Data mining and network analysis ITI8665 Tiina Sumeri

Home assignment 1

Distance function, clustering.

Exercise 2. Representative based clustering.

K-means clustering algorithm

Argument for k-means clustering is k, that corresponds to cluster count.

1. Given a set of points, the algorithm finds randomly k points, which will count at first as centroids of the clusters.
2. Next, the algorithm finds all distances between other points of dataset and these centroids and divides the points into k-clusters.
3. After that it calculates a new centroid for each cluster and repeats the second step. New distances are calculated and the points are divided into k-clusters.
4. The second step is repeated until the clusters don’t change anymore.

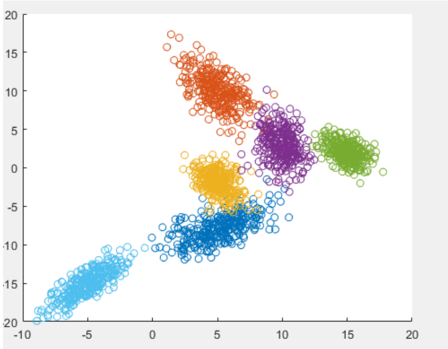


Figure 1. Clusters generated

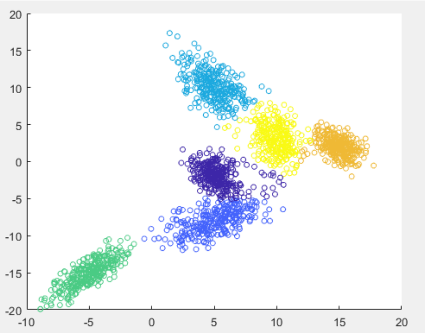


Figure 2. Clustering, p=1

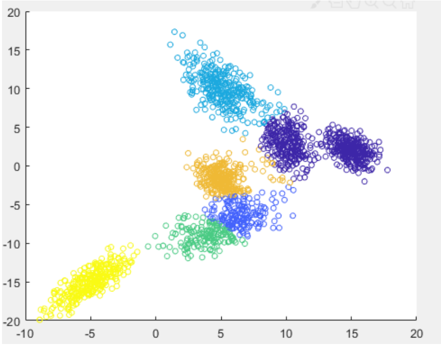


Figure 3. Clustering, p=2

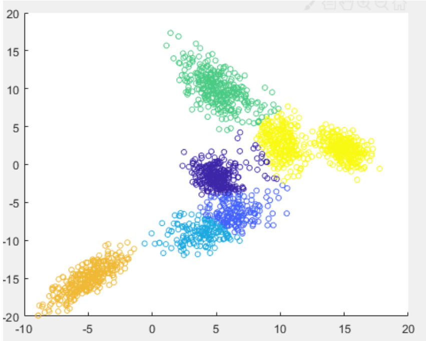


Figure 4. Clustering, p=3

Exercise 3. Density based clustering.

DBSCAN algorithm

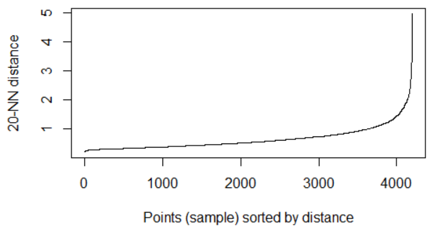
Arguments for DBSCAN algorithm are: distance function, ε (eps) – the radius of a neighborhood of a point and the minimum number of points required to form a cluster (minPts).

1. For each point is found a neighborhood, that stays in radius ε for that point. If the neighborhood for a point is smaller than minPts, a cluster is not formed and the point is labeled as noise point. If the number of points in neighborhood is bigger or even to

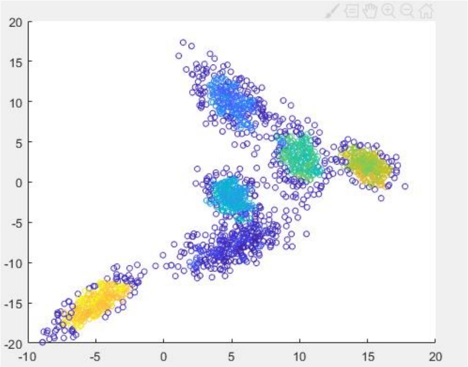
minPts, a cluster is formed and all points in neighborhood become a member of this new cluster.

2. The first step is repeated for each point until all points are either points in a cluster or noise points.

The parameters minPts and ε must be selected well for good clustering performance. For estimating minPts, minimum minPts can be chosen according to the number of dimensions in the data set, minPts >= dimension + 1. ε can be chosen by using a k-distance graph.



Joonis 1. k-distance graph



Joonis 2. Example of clustering with DBSCAN.

In this case, the density in clusters is different, so the cluster in the middle and down is treated like almost all outliers. Other clusters density is smaller, they have less outliers, but still too much.

Analysis

When comparing the clustering by kmeans and DBSCAN, it can be seen that DBSCAN finds right clusters, but too many outliers or noise points, but in some cases kmeans finds false clusters, as example one real cluster is divided into two clusters. DBSCAN can find clusters of more complex shapes and it can find clusters without knowing cluster count. Disadvantage is that DBSCAN can cluster data sets with very different densities.

Exercise 4. Dataset generation

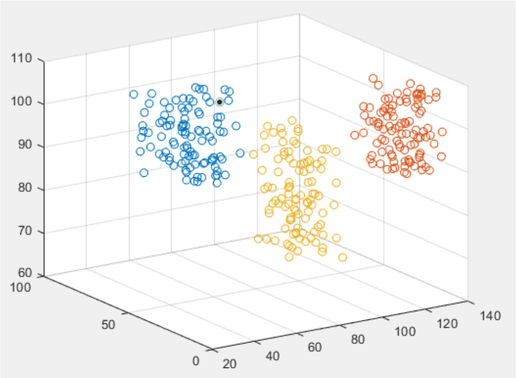


Figure 5. Clusters generated

Hopkins statistics for different : cluster 1 and cluster 2 x-coordinate: H=0.98, cluster 2 and cluster 3: H=0.573

Cluster 1 and cluster 2 y-coordinate: H=0.99, cluster 2 and cluster 3: H=0.99

Cluster 1 and cluster 2 z-koordinate: H=0.49, cluster 2 and 3: H=0.98

From the results can be seen that the clusters 1 and 2 are similar by z-coorinate and clusters 2 and 3 are similar by x-coordinate. Each cluster is well distinguishable by y-coordinate, feature 2.

Code: https://gitlab.cs.ttu.ee/Tiina.Sumeri/iti8665-2019

Materials:

https://www.mathworks.com/help/matlab/math/create-arrays-of-random-numbers.html

https://en.wikipedia.org/wiki/DBSCAN

https://stackoverflow.com/questions/46034396/the-mahalanobis-distance-between-a-point-and-the-mean-vector-is-always-the-same

https://blogs.sas.com/content/iml/2012/02/22/how-to-compute-mahalanobis-distance-in-sas.html

https://www.datanovia.com/en/lessons/assessing-clustering-tendency/#statistical-methods