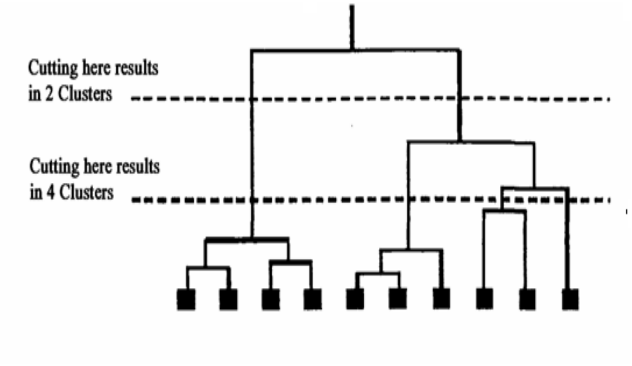
Cutting here results in 2 Clusters
Cutting here results in 4 Clusters

Figure 2: Hierarchical Clustering

2.2.1 CURE

Clustering Using Representatives (CURE)[9] is an agglomerative method introducing two novelties. First, clusters are represented by a fixed number of well-scattered points instead of a single centroid. Second, the representatives are shrunk toward their cluster centers by a constant factor. At each iteration, the pair of clusters with the closest representatives is merged. The use of multiple representatives allows CURE to deal with arbitrary-shaped clusters of different sizes, while the shrinking dampens the effects of outliers and noise. CURE uses a combination of random sampling and partitioning to improve scalability.

2.2.2 CHAMELEON

CHAMELEON [11] improves the clustering quality by using more elaborate merging criteria compared to CURE. Initially, a graph containing links between each point and its k-nearest neighbors is created. Then a graph-partitioning algorithm recursively splits the graph into many small-unconnected sub-graphs. During the second phase, each sub-graph is treated as an initial sub-cluster and an agglomerative hierarchical algorithm repeatedly combines the two most similar clusters. Two clusters are eligible for merging only if the resultant cluster has similar inter-connectivity and closeness to the two individual clusters before merging. Due to its dynamic merging model CHAMELEON is more effective than CURE in discovering arbitrary-shaped clusters of varying density. However, the improved effectiveness comes at the expense of computational cost that is quadratic in the database size.

2.2.3 BIRCH

Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) [8] introduces a novel hierarchical data structure, CF-tree, for compressing the data into many small sub-clusters and then performs clustering with these summaries rather than the raw data. Sub-clusters are represented by compact summaries, called cluster-features (CF) that are stored in the leafs. The non-leaf nodes store the sums of the CF of their children. A CF-tree is built dynamically and incrementally, requiring a single scan of the dataset. An object is inserted in the closest leaf entry. Two input parameters control the maximum number of children per non-leaf node and the maximum diameter of sub-clusters stored in the leafs. By varying these parameters, BIRCH can create a structure that fits in main memory. Once the CF-tree is built, any partitioning or hierarchical algorithms can use it to perform clustering in main memory. BIRCH is reasonably fast, but has two serious drawbacks: data order sensitivity and inability to deal with non-spherical clusters of varying size because it uses the concept of diameter to control the boundary of a cluster.

The advantages and disadvantages of hierarchical clustering methods are:

II.

III. Advantages

- Embedded flexibility regarding the level of granularity.
- Well suited for problems involving point linkages, e.g. taxonomy trees.

A. Disadvantages

- Inability to make corrections once the splitting/merging decision is made.
- Lack of interpretability regarding the cluster descriptors.
- Vagueness of termination criterion.
- Prohibitively expensive for high dimensional and massive datasets.
- Severe effectiveness degradation in high dimensional spaces due to the curse of dimensionality phenomenon.

2.3.DENSITY-BASED CLUSTERING ALGORITHMS

Density-based clustering methods group neighboring objects into clusters based on local density conditions rather than proximity between objects[15]. These methods regard clusters as dense regions being separated by low density noisy regions. Density-based methods have noise tolerance, and can discover non-convex clusters. Similar to hierarchical and partitioning methods, density-based techniques encounter difficulties in high dimensional spaces because of the inherent scarcity of the feature space, which in turn, reduces any clustering tendency. Some representative examples of density based clustering algorithms are:

2.3.1 DBSCAN

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [5] seeks for core objects whose neighborhood (radius) contains at least Minpts points. A set of core objects with overlapping neighborhoods define the skeleton of a cluster. Non-core points lying inside the neighborhood of core objects represent the boundaries of the clusters, while the remaining are noise. DBSCAN can discover arbitrary-shaped clusters, is insensitive to outliers and order of data input, while its complexity is O(N2). If a spatial index data structure is used the complexity can be improved up to O(N log N). DBSCAN breaks down in high dimensional spaces and is very sensitive to the input parameters and Minpts.

DENCLUE

Density-based Clustering (DENCLUE) uses an influence function to describe the impact of a point about its neighborhood while the overall density of the data space is the sum of influence functions from all data. Clusters are determined using density attractors, local maxima of the overall density function. To compute the sum of influence functions a grid structure is used. DENCLUE scales well (O(N)), can find arbitrary-shaped clusters, is noise resistant, is insensitive to the data ordering, but suffers from its sensitivity to the input parameters. The curse of dimensionality phenomenon heavily affects Denclue's effectiveness. Moreover, similar to hierarchical and partitioning techniques, the output, e.g. labeled points with cluster identifier, of density-based methods can not be easily assimilated by humans.

In short, the advantages and disadvantages of density-based clustering are:

IV. Advantages

- Discovery of arbitrary-shaped clusters with varying size
- Resistance to noise and outliers

V. Disadvantages

- High sensitivity to the setting of input parameters
- Poor cluster descriptors
- Unsuitable for high-dimensional datasets because of the curse of dimensionality phenomenon.

Applications of Clustering

Clustering has wide applications in

[1] Pattern Recognition
[2] Spatial Data Analysis:
[3] Image Processing
[4] Economic Science (especially market research)
[5] Document classification
[6] Cluster Web log data to discover groups of similar access patterns

III. CONCLUSION

In this paper we study the different kind of clustering techniques in details and summarized it.we included definition, requirement, application of clustering techniques. we also give detail about classification of clustering techniques and their respective algorithms with the advantages and disadvantages. So this paper provides a quick review of the different clustering techniques in data mining.

REFERENCES

[1] Jiawei Han and Michheline Kamber,Data mining concepts and techniques-a reffrence book ,pg. no.-383-422.

[2] Arun K. Pujari,Data mining techniques-a reffrence book ,pg. no.-114-147.

[3] Ji He,Man Lan, Chew-Lim Tan, Sam-Yuan Sung, Hwee-Boon Low, "Initialization of Cluster refinement algorithms: a review and comparative study", Proceeding of International Joint Conference on Neural Networks [C]. Budapest, 2004.

[4] P. Arabie, L. J. Hubert, and G. De Soete. Clustering and Classification. World Scietific, 1996

[5] Martin Ester, Hans-Peter Kriegel, Jorg Sander, Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". Published in Proceeding of 2nd international Conference on Knowledge Discovery and date Mining (KDD 96)

[6] Anderberg, M.R., Cluster Analysis for Applications, Academic Press, New York, 1973, pp. 162-163.

[7] Biswas, G., Weingberg, J. and Fisher, D.H., ITERATE: A conceptual clustering algorithm for data mining. IEEE Transactions on Systems, Man, and Cybernetics. v28C. 219-230.

[8] Tian Zhang , Raghu Ramakrishnan , Miron Livny, BIRCH: an efficient data clustering method for very large databases, Proceedings of the 1996 ACM SIGMOD international conference on Management of data, p.103-114, June 04-06, 1996, Montreal, Quebec, Canada

[9] Sudipto Guha , Rajeev Rastogi , Kyuseok Shim, CURE: an efficient clustering algorithm for large databases, Proceedings of the 1998 ACM SIGMOD international conference on Management of data, p.73-84, June 01-04, 1998, Seattle, Washington, United States.

[10] Mihael Ankerst , Markus M. Breunig , Hans-Peter Kriegel , Jörg Sander, OPTICS: ordering points to identify the clustering structure, Proceedings of the 1999 ACM SIGMOD international conference on Management of data, p.49-60, May 31-June 03, 1999, Philadelphia, Pennsylvania, United States

[11] George Karypis , Eui-Hong (Sam) Han , Vipin Kumar, Chameleon: Hierarchical Clustering Using Dynamic Modeling, Computer, v.32 n.8, p.68-75, August 1999 [doi>10.1109/2.781637

[12] He Zengyou , Xu Xiaofei , Deng Shengchun, Squeezer: an efficient algorithm for clustering categorical data, Journal of Computer Science and Technology, v.17 n.5, p.611-624, May 2002

[13] He, Z., Xu, X. and Deng, S., Scalable algorithms for clustering large datasets with mixed type attributes. International Journal of Intelligence Systems. v20. 1077-1089