Distance function and clustering

Omar El Nahhas, 214316IV

I. Introduction

The goal of this assignment is to analyse and compare the application of unsupervised clustering methods and validation thereof. A 3D data set with 4 clusters is created to execute the experiments, see Fig. 1. All functions have been developed from scratch in R, except for functions allowed by the teaching staff. Relevant sources for code (besides course slides/code) have been mentioned in the scripts. The Euclidian distance is used as the distance function in every experiment. All self-written functions have been compared and validated with R packages to confirm their correctness.

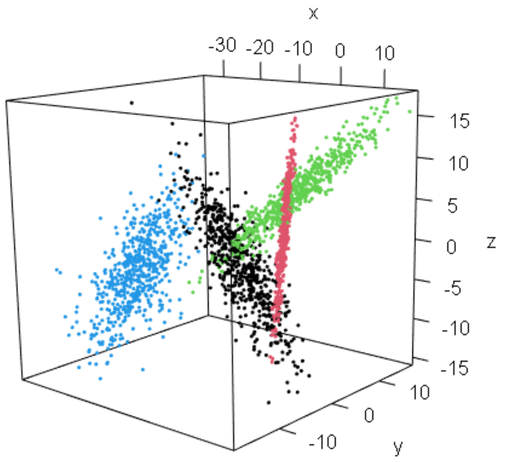

Fig. 1: 3D plot of the generated input data.

II. Feature selection

In Table I the mean Silhouette Coefficient, Entropy and Hopkins Statistic is shown for every feature set and its clustering method. The results overall are quite poor; there is no 2D feature set which convincingly proves a clear winner. However, for the sake of ranking the cluster validation, parameters show that the XY features are the best alternative from the 2D plane. The Entropy and Hopkins Statistic of YZ look quite attractive, but the accompanying mean Silhouette Coefficient of 0.36 and 0.12 for K-means and EM respectively show a poor consistency within the clusters.

TABLE I: Cluster validation K-means and EM (4 clusters).

Method	Features	S. Coeff	Entropy	Hopkins S.
K-means	XY	0.46	4.55	0.86
	XZ	0.45	4.64	0.83
	YZ	0.36	4.22	0.90
EM	XY	0.26	4.55	0.86
	XZ	0.29	4.64	0.83
	YZ	0.12	4.22	0.90

A. K-means

The K-means (K = 4) algorithm uses stopping criteria related to the centroids' movement. If the centroids move less than $1e^{-8}$ compared to its previous location, the algorithm is considered to have converged. The 4-means output in Fig. 2. has one good cluster (blue) and three poor clusters (green, red, black), derived from the Silhouette Coefficient for every separate cluster and visual comparison with the 3D input data.

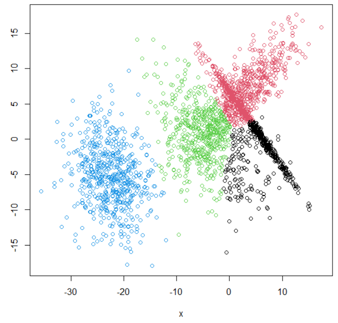

Fig. 2: 2D plot of the K-means with features XY.

B. Expectation-Maximization

The EM algorithm takes 4 clusters as an input parameter. From the output results seen in Fig. 3, it can be observed that the thin dark-blue cluster penetrates the light-blue cluster, and the light-blue cluster penetrates the green one, which shows similar characteristics as the input data in Fig. 1. EM performs better clustering than K-means, which is due to the Gaussian nature of the data and the complex cluster overlap in the data. The EM computation in 3D space with features XYZ can be seen in Fig. 4, showing the importance of using all three features in computations given this data set.

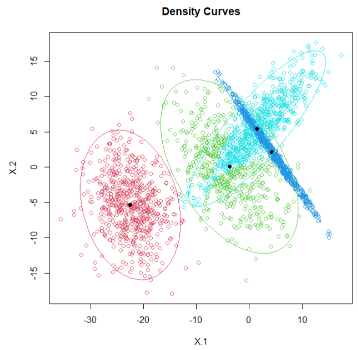

Fig. 3: 2D plot of the EM with features XY.

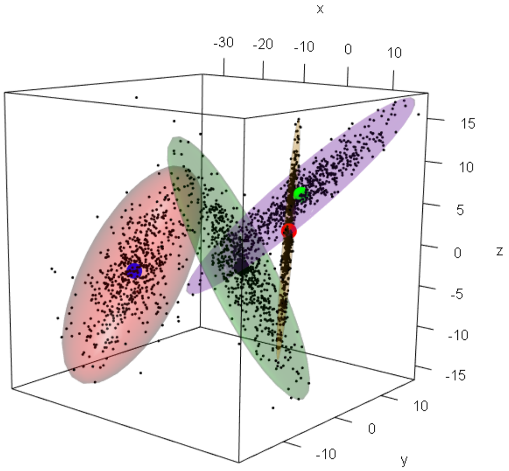

Fig. 4: 3D plot of the EM with features XYZ.

III. Outlier analysis

Using all three dimensions, i.e. feature set XYZ, the Local Outlier Factor is computed with the K-Nearest Neighbours parameter set to 3. The density plot of the LOF is seen in Fig. 5 where a $LOF > 2$ is considered an outlier based on the LOF density.

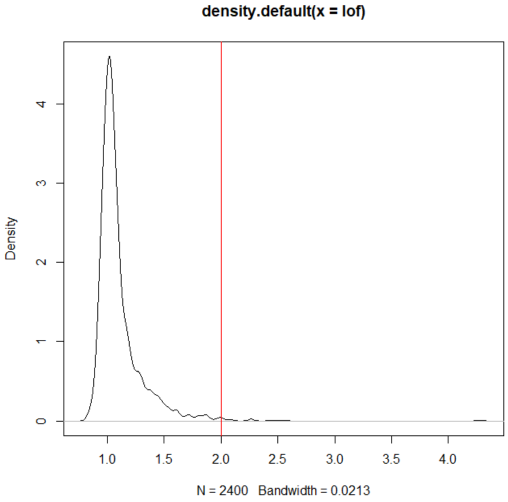

Fig. 5: Density plot of LOF, outliers considered past red line.

Setting the inlier/outlier limit to 2 on the density plot results in a total of 13 outliers in this data set. The results are shown in Fig. 6, with red crosses depicting the outliers according to LOF with k-neighbours equal to 3.

Considering the size of the dataset, 2400 datapoints, and the relatively low number of k-neighbours, the k-neighbours were increased to 30 as an experiment. This 10x increase in k-neighbours resulted in 16 outliers according to LOF, 3 more outliers than using 3-neighbours LOF. Therefore, the k-neighbours parameter should be analysed in more detail for its effect on the LOF in future research, as the current relation of k-neighbours to the output results is unclear from the small sample of this single experiment.

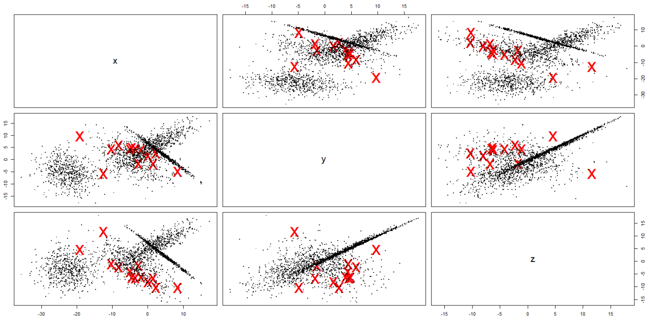

Fig. 6: Pairs plot showing 13 outliers (red cross) from all perspectives (X, Y and Z).

IV. Density based methods

The DBSCAN algorithm has been implemented with an epsilon value of 2 and a minimum sample requirement of 10. The results are seen in Fig. 7, where the black points are defined as outliers. DBSCAN found 3 clusters instead of 4, because of the crossing of two clusters now both labeled as green. However, considering the way DBSCAN is implemented, this is a very logical outcome and can therefore be considered correct within DBSCAN's limitations.

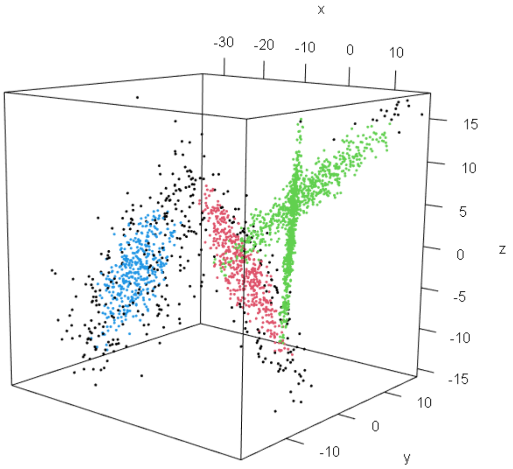

Fig. 7: DBSCAN showing three clusters and outliers (in black) with $\epsilon = 2$ and minimum samples of 10.