Technical University of Denmark

02450 Introduction to Machine Learning and Data Mining

Project 3

s191985 Julian Böhm (Outlier Detection)
s196119 Emil Chrisander (Association Mining)
s192184 Jorge Montalvo Arvizu (Clustering Analysis)

December 2019

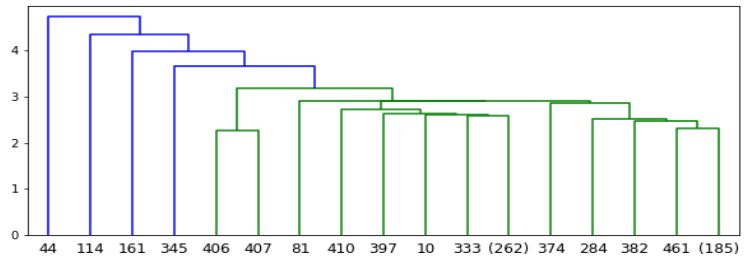
1 Clustering Analysis

To perform the hierarchical clustering it was necessary to perform the standarization of the observations, given that our dataset presents high variance. Apart from this, the variable **famhist**, which is a classification binary variable taking values 0 or 1, was one-out-of-K coded. Therefore, our new dataset $\mathbf{X}$ is a NxM matrix, where $N = 462$ and $M = 14$. The hierarchical/agglomerative clustering was performed on the data matrix $\mathbf{X}$ using a maximum number of clusters of **2**, given that our target variable **y** consists of binary variables indicating the presence of the sickness of the patient (observation). The clustering was assessed by changing the linkage function and using the Euclidean distance as the distance metric.

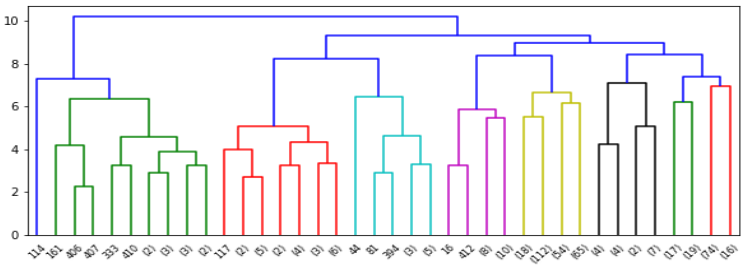
Linkage Function	Cluster 1	Cluster 2
single	461	1
complete	16	446
average	461	1
ward	174	288

Table 1: Linkage functions - Observations allocated to each cluster

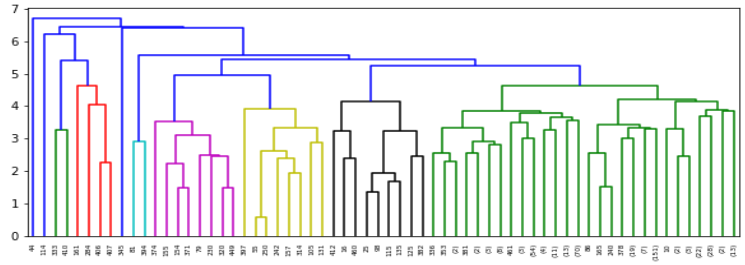
Table 1 shows the allocation of the observations in each cluster for each of the linkage functions. The first three functions allocate most of the observations in one cluster, as can be seen also from the dendrograms show in **Figure 1a** to **1c**. We can suggest that there may be outliers in the data, since the left-most side of the dendrograms show a big height with only a few clusters. Nevertheless, the ward function seems to obtain better results thanks to it's nature of computing the sum-of-squares error of the distance from each observation to its cluster center. As a reference, the real classification is 160 persons with the disease and 302 without the disease; which seem to somewhat match the ward function classification results. Given the high dimensionality of our data, it's hard to see or interpret the results of the hierarchical clustering but we can at least obtain a better visualization of them by projecting on the Principal Component 1 and 2 obtained in the Principal Component Analysis (PCA).



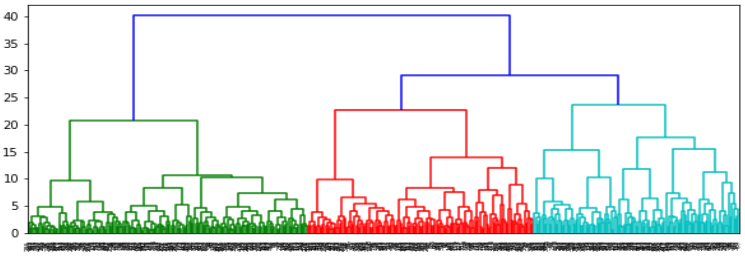
(a) Single



(b) Complete



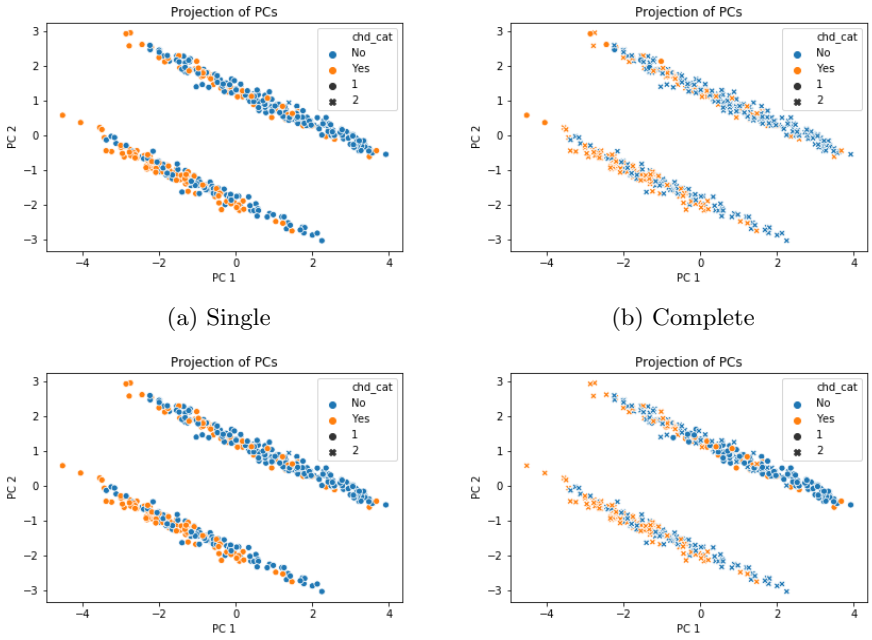
(c) Average



(d) Ward

Figure 1: Hierarchical Clustering Results - Dendrograms

Figure 2 shows the results of each linkage function projected on PC1 and PC2; where the colors represent the presence of the sickness (blue = not present, orange = present), while the shape represent the cluster (circle = cluster 1, cross = cluster 2). Only the ward function appears to cluster the observations and allocate some of them correctly, compared to the real **y** classification, i.e. blue circles in the upper-right side of **Figure 2d**.



(a) Single

(b) Complete

(c) Average

(d) Ward

Figure 2: Hierarchical Clustering Results - Projections on PC1 and PC2

We further analyzed the clustering by computing the supervised measures of cluster validity: Rand Statistic, Jaccard coefficient and normalized mutual information (NMI), the results are shown in **Figure 3** where the x-axis indicates the linkage function (1=single, 2=complete, 3=average, 4=ward). Even though the linkage functions single, complete and average seem to have better results (similar to the real classification), this is in reality related to the results of **Table 1**; where almost all of the observations were classified in one particular cluster, thus the validity measures show a high similarity value even though it's only taking all the observations and classifying them in one cluster. Again, the ward function seems to give good results compared to the other linkage functions.

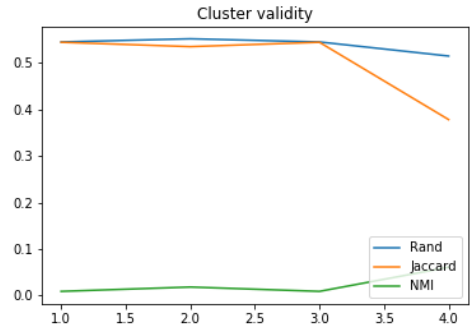


Figure 3: Validity measures - Linkage Functions

Next, we clustered our data using the Gaussian Mixture Model (GMM), where we make use of the empirical mean, empirical covariance and empirical mass of the clusters and iteratively update the parameters and calculate the likelihood until we reach convergence or maximum iterations. Given the non-deterministic nature of the model, we initialized it's parameters by setting the number of fits with different initialization to 10 and using the k-means method as initialization procedure. We then proceeded to create a K-fold cross-validation for $K = 1$ to 10, the results are shown in **Figure 4**.

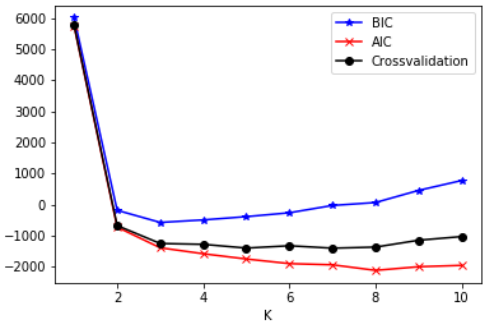


Figure 4: GMM Component - K-folded Crossvalidation, BIC and AIC

From the results, we can either take $K = 2$ or $K = 3$. By following the elbow method, $K = 2$ is preferred by us. By selecting $K = 2$, we obtain the empirical mean, covariance and mass of the clusters, i.e. the cluster center, shape and relative size/density. By focusing on the centers μ, we can argue that these empirical means are the centroids of the clusters that best fit the data, i.e. the value of the parameter that maximizes the log of the likelihood of the data.

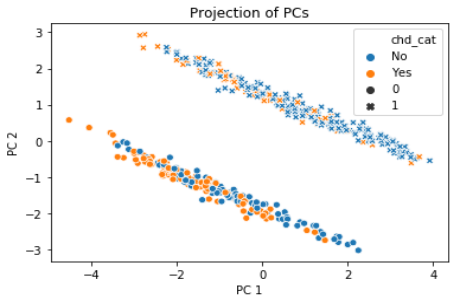


Figure 5: GMM Results - K=2

Figure 5 shows the results of the GMM with K=2. We can see that compared to **Figure 2d** the results are similar, with the difference that GMM clustered the top part of the graph in a single cluster, while hierarchical clustering did it only in the bottom part of the top cluster. By computing the supervised measures of cluster validity (Rand, Jaccard and NMI), we can compare the results of GMM against the real classification and against the hierarchical clustering at K=2 (which are the same results as above). **Table 2** shows the results of the comparison. We can see that even though the comparison between GMM and Hierarchical clustering isn't 100% similar, they both have almost identical validity measures against the real classification. It is hard to decide which model might be better given the high dimensionality of our data, but we can have a good idea of how both approaches work and we can say that at least the quality of the classification on both models is around 50%, depending on the validity measure.

Measure	GMM vs. Real	GMM vs. Hierarchical
Rand	54.63	67.00
Jaccard	40.03	51.92
NMI	5.60	4.21

Table 2: Validity Measures - GMM vs. Real and GMM vs. Hierarchical Clustering

2 Outlier and Anomaly Detection

In this section, we focus on the detection of outliers and anomalies. In general, observation outliers are assumed to lie in regions with low density[1]. Therefore, we firstly estimate the density of the data set. Secondly, the observations located in the low density regions are identified.
Following methods have been applied to evaluate the density distribution:

[1]Introduction to Machine Learning and Data Mining. Chapter 20, 2019; Tue Herlau, Mikkel N. Schmidt, Morten Mørup.

- Normalization of the data due to scale variation of the different parameters
- Kernel density estimation (KDE) using the leave-one-out cross-validation
- KNN density
- KNN average relative density (ARD)

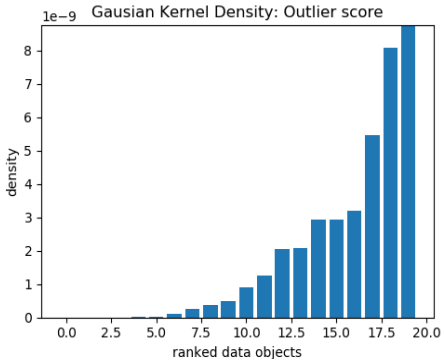


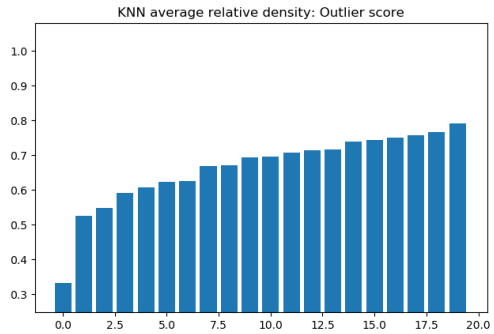
Figure 6: Kernel Density Estimation: 20 observation located in the lowest density region

	numbers of the 6 lowest-density observations (ascending)
Kernel density estimation	[44, 114, 345, 161, 374, 81]
KNN density (K=5)	[44, 114, 345, 161, 397, 374]
KNN average relative density (K=5)	[44, 114, 217, 218, 345, 67]

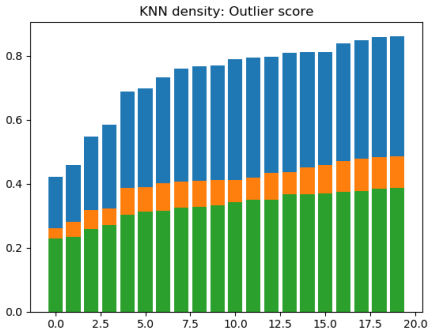
Table 3: The 6 lowest-density observations generated by different density estimators

Results - Kernel density estimator (KDE): The results are generated with an optimal kernel width λ of 0.25. In Figure 6, one can note that some observations are present in regions with very low densities. After investigating the table with sorted densities (excerpt in table 3), following observations could be framed. Observation number 44 lies in the region with the lowest density (1.68192e-23). Having a closer look at this observation, it seems unusual that the value of obesity is with 46.58 the highest of the whole data set and adiposity is in the lower range with 9.74 (range of adiposity in the data: min 6.74, max 42.49). Nevertheless, it is physically possible that a person has a very high Body-Mass-Index but a low body fat percentage, e.g. athletes or muscular

people can show these characteristics[2]. Supported by the fact that the suspect neither consumes alcohol nor tobacco, we keep the observation in the data set. Additionally, observations 114, 345 and 161 are located in a region with a low density (from 1.47275e-20 to 1.87351e-14). However, in this observations no contradictory or unrealistic values could be detected.



(a) KNN average relative density: 20 observation located in the lowest density region ($K = 5$)



(b) KNN density: 20 observation located in the lowest density region (orange: $K = 5$)

Results - KNN density and average relative density (ARD): The results are obtained using $K = 5$. Green and blue plots in figure 7b were produced using $K = 2$ and $K = 10$, respectively. The results gained by KNN density and ARD are comparable to the ones of KDE. Table 3 shows the same 4 lowest-density solutions for KDE and KNN density. Besides, one can see a deviation of ARD observations after the second lowest object. This may be caused by the different non-probability based methodology of ARD. Finally, it

[2]see `https://www.hsph.harvard.edu/obesity-prevention-source/obesity-definition/obesity-definition-full-story/`

can be summarized that the methods showed us the observations which feature some odd values. However, after checking the credibility of the observations, we decided to exclude non.

3 Association Mining

We decided to set the aim of our association mining to find association rules for a positive heart disease diagnose. That is, we want to explore whether there are specific item sets (patient attributes) which are associated with a positive heart disease diagnose. This allows us to create a simple screening device that in principle could be used manually by health care personal in South African villages. Before we present our association mining results, we explain our procedure.

3.1 Association Mining Procedure

Our approach is:

1. Seperate numerical and categorical attributes into two groups.
2. For each numerical attributes we create a new string attribute indicating whether the row value is within 1^{st}, 2^{nd}, 3^{rd}, or 4^{th} quartile. We apply quartiles, as it gives us a good mix of differences in attribute levels while maintaining a sufficient amount of persons in each 'bin'. We did try with both less and more granular percentiles for binarization. The most convincing association rules were obtained when we applied quartiles.
3. We binarize the variables from step two using one-hot-encoding. Thus, we obtain 4 binary attributes for each numerical attribute.
4. We binarize the categorical attributes using one-hot-encoding.
5. We combine the binarized numerical and categorical attributes into a single data set.
6. We transform the binary values to attribute labels. I.e. a label value may be *age_75%_to_100%*, which indicates that the person is in the 4^{th} quartile in the age distribution of our sample.
7. We run the apriori algorithm using the apyori module in python. We tried different values of $\epsilon_{support}$ and $\epsilon_{confidence}$ to explore the forthcomming rules. In our final analysis we apply $\epsilon_{support}$ at 0.05 and $\epsilon_{confidence}$ at 0.6. Recall, that only 35 percent of the sample have a positive heart disease. Hence, we cannot expect to get any higher support than 0.35. A 0.05 value of $\epsilon_{support}$ is equivalent to at least 23 persons in our sample having the item set $\{X\}$. We do not go any lower than 0.05 to avoid association rules based on very few people in our sample. We set confidence at 0.6 to

ensure that more than 60% do actually have the heart disease when they have an item set $\{X\}$.

8. As an additional feature we also calculate the so-called lift. Lift is a measure of how more likely the item set X is when Y is include[3]. In our context it tells us something about how much more likely a person is to be diagnosed with a heart disease *if* the person has a given item set X. I.e. if the lift is 3 for $X, Y = \{age\_75\%\_to\_100\%\, chd\_yes\}$, we know the 4^{th} quartile of age is three times more likely to have a heart disease diagnosed.

3.2 Association Mining Results

Table 4 shows our 7 identified association rules. The rules are ordered by their support size. The rules are surprisingly intuitive and close to what one would expect. We only obtain rules based on two or three items. 6 out of 7 rules include a presence of heart diseases in the family. For example, if a person has a family history of heart diseases *and* the person is in the 4^{th} quartile of the ldl ("bad cholesterol") distribution, the person is 7.2 times more likely to have a heart disease than not having one. The support of this rule is 0.1. Hence, 10 percent of the sample apply to this rule. Another interesting association rule is $\{X, Y\} = \{[tobacco\_75\%\_to\_100\%, age\_75\%\_to\_100\%], chd\_yes\}$. It has a support of 0.07 and lift of 9.1 making it a relatively strong screening rule. Moreover, the doctors would not need to make any medical examinations of the patient to determine the age and tobacco consumption. Thus, in practice this would make it a very low cost and feasible screening device rule. Obviously, the rule would have to be re-transformed into actual levels of age and tobacco consumption prior to handing out to local village doctors.

We conclude that association rule mining has a strong potential to serve as simple low cost screening devices for the Ischaemic heart disease in local South African villages.

X	Y	support	confidence	lift
ldl_75%_to_100%, famhist_present_yes	chd_yes	0.10	0.72	7.2
adiposity_75%_to_100%, famhist_present_yes	chd_yes	0.08	0.63	7.9
age_75%_to_100%, famhist_present_yes	chd_yes	0.08	0.70	8.8
sbp_75%_to_100%, famhist_present_yes	chd_yes	0.08	0.64	8.0
tobacco_75%_to_100%, famhist_present_yes	chd_yes	0.08	0.64	8.0
tobacco_75%_to_100%, age_75%_to_100%	chd_yes	0.07	0.64	9.1
adiposity_75%_to_100%, obesity_75%_to_100%, famhist_present_yes	chd_yes	0.05	0.61	12.2

Table 4: Association rules for heart disease being diagnosed

[3]see https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/