WEEK 8 ASSIGNMENT

Data Systems in the Life Sciences – BIOL 51000 | Fall 2 2020

Christina Morgenstern

CLUSTERING METHODS

Clustering refers to the grouping of data objects into discreet groups or clusters based on a similarity criterion. Thus, objects that are similar to each other are placed into the same cluster whereas objects that are dissimilar to each other are assigned different clusters. The goal of clustering as an unsupervised machine learning algorithm is to find patterns in otherwise unlabeled data. Once a clustering algorithm is implemented, supervised machine learning can be employed in order to assign a new object into one of the defined clusters.

Several clustering algorithms have been developed that differ in the way how a cluster is defined as well as in their efficiency. Criteria that are used to define clusters are based on distance metrics, density as well as intervals or certain statistical distributions [1]. Clustering is applied in a variety of fields such as image analysis, computer graphics, machine learning and bioinformatics.

In the following section, different clustering algorithms and their application are discussed. Generally, we can divide clustering approaches into agglomerative, divisive and hierarchical approaches. The first one starts with every object in an own cluster and iteratively joins clusters together until the optimum is reached. The divisive approache starts with one cluster and iteratively divides the cluster into smaller clusters until the optimal solution is reached. Lastly, hierarchical clustering organizes objects in a tree-like fashion.

1. Simple Threshold Clustering

A simple approach of clustering is taken by the Simple Threshold Clustering algorithm which places similar objects into the same cluster. The algorithm starts off with an object that is assigned to be a cluster and expands this group by adding other objects that are close by. Using a threshold value which defines the furthest distance from the cluster, the algorithm places all objects below that value into the cluster. While this clustering approach is quick, a drawback is that the method is sensitive to the arrangement of the data objects [2]. In general, the algorithm performs in the following steps [3]:

1. The first data object is assigned to the first cluster
2. Repeat until all objects are clustered

The minimum distance between the selected object and the centroid of the clusters is determined. A comparison of the distance with the set threshold value leads to a grouping of the object into the existing cluster or to establish a new cluster if the threshold criterion is not met.

2. Density-Based Clustering

When the distance alone is not sufficient, then the density of the data objects can also be taken into consideration when defining a cluster. These density-based clustering algorithms consider the number and distribution of the neighbor objects. Thus, areas with a high density of objects are considered when forming clusters while objects in sparse regions are considered as noise. DBSCAN (Density-based spatial clustering of applications with noise) is one of the most popular density-based clustering algorithms [4] which has been proposed by Martin, Ester, hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996 [4].

DBSCAN is dependent upon two parameters: *minPts* which refers to the minimum number of points of a region that need to be clustered together to be considered a dense area. And *eps*, a distance measure that is used to locate neighboring points [5]. Furthermore, there are three points that are considered when performing the clustering: core, border and noise. Core refers to a point within the data set that has at least a certain number of points within a certain distance from itself. Border denotes a point with at least one core point within a certain distance. And noise refers to neither a core nor a border point [5].

The DBSCAN algorithm performs in the following steps [5]:

1. Picking up points within the dataset randomly until all points have been visited.
2. If there are at least *minPoints* within radius *eps* then all of these points are part of the same cluster
3. Expansion of clusters by repeating the neighborhood calculations for each neighboring point.

The advantages of DBSCAN are the possibility to find irregular-shaped clusters, the robustness to outliers as well as that the number of clusters doesn´t need to be known in advance as opposed to *k*-means clustering (see part 3) [6].

Drawbacks of this kind of algorithms are that they require dense areas in the data in order to detect clusters.

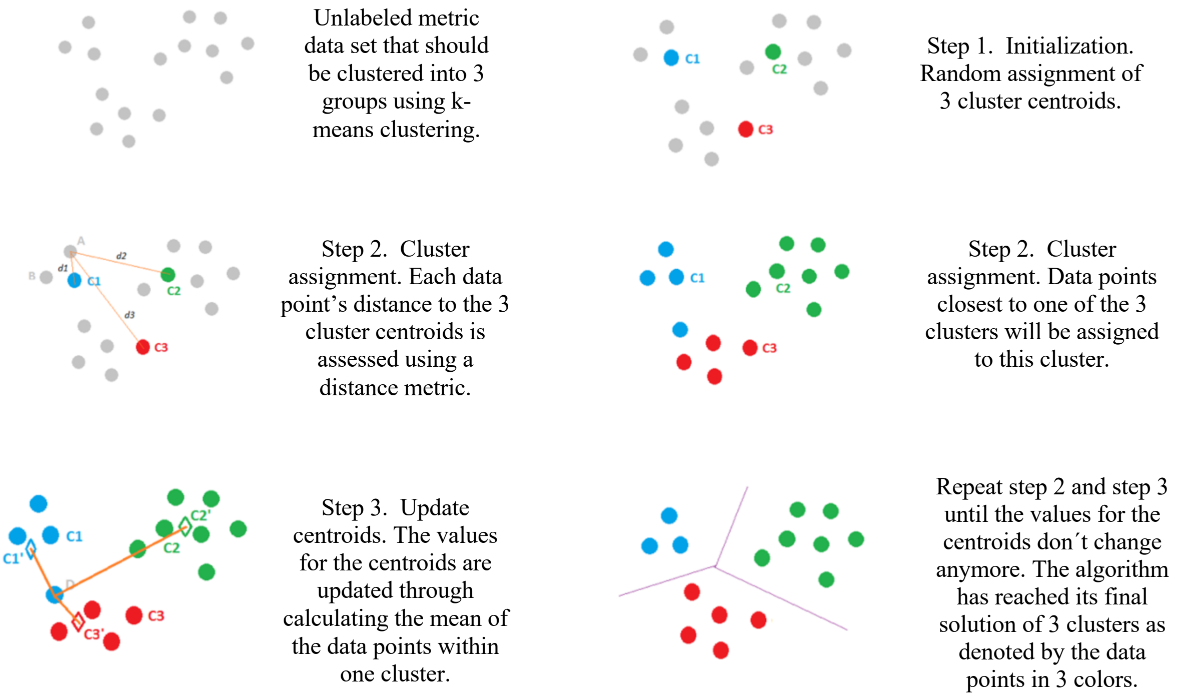
3. *k*-means Clustering

A widely used clustering algorithm is *k*-means clustering which groups objects based on features into *k* number of clusters with *k* being a positive integer number that has to be defined by the user beforehand [3]. The algorithm was first proposed by Stuart Lloyd of Bell Labs in 1957 but the ide goes back to Hugo Steinhaus in 1956 [7], [8].

The clustering is an iterative process which in the case of *k*-means aims at minimizing the sum of squares of distances between the data point and the cluster center. The steps of this algorithm are as follows: a number of *k* clusters is defined, and *k* centroids are randomly chosen. Then the distances of each object to the centroid is determined and the object is grouped based on the minimum distance to the centroid. The process is repeated in an iterative way until the cluster classes don´t change anymore and the algorithm has converged [3].

While the *k*-means partitional clustering algorithm is relatively simple to implement, it has several drawbacks such as the need to know the number of clusters present beforehand. It is also sensitive to outliers and can get stuck in a local optimum thereby missing the global optimum. Variations of the *k*-means clustering algorithm have been developed to improve the algorithm. *k*-medians is related to *k*-means, except instead of using the mean for calculating the centroids, the median value is used [2].

The *k*-means algorithm is a popular non-hierarchical clustering algorithm that segments metric data into predefined number of *k* clusters. In the first step, the initialization, the algorithm chooses *k* random data points as initial centroids, with a centroid being the centre of a cluster. During the second step, cluster assignment, each data point is assigned to a cluster based on its distance to the centroid. The data point with the smallest distance to a centroid and thus closest to this cluster will be assigned to this cluster. Most often, the distance measure Euclidean distance is used to calculate the similarity within clusters. In step three, the clusters need their centroids to be updated. Therefore, the mean of all examples in a cluster will be the value of the new centroid. Steps two and three in this process are repeated until the centroids don´t change anymore and the data points have been separated into the *k* classes. Figure 1 shows a graphical representation of the *k*-means clustering process.



Unlabeled metric data set that should be clustered into 3 groups using k-means clustering.

Step 1. Initialization. Random assignment of 3 cluster centroids.

Step 2. Cluster assignment. Each data point's distance to the 3 cluster centroids is assessed using a distance metric.

Step 2. Cluster assignment. Data points closest to one of the 3 clusters will be assigned to this cluster.

Step 3. Update centroids. The values for the centroids are updated through calculating the mean of the data points within one cluster.

Repeat step 2 and step 3 until the values for the centroids don´t change anymore. The algorithm has reached its final solution of 3 clusters as denoted by the data points in 3 colors.

Figure 1. Visualization of the k-means clustering algorithm (pictures taken from: https://healthcare.ai/step-step-k-means-clustering/).

One major advantage of the *k*-means clustering is its application to large data sets because of the sequential processing and thus the avoidance of similarity matrices among all observations. Furthermore, results are less prone to outliers, the distance metric used or the inclusion of irrelevant variables. On the other hand, *k*-means can only realize its full potential when nonrandom seed points are used. Also, the clustering solutions do not guarantee an optimal solution and require further validation and analysis. The resulting clusters which most often are equally sized and spherical in shape are sometimes not ideal. However, the major limitation of *k*-means clustering is its application to metric data only.

4. Jump Method

If we don´t know in advance the number of clusters and if a threshold-based algorithm is not possible, the jump method can be used. Basically, it is a variety of *k*-means clustering but it performs clustering with different values of *k*. By increasing the number of *k* the performance of the algorithm is monitored and subsequently the best solution involving the number of clusters and taking into account the complexity is chosen. For each value of *k*, the average spread of the data points from the cluster center is calculated for all clusters. This value is regarded as a distortion value which is adjusted for the covariance in the data and is and indicator of the variation of the values around the axes [2].

In order to evaluate the performance of a clustering analysis, several metrics can be employed. Generally, the clustering did well if objects within one cluster are compacted around the centroid and a maximum separation from all other objects of other clusters is present.

Bibliography

[1] 'Cluster analysis', *Wikipedia*. Nov. 29, 2020, Accessed: Dec. 17, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Cluster_analysis&oldid=991283431.

[2] T. J. Stevens and W. Boucher, 'Python Programming for Biology: Bioinformatics and Beyond', *Cambridge Core*, Feb. 2015. /core/books/python-programming-for-biology/61762A9F672FDD8B2DD3FFF8773027B2 (accessed Nov. 07, 2020).

[3] M. Mittal, R. K. Sharma, and V. P. Singh, 'Validation of k-means and Threshold based Clustering Method', vol. 5, no. 2, p. 8, 2014.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, 'A density-based algorithm for discovering clusters in large spatial databases with noise', 1996, pp. 226–231.

[5] 'DBSCAN Clustering Algorithm in Machine Learning', *KDnuggets*. https://www.kdnuggets.com/dbscan-clustering-algorithm-in-machine-learning.html/ (accessed Dec. 19, 2020).

[6] 'DBSCAN', *Wikipedia*. Dec. 05, 2020, Accessed: Dec. 17, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=DBSCAN&oldid=992391686.

[7] S. P. Lloyd, 'Least squares quantization in pcm', *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.

[8] '*k*-means clustering', *Wikipedia*. Dec. 17, 2020, Accessed: Dec. 19, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=994699836.