Technical University of Denmark

02450: Introduction to machine learning and data mining

Project 3: Indian Liver data set, Clustering, association mining, anomaly and outlier detection

Authors:	**Studynumbers**
Mads Christian B. Andersen	s173934
Johannes Boe Reiche	s175549
August Semrau Andersen	s183918

Mads C. B. Andersen, August S. Andersen, Johannes B. Reiche

December 3, 2019

Introduction

Problem at interest

Opposed to supervised learning, which in report 2 was achieved by mapping from some input space X, to some output space y, the goal of unsupervised learning is to extrapolate structure and patterns in the data only given X.
In this report different methods of unsupervised learning will be used for the 3 different purposes. First of clusterings methods such as Hierarchical Clustering and Gaussian Mixture Models will be used to group our observations. Secondly to find outliers with density estimations and at last to find common relationships between attributes with Association Mining techniques.

Clustering

For a larger data set X a problem of interest of unsupervised learning could be to identify groups of data points in an arbitrary space dimension. Two methods for this could be the K-means and the hierarchical clustering methods. The K-means algorithm requires a distance measure (typically euclidean distance measure) between observations, whereas the hierarchical clustering measure the distance between groups of observations. For this report we will consider the hierarchical clustering method. A different clustering problem of interest in unsupervised learning could be to derive a distribution of a M-dimensional data point using the Gaussian Mixture Model (GMM). This will also be covered in one of the following sections. Lastly, we will try to compare the two clustering methods using three different similarity measures.

Hierarchical Clustering

For our particular Indian Liver data set with M=11 attributes (including *is_patient*), we are very much interested in the attribute *is_patient*. Previously in Project 2 we have tried to use supervised learning classification methods like ANN to try and classify if a person has a disease or not, based on M-1 attributes. In this section, we try to cluster two groups of data points on the basis of all M attributes and hope to achieve some kind of unsupervised classification, where the two groups of patients are clustered separately. Hierarchical agglomerative clustering starts by placing each observation into separate groups, and then iteratively merge the two closest clusters (initially single observations). Merging two clusters is done on the basis of a distance/dissimilarity measure. The proximity between two clusters depends on this distance measure and the linkage function. For this section we will consider the euclidean distance, and the complete (maximum) linkage function. The complete (maximum) linkage function is illustrated in figure 1:

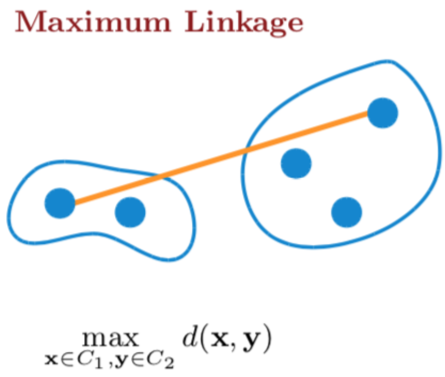
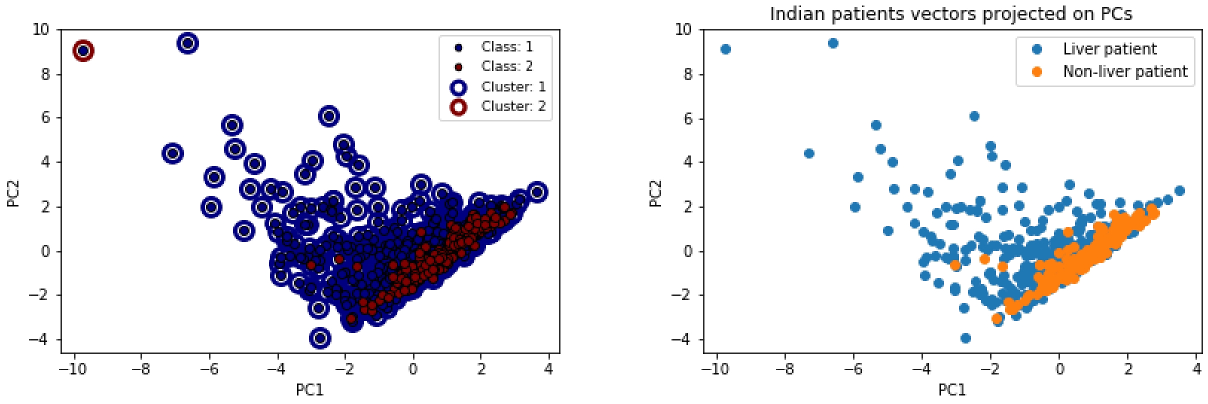

Maximum Linkage
$\max_{\mathbf{x}\in C_1,\mathbf{y}\in C_2} d(\mathbf{x},\mathbf{y})$

Figure 1: Illustration of the 'Maximum Linkage' distance measure.

It means that the proximity between two clusters is the distance between the most distant pair of observations in the considered clusters. The argument for choosing this linkage function is that our data set consists of attributes with high variance for patients and very similar values for non-patients. To illustrate the result of the hierarchical agglomerative clustering, the clusters have been projected into the 2-dimensional PCA-space. To visually see how good a job the hierarchical agglomerative clustering method have done, we can compare it to the original PCA-plot from project 1 in figure 2:



(a) Hierarchical clustering

(b) Original PCA

Figure 2

As the two plots reveal, the hierarchical agglomerative clustering method does a pretty good job of clustering the liver patients in one cluster and the non-liver patients in a different cluster. Of course a 2D-dimensional interpretation simplifies thing a bit, and it is difficult from these plots to interpret the M-dimensional trend of the clusters. Yet from a classification point of view, it looks like the two clusters nicely contain the two types of patients.

Gaussian Mixture Model

In this section we will try to cluster the Indian data liver set by the GMM using cross-validation. The GMM tries to derive a distribution of the M attributes for a data point (patient). To find the optimal number of components in the GMM, we use one layer K1=10 fold cross-validation. The range of components tested will be K=1..15, since we have a M=11 dimensional data set and it remains to be seen which number of components is optimal. Using the Expectation maximization (EM) algorithm to train the GMM and maximizing the negative log-likelihood, the optimal number of components in the GMM can be obtained in figure 3, with the number of components K on the x-axis and the negative log-likelihood on the y-axis:

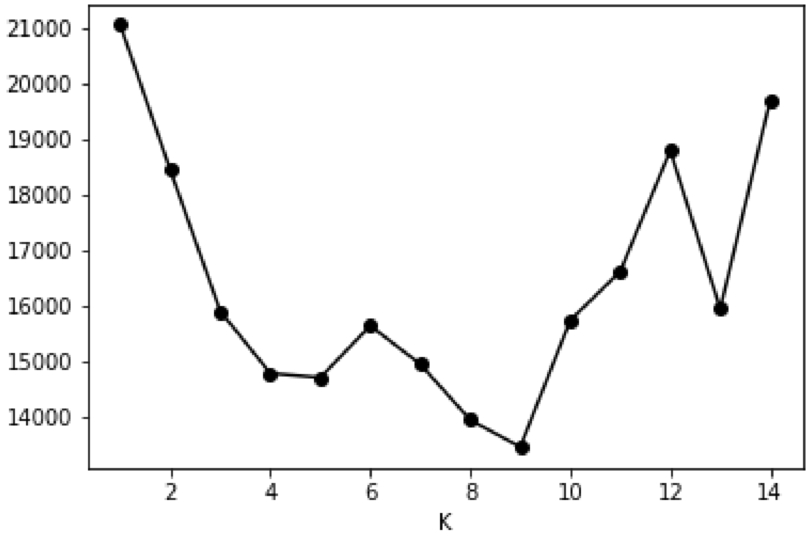


Figure 3: CV for number of components in GMM

What we find is that the optimal number of components is K=9. This means that the GMM model best approximates the N·M dimensional dataset X with 9 components, which almost corresponds to the number of attributes M=11. This implies that the GMM tries to find the distribution of the M attributes. But why only have 9 and not 11? Well one argument could be that we have two binary attributes (*gender* and *is_patient*), which may increase the test error when testing with K=10 or K=11 components. Another argument could be that two of the attributes (*tot_proteins* and *ag_ratio*) have very high standard deviation compared to the rest of the attributes. Additionally these two attributes aren't normally distributed, as found in project 1 and the GMM is using the multivariate normal distribution, which could lead to increased test errors.

One way of interpreting what the GMM learns from the dataset is to try and interpret the cluster centers. Here we will consider the cluster centers for K=9 components. The cluster centers and a statistical overview of the attributes can be found in appendix, figure 4 and table 4. Even though it is hard to interpret these cluster centers meaningfully, some of the center coordinates is interesting. Etc. the first cluster center coordinate is basically the mean of all the attributes. A cluster that differentiates a bit from the first cluster is the last cluster, here the coordinates indicate younger people (age=36.98 compared to mean age=44.78), typically men (gender=0.15 compared to mean gender=0.24, 'man' = 0 and 'female' = 1) with higher values of *tot_bilirubin*=9.11 compared to mean *tot_bilirubin*=3.31.

Evaluation of clustering methods

In this section we will try to evaluate the quality of the clustering in terms of the label *is_patient* for the GMM and the hierarchical clusters. The estimated number of clusters by the GMM was found in the previous section to K=9 and therefore we will also consider a hierarchical clustering with 9 clusters. To evaluate the quality of the clustering we consider three different similarity measures, namely Jaccard, Rand and NMI. The results can be seen in table 1:

Similarity measure	Jaccard	Rand	NMI
GMM	0.45	0.65	0.41
Hierarchical clustering	0.45	0.51	0.04

Table 1

What we find is that the Jaccard index is the same for the two types of clustering. A Jaccard index is the number of times a pair of observations is in the same cluster in both partitions (S) relatively to the total number of pairs ($\frac{1}{2}N(N-1) = \frac{1}{2}579(579-1) = 167331$ pairs) with the number of pairs where the observations are in different clusters in bots cluster-model (D) subtracted. This is put together in (1) where Q and P are the two considered partitions, which in our case is the partitions found by our two clustering methods and our label information:

$$J(Q,P) = \frac{S}{\frac{1}{2}N(N-1) - D} \tag{1}$$

The Jaccard index thereby discard the number of non-matches (number of pairs that have different labels and are assigned to different clusters). This is opposite to the Rand index, expressed in (2):

$$R(Q,P) = \frac{S+D}{\frac{1}{2}N(N-1)} \tag{2}$$

In which the non-matches are in-counted, and for the same reason we obtain that R(Q,P)>J(Q,P). But what's interesting is that the Rand index is almost the same as the Jaccard index for Hierarchical clustering, which indicates that the number of non-matches is low, and lower compared to the GMM. This means that relatively low number of pairs with different labels (patient vs non-patient) is assigned to different clusters. In other words, there are a lot of pairs that have the same label but are in different clusters or the other way around. This can maybe be explained by that some of the observations (patients and non-patients) are very similar and will be clustered together by the hierarchical clustering method which is distance based. On the other hand, S is larger in hierarchical clustering than in GMM but the GMM have more non-matches D.

For the NMI (normalized mutual information) we obtain very different results. The NMI is expressed in (3) as:

$$NMI[Q,P] = \frac{MI[Q,P]}{\sqrt{H[Q]}\sqrt{H[P]}} \tag{3}$$

What it explains is the amount of information shared by the two partitions (in bits) relatively to the total amount of information in the two partitions. For the hierarchical clustering we get a fairly low NMI, meaning that the label information and the clusters are quite dissimilar and one partition therefore doesn't provide much information about the other partition. For the GMM it is a different story, since the NMI indicates that the label information and the clusters in the GMM are more similar and that one partition provides some information about the other partition. It makes sense that the GMM has higher 'scores' overall than the hierarchical clustering method, since the GMM dictated the number of clusters, whereas the results might be different had we considered a hierarchical clustering of just two clusters as previous.

Outlier detection / Anomaly detection

Irregularities in data can come in many shapes and forms and might be caused by an even wider range of events. In certain situations these anomalies is unwanted, and therefore needs to be tracked. A typical example of this is the one of credit card transactions, where an anomaly could be an unusually large withdrawal at a late hour. A series of irregularities could potentially be the work of a thief and an anomaly detection system could automatically block the given credit card. One way of detecting outliers in a data set is to estimate the densities from which the data come from. Another way to put it is that if an observation lies in a region in hyperspace that has a low density based on current empirical data, it is plausible that there is a special reason that is ended there. In our case of medical measurements, a highly irregular attribute might be vital to a doctors diagnosis and treatment. In this section different methods of estimating the densities from which the data can be described have been modelled. Specifically the Gaussian kernel density, KNN density and KNN average relative density have been calculated for each patient, and a search have been performed to identify outliers. For all kernel density estimator methods the bandwidth / kernel width is a parameter that strongly influence the observation scores and the methods of how this is chosen differs between the models. For the Gaussian kernel estimator, a probabilistic approach is used, and a normal distribution is fitted to each observation and the optimal variance is found with leave-one-out cross validation. All of these individual normal distributions are then combined and a shared density data distribution is the result. The KNN density is sometimes called the inverse average distance, and this the reason is quite obvious when you look at the formula:

$$\text{density}_{\boldsymbol{X}_{|i}}(\boldsymbol{x}_i, K) = \frac{1}{\frac{1}{K}\sum_{\boldsymbol{x}' \in N_{\boldsymbol{X}}}(\boldsymbol{x}_i, K) \cdot d(\boldsymbol{x}_i, \boldsymbol{x}'})$$

The density corresponds to the inverse distance to the observations neighbours, which means that observations with the lowest scores are pointed out as outliers.

Both the Gaussian Kernel density and the K nearest neighborhood density methods are not able to handle different data densities well. However the deterministic KNN-average relative density is designed to be able to account for several clusters of data with different densities. This is also part of the reason why the GMM and KNN kernels seem to agree more in the rankings of observations, which is showed in table 2. The formula for the KNN-ard is:

$$\text{ard}\, x\,(x_i, K) = \frac{\text{density}_{X_{|i}}(x_i, K)}{\frac{1}{K}\sum_{x_j \in N_{X \setminus i}(x_i, K)} \text{density}_{X_j}(x_j, K)}$$

From the formula, it can be seem that the density measure of an observation depends on the density of it's neighbors, which means that several density spreads can be accounted for.

Rank

The outlier results can be seen in the table [2] beneath. These result are obtained by first standardizing the inputs, however due to very large fluctuations in magnitude of the inputs, some observations receive very low scores.

Methods:	Gaussian Kernel Density		KNN		KNN ARD	
Outlier no:	Patient No:	Score	Patient	Score	Patient No:	Score
1	**135**	1.06e-161	**135**	0.109	**166**	0.355
2.	**166**	6.28e-115	**166**	0.154	**135**	0.425
3.	**117**	1.66e-65	**117**	0.169	217	0.470
4.	**199**	1.32e-39	**199**	0.211	565	0.483
5.	**571**	3.19e-30	**571**	0.268	352	0.544
6.	**443**	1.20e-27	**443**	0.277	528	0.549
7.	207	3.19e-22	**27**	0.302	454	0.557
8.	**47**	2.27e-18	**47**	0.310	280	0.562
9.	**27**	2.27e-18	568	0.313	323	0.567
10.	529	2.23e-16	544	0.326	264	0.574

Table 2: Outlier rankings with 3 different methods

As can be inferred from the table, there are patients that are identified by all the anomaly detection techniques. The patients with ID: 135, and 166 seem to be picked out from the crowd by all three methods, and beneath their attributes have been displayed with mean and median values:

Attribute:	Mean μ:	STD σ:	Median	Patient No.135	Patient No.166
age	44.782	16.222	45.0	66	55
gender	NaN	NaN	NaN	Male	Male
tot_bilirubin	3.315	6.228	1.0	11.3	75
direct_bilirubin	1.494	2.816	0.3	5.6	3.6
tot_proteins	291.366	243.562	208.0	1110	332
albumin	81.126	183.183	35.0	1250	40
ag_ratio	110.415	289.850	42.0	4929	66
sgpt	6.482	1.085	6.6	7	6.2
sgot	3.139	0.794	3.1	2.4	2.5
alkphos	0.947	0.320	0.9	0.5	0.6
is_patient	NaN	NaN	NaN	Yes	Yes

Table 3: Summary statistics of data set with 2 of the most prominent outliers.

When making a qualitative evaluation of the patients with ID: 135 166 seen in table [3], it seems reasonable to call these outliers. For patient with ID 135, almost every attribute seems to deviate with more than 2 standard deviations from the mean of the attribute and especially *ag_ratio* is more than 16 standard deviations away. We know that the attributes are not normally distributed, but even so, the likelihood is very low, hence it is a very reasonable conclusion. Patient with ID 166 can be said to be a less obvious outlier, and his measurements are for most attributes within a reasonable range (based upon the standard deviation) of either the mean or the median. However the measurement that justifies that he is an outlier is his very large *tot_bilrubin*, which is quite far from the mean and median.

Association mining

We will in this last part of the analysis of the Indian-Liver-Patients data set use association mining in order to find connections between attributes. Association mining is based on the Apriori principle, which states that; *If an item set is frequent, then all of its subsets must also be frequent.*
In practice this means, that if some attributes appear frequently together, then the subsets of this group of attributes must all also appear frequently together. More importantly it also works the other way around, meaning that a group of etc. two attributes that are infrequently represented, will result in all groups containing these two also being infrequently represented.

Based on this principle, Association Rule Discovery will be used, in which we seek to identify important co-occurrence relation- ships. These, which we call 'rules', are found given transactions T (in our case the attributes of the data set) that meet thresholds of support and confidence;
Support is calculated as the percentage of all transactions that contain X, etc. 76 % of test patients are male, therefor the support of male patients is 76 %.
Confidence is the percentage of transactions containing X that also contain Y, ie. the confidence of having a liver disease for male patients is 74 % as almost three quarters of male patients have a disease in the data set.

$$support = s(Y \leftarrow X) = \frac{\sigma(X \cup Y)}{N} = P(X,Y) \geq minsup$$

$$confidence = c(Y \leftarrow X) = \frac{\sigma(X \cup Y)^T}{\sigma(X)} = \frac{P(X,Y)}{P(X)} = P(Y|X) \geq minconf$$

where *minsup* and *minconf* are the corresponding support and confidence thresholds.

The data set is for this task transformed into a binary set, in which all attributes, except for gender and is_patient, are split into two halves. This means that etc. for the attribute age, there is a column that is a one if the person is under the median age (ie. age is in 0-50 percentile) and a zero if the age is over median. There is a matching column that mirrors this. For gender and is_patient the attribute is represented by two columns for male/female and sick/not sick.

When running the Apriori algorithm, the threshold values for support and confidence need to be low enough that any rules are formed at all, while high enough that the rules found are of analytical interest. There should only be so many rules that a clear picture can be attained from the rules.

The Apriori algorithm is now run on the data set, first with the threshold of *minsup = 0.45*

and *minconf = 0.75*:
Here, six different associations/rules are mined, which can be seen below:

$$\begin{gathered}
1 : \{\} \rightarrow \{gender\_Male\}(supp : 0.758, conf : 0.758)\\
2 : \{direct\_bilirubin\_50 - 100perc.\} \rightarrow \{gender\_Male\}(supp : 0.466, conf : 0.833)\\
3 : \{direct\_bilirubin\_50 - 100perc.\} \rightarrow \{is\_patient\_Sick\}(supp : 0.454, conf : 0.812)\\
4 : \{direct\_bilirubin\_50 - 100perc.\} \rightarrow \{tot\_bilirubin\_50 - 100perc.\}(supp : 0.522, conf : 0.932)\\
5 : \{tot\_bilirubin\_50 - 100perc.\} \rightarrow \{direct\_bilirubin\_50 - 100perc.\}(supp : 0.522, conf : 0.990)\\
6 : \{is\_patient\_Sick\} \rightarrow \{gender\_Male\}(supp : 0.558, conf : 0.780)
\end{gathered}$$

Amongst these, we see that some are more interesting than others;
The first rule is caused solely by the larger percentage of males in the data set exceeding the confidence threshold, and this is therefor disregarded.
The fourth and fifth rule is caused by the high correlation between Total and Direct Bilirubin.
Rule two shows that around 83 % of patients that have relatively high levels of Direct Bilirubin (over median) are male.
Rule three shows that patients with high direct_bilirubin in 81 % of cases are also sick.
Finally rule six says sick patients are in 78 % of cases also male.

From this first mining we see that Total and Direct Bilirubin are much like in report 2 again the most 'interesting' of the attributes, where etc. Direct Bilirubin was the most important feature to the PCA. We also see (which is proved further when lowering the thresholds) that many attributes associate with being male, simply because there are many male patients in the data set, and we hereby both find that these may be less of an interest in this analysis. We also see that there may be a problem with having big size differences in how etc. gender is divided.

To get a wider picture of associations, the Apriori algorithm is run again with lower support threshold; *minsup = 0.40* and *minconf = 0.75*, which produces 31 (25 more) different rules. The most interesting of these being:

$$\begin{gathered}
5 : \{ag\_ratio\_50 - 100perc.\} \rightarrow \{is\_patient\_Sick\}(supp : 0.418, conf : 0.832)\\
7 : \{age\_50 - 100perc.\} \rightarrow \{is\_patient\_Sick\}(supp : 0.406, conf : 0.768)\\
9 : \{albumin\_50 - 100perc.\} \rightarrow \{is\_patient\_Sick\}(supp : 0.428, conf : 0.835)\\
10 : \{alkphos\_50 - 100perc.\} \rightarrow \{sgot\_50 - 100perc.\}(supp : 0.408, conf : 0.811)
\end{gathered}$$

The first three of these show associations between having high ag_ratio, being old and having high levels of Albumin with being sick, at high levels of confidence. The last interesting rule shows an association between having high levels of Alkphos and sgot, two not directly linked medical measurements, with confidence of 81 %.

For the third and final time the thresholds are lowered to; *minsup = 0.35* and *minconf = 0.75*, which produces 63 (32 more) different rules. As there are strong correlations between some of the attributes as found in previous reports, here we look exclusively to the associations predicting whether a patient has a disease or not:

$$\begin{gathered}
13 : \{alkphos\_0 - 50perc.\} \rightarrow \{is\_patient\_Sick\}(supp : 0.387, conf : 0.778)\\
37 : \{albumin\_50 - 100perc., ag\_ratio\_50 - 100perc.\} \rightarrow \{is\_patient\_Sick\}
\end{gathered}$$

(supp: 0.351, conf: 0.871)

$$42 : \{ag\_ratio\_50 - 100perc., gender\_Male\} \rightarrow \{is\_patient\_Sick\}(supp : 0.352, conf : 0.843)$$

$$46 : \{albumin\_50 - 100perc., gender\_Male\} \rightarrow \{is\_patient\_Sick\}(supp : 0.363, conf : 0.847)$$

$$48 : \{direct\_bilirubin\_50 - 100perc., gender\_Male\} \rightarrow \{is\_patient\_Sick\}$$

(supp: 0.382, conf: 0.819)

$$56 : \{tot\_bilirubin\_50 - 100perc., gender\_Male\} \rightarrow \{is\_patient\_Sick\}$$

(supp: 0.370, conf: 0.826)

$$61 : \{tot\_bilirubin\_50 - 100perc., direct\_bilirubin\_50 - 100perc., gender\_Male\} \rightarrow \{is\_patient\_Sick\}$$

(supp: 0.364, conf: 0.824)

$$58 : \{tot\_proteins\_50 - 100perc., gender\_Male\} \rightarrow \{is\_patient\_Sick\}(supp : 0.351, conf : 0.853)$$

Generally it can be understood that higher levels of many of the medical compounds measured in the patients often associate to being sick, and definitely not the other way around.

Implement association rules

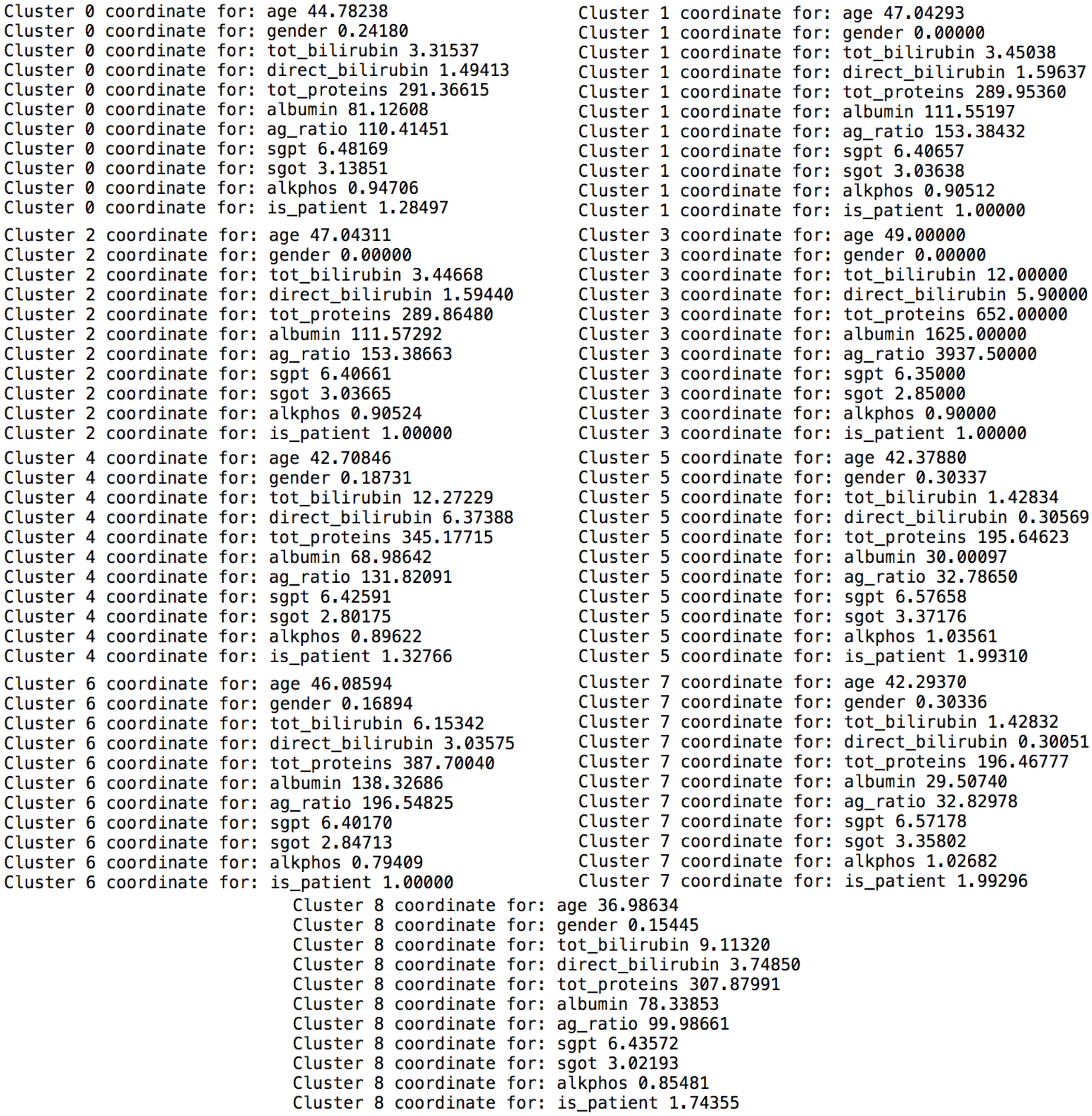
The most noticeable thing about the generated association rules is that they all indicate, that higher than median levels of most of the measured compounds (attributes) is strongly associated with having a liver disease. Therefor, when implementing the association rules for predicting whether a patient is sick or not, looking for these high levels is a very good starting point. Out of all the measurements indicating sickness, Direct Bilirubin has the highest support of 0.454, though the confidence of the association is actually lower than that of ag_ratio as well as Albumin. In practise, this means that more people will have higher Direct Bilirubin levels than the other two, but when high levels of ag_ratio or Albumin it serves as a bigger indication that the person will be sick.

Commenting on the larger size of male patients:
The problem with the vast majority of patients being male is that, for there to be any rules generated that associates to being female - maybe high levels of Albumin in females are not linked with being sick - the support threshold has to be lowered to 0.2, in which case a very large number of rules, 812, is generated that is hard to get a grasp around.

Appendix

Attribute:	**Mean μ:**	**STD σ:**	**Range:**	**Q1**	**50%**	**Q3**
age	44.782	16.222	4.0 - 90.0	33.0	45.0	58.0
gender	NaN	NaN	Male/Female	NaN	NaN	NaN
tot_bilirubin	3.315	6.228	0.4 - 75.0	0.8	1.0	2.6
direct_bilirubin	1.494	2.816	0.1 - 19.7	0.2	0.3	1.3
tot_proteins	291.366	243.562	63.0 - 2110.0	175.5	208.0	298.0
albumin	81.126	183.183	10.0 - 2000.0	23.0	35.0	61.0
ag_ratio	110.415	289.850	10.0 - 4929.0	25.0	42.0	87.0
sgpt	6.482	1.085	2.7 - 9.6	5.8	6.6	7.2
sgot	3.139	0.794	0.9 - 5.5	2.6	3.1	3.8
alkphos	0.947	0.320	0.3 - 2.8	0.7	0.9	1.1
is_patient	NaN	NaN	Yes/No	NaN	NaN	NaN

Table 4: Statistical summary of attributes



```
Cluster 0 coordinate for: age 44.78238
Cluster 0 coordinate for: gender 0.24180
Cluster 0 coordinate for: tot_bilirubin 3.31537
Cluster 0 coordinate for: direct_bilirubin 1.49413
Cluster 0 coordinate for: tot_proteins 291.36615
Cluster 0 coordinate for: albumin 81.12608
Cluster 0 coordinate for: ag_ratio 110.41451
Cluster 0 coordinate for: sgpt 6.48169
Cluster 0 coordinate for: sgot 3.13851
Cluster 0 coordinate for: alkphos 0.94706
Cluster 0 coordinate for: is_patient 1.28497

Cluster 1 coordinate for: age 47.04293
Cluster 1 coordinate for: gender 0.00000
Cluster 1 coordinate for: tot_bilirubin 3.45038
Cluster 1 coordinate for: direct_bilirubin 1.59637
Cluster 1 coordinate for: tot_proteins 289.95360
Cluster 1 coordinate for: albumin 111.55197
Cluster 1 coordinate for: ag_ratio 153.38432
Cluster 1 coordinate for: sgpt 6.40657
Cluster 1 coordinate for: sgot 3.03638
Cluster 1 coordinate for: alkphos 0.90512
Cluster 1 coordinate for: is_patient 1.00000

Cluster 2 coordinate for: age 47.04311
Cluster 2 coordinate for: gender 0.00000
Cluster 2 coordinate for: tot_bilirubin 3.44668
Cluster 2 coordinate for: direct_bilirubin 1.59440
Cluster 2 coordinate for: tot_proteins 289.86480
Cluster 2 coordinate for: albumin 111.57292
Cluster 2 coordinate for: ag_ratio 153.38663
Cluster 2 coordinate for: sgpt 6.40661
Cluster 2 coordinate for: sgot 3.03665
Cluster 2 coordinate for: alkphos 0.90524
Cluster 2 coordinate for: is_patient 1.00000

Cluster 3 coordinate for: age 49.00000
Cluster 3 coordinate for: gender 0.00000
Cluster 3 coordinate for: tot_bilirubin 12.00000
Cluster 3 coordinate for: direct_bilirubin 5.90000
Cluster 3 coordinate for: tot_proteins 652.00000
Cluster 3 coordinate for: albumin 1625.00000
Cluster 3 coordinate for: ag_ratio 3937.50000
Cluster 3 coordinate for: sgpt 6.35000
Cluster 3 coordinate for: sgot 2.85000
Cluster 3 coordinate for: alkphos 0.90000
Cluster 3 coordinate for: is_patient 1.00000

Cluster 4 coordinate for: age 42.70846
Cluster 4 coordinate for: gender 0.18731
Cluster 4 coordinate for: tot_bilirubin 12.27229
Cluster 4 coordinate for: direct_bilirubin 6.37388
Cluster 4 coordinate for: tot_proteins 345.17715
Cluster 4 coordinate for: albumin 68.98642
Cluster 4 coordinate for: ag_ratio 131.82091
Cluster 4 coordinate for: sgpt 6.42591
Cluster 4 coordinate for: sgot 2.80175
Cluster 4 coordinate for: alkphos 0.89622
Cluster 4 coordinate for: is_patient 1.32766

Cluster 5 coordinate for: age 42.37880
Cluster 5 coordinate for: gender 0.30337
Cluster 5 coordinate for: tot_bilirubin 1.42834
Cluster 5 coordinate for: direct_bilirubin 0.30569
Cluster 5 coordinate for: tot_proteins 195.64623
Cluster 5 coordinate for: albumin 30.00097
Cluster 5 coordinate for: ag_ratio 32.78650
Cluster 5 coordinate for: sgpt 6.57658
Cluster 5 coordinate for: sgot 3.37176
Cluster 5 coordinate for: alkphos 1.03561
Cluster 5 coordinate for: is_patient 1.99310

Cluster 6 coordinate for: age 46.08594
Cluster 6 coordinate for: gender 0.16894
Cluster 6 coordinate for: tot_bilirubin 6.15342
Cluster 6 coordinate for: direct_bilirubin 3.03575
Cluster 6 coordinate for: tot_proteins 387.70040
Cluster 6 coordinate for: albumin 138.32686
Cluster 6 coordinate for: ag_ratio 196.54825
Cluster 6 coordinate for: sgpt 6.40170
Cluster 6 coordinate for: sgot 2.84713
Cluster 6 coordinate for: alkphos 0.79409
Cluster 6 coordinate for: is_patient 1.00000

Cluster 7 coordinate for: age 42.29370
Cluster 7 coordinate for: gender 0.30336
Cluster 7 coordinate for: tot_bilirubin 1.42832
Cluster 7 coordinate for: direct_bilirubin 0.30051
Cluster 7 coordinate for: tot_proteins 196.46777
Cluster 7 coordinate for: albumin 29.50740
Cluster 7 coordinate for: ag_ratio 32.82978
Cluster 7 coordinate for: sgpt 6.57178
Cluster 7 coordinate for: sgot 3.35802
Cluster 7 coordinate for: alkphos 1.02682
Cluster 7 coordinate for: is_patient 1.99296

Cluster 8 coordinate for: age 36.98634
Cluster 8 coordinate for: gender 0.15445
Cluster 8 coordinate for: tot_bilirubin 9.11320
Cluster 8 coordinate for: direct_bilirubin 3.74850
Cluster 8 coordinate for: tot_proteins 307.87991
Cluster 8 coordinate for: albumin 78.33853
Cluster 8 coordinate for: ag_ratio 99.98661
Cluster 8 coordinate for: sgpt 6.43572
Cluster 8 coordinate for: sgot 3.02193
Cluster 8 coordinate for: alkphos 0.85481
Cluster 8 coordinate for: is_patient 1.74355
```

Figure 4: Cluster centers from GMM

Table 5: Responsibility table

Section\Name	Mads C. B. Andersen s173934	Johannes Boe Reiche s175549	AugustS. Andersen s183918
Introduction	O		X
Clustering		X	
Hierarchical clustering		X	
Gaussian Mixture Model		X	
Evaluation of clustering methods		X	
Outlier detection / Anomaly detection	X		
Rank	X		
Discuss	X		
Association mining			X
Apriori algo			X
Implement association rules			X
Editing	O	O	O

The person in charge of a section is marked with a 'X' and have been at least 50% responsible for the section. Co-editor(s) have received an 'O'.