Project 3

Valentin Rosenberg, Zacharias Knudsen

May 6, 2014

1 Dataset

Already during project 1 we noticed that the dataset could have some anomalous objects (by looking at for example a plot of the first two principal components, see figure 1). Interestingly, another version of the dataset was available, by Redmond[1], which has been preprocessed in a number of ways; first each attribute has been normalized into a 0-1 interval, and then values more than 3 standard deviations above the mean are normalized to 1 while values more than 3 std. dev. below the mean are standardized to 0.

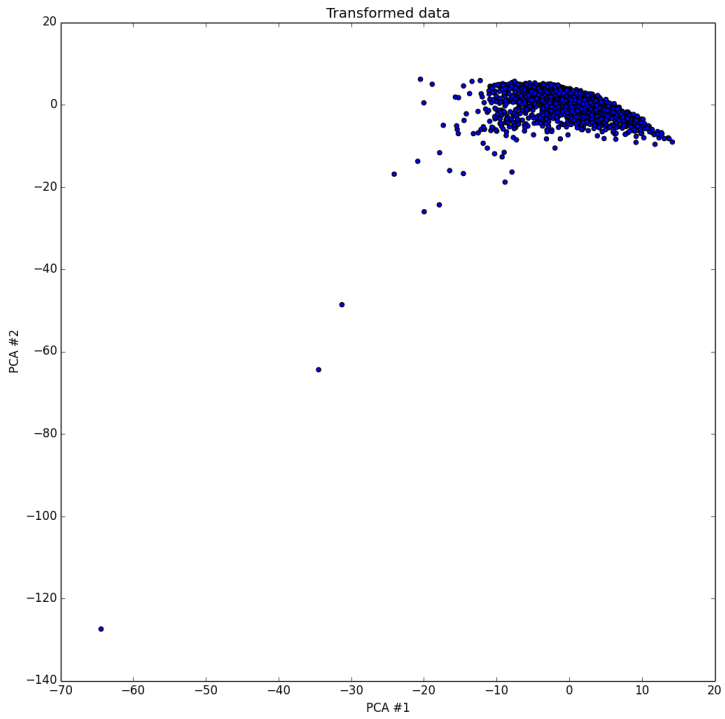

Figure 1: figure
Plot of first two principal components computed on the (standardized) raw dataset

Whether we use the preprocessed dataset by [1] depends on what machine learning task is being performed. Unless otherwise specified, we use this preprocessed dataset.

2 Clustering

2.1 GMM

We will find clusters of a subset of our dataset, as the dataset is meant for regression, many attributes have little or no correlation. We want to examine if there are clusters in the following demographic, social and economic attributes:
racePctHisp, racePctWhite: The percentage of hispanics and whites in the community population respectively.
medIncome: The median income of the community.
NumStreet: The number of people living on the street.
NumImmig: The number of immigrants.
PctEmploy: The percentage of employed people.
PctPopUnderPov: The percentage under the poverty limit.
pctUrban: The percentage of people living in urban areas.

The data is fitted with a Gaussian Mixture Model and the AIC and BIC scores are calculated for each model as shown below. Each attribute is standardized and 200 objects are chosen for better runningtimes.

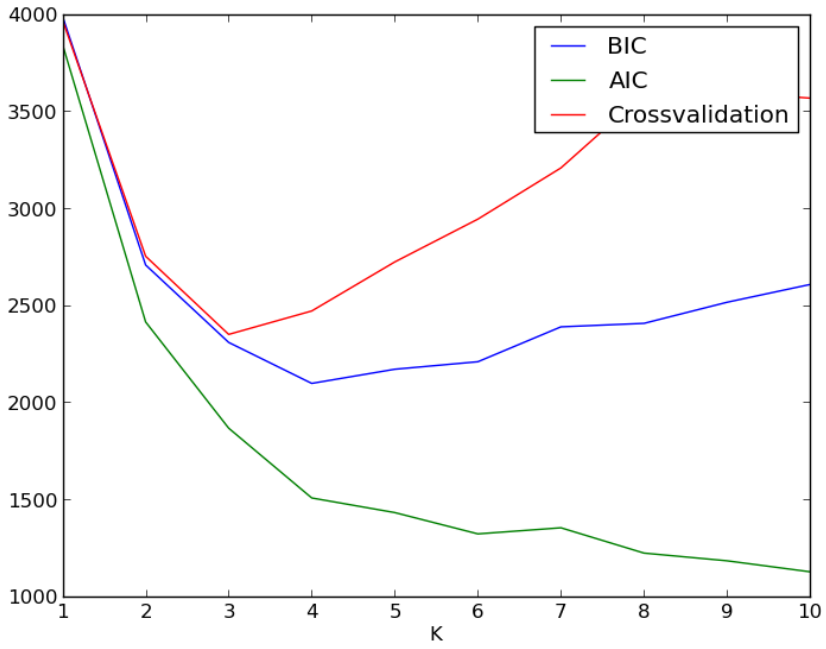

Figure 2: 10-fold crossvalidation, BIC and AIC for 1 to 10 clusters with a GMM. The vertical axis is the BIC, AIC and crossvalidation scores respectiveliy - crossvalidation is proportional to the negative log likelihood

The the crossvalidation score clearly has a minimum at 3 clusters, where BIC has a minimum at 4. AIC seems to not find its minimum at these numbers of clusters, maybe because of the high number of data objects. We chose 3 clusters as the cluster centres made the most sense.

The cluster centres are extracted and are as shown below.

```
['racePctHisp' 'racePctWhite' 'medIncome' 'NumStreet' 'NumImmig' 'PctEmploy'
 'PctPopUnderPov' 'pctUrban']
[[ 0.53793919 -0.73806562 -0.94107439 -0.05433901 -0.20753335 -0.94003235
   1.35819032 -1.02686909]
 [-0.4669858   0.58872669  0.39039555 -0.22532924 -0.27917154  0.14904979
```

```
  -0.49926193 -0.14320119]
 [ 0.51819432 -0.35295837 -0.00963756  0.77525982  0.51338927  0.19084485
  -0.09279508  0.68093829]]
```

It seems the first cluster is one with many hispanics, low income, low employment, many living in poverty, and low urbanisation. The second cluster has a high number of whites, high income and few people under poverty limit. The third cluster is more interesting with many hispanics, normal income, high number of people on the street, high number of immigrants and high degree urbanization.

2.2 Hierarchical

The data is also clustered with a hierarchical method, using euclidian distances as distance metric and a complete linkage function. The complete linkage function is chosen because we want well seperated clusters representing groupings in the communities. Also single linkage led to there being only one cluster, probably due to the chaining phenomenon.

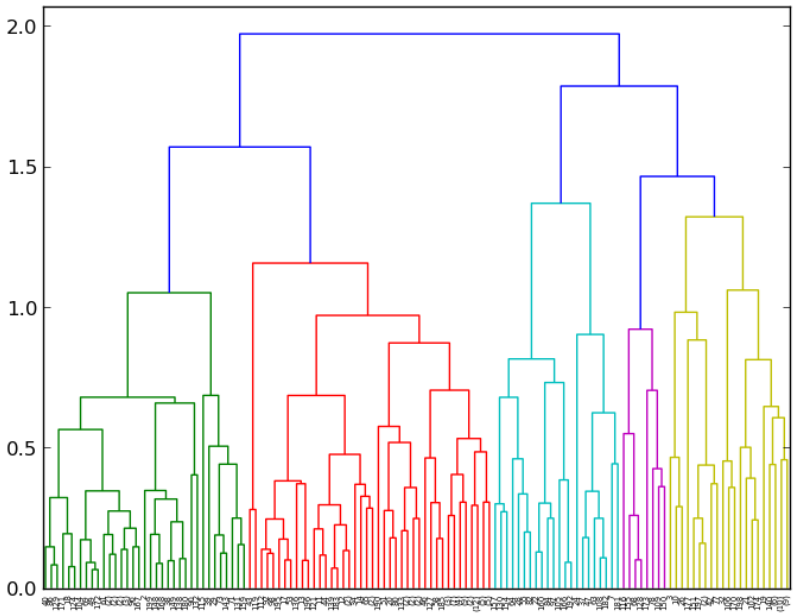

Figure 3: Dendrogram of hierarchical clustering

2.3 Evaluation of clusters

We split the racePctWhite attribute into two classes deviding it by the median, as this might give a real separation of communities in these attributes given the history of America. If the clusters are grouped in this way, a confusion matrix will show a clear separation in the distribution of classes into the two categories.

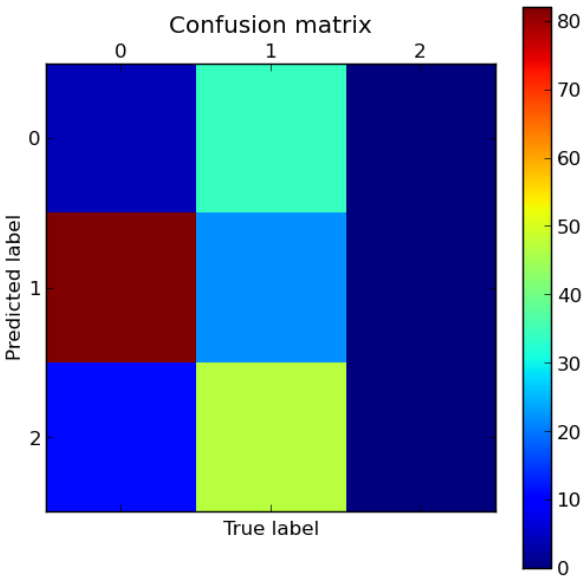

Figure 4: Confusion matrix of true values of splitting racePctWhite and predicted clusters of the GMM.

As we can see the GMM has a clear separation of the two classes as the models cluster number 1 accounts for most of the low half of the racePctWhite attribute. The two other clusters 0 and 2 account for most of the upper half.

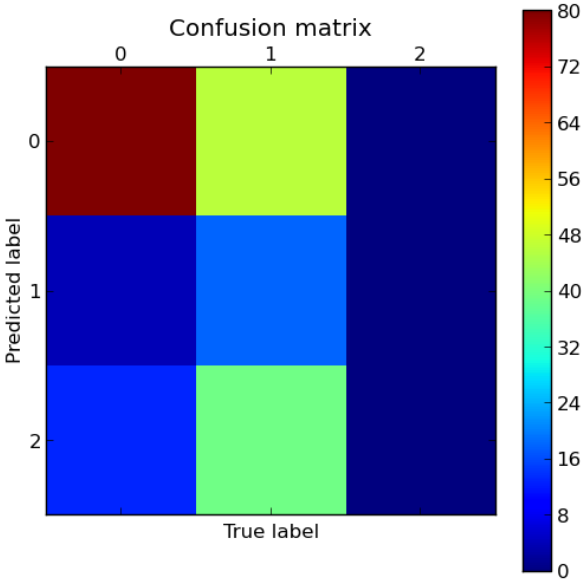

Figure 5: Confusion matrix of true values of splitting racePctWhite and predicted clusters of the hierarchical clustering.

For the hierarchical clustering most of both the lower and upper half of racePctWhite are placed in the same cluster 0. Only some of the upper half is placed in cluster 2. The hierarchical clustering seems not to capture this given separation.

2.4 Association mining

We thought it would be interesting to investigate association rules on attributes regarding race, income and violent crimes so we extracted relevant features and ran the Apriori algorithm on them.

Since our data consists mostly of ratio-attributes we had to first binarize each attribute (by setting each value to either 0 or 1 depending on whether it was above or below the median). In order to capture association rules about both low and high values, we also made duplicates of all these binarized attributes with 0 and 1 switched.

Due to the above, all item sets consisting of a single item has around 50% support (since the values are split around the median). Therefore, we should not set minimum support for the Apriori algorithm above 50. We chose to use a minimum support of 40.

Using a minimum confidence of 80, the Apriori algorithm yields the following rules:

```
Rule: HighPctPopUnderPov <- LowmedIncome (92.0782)
Rule: HighmedIncome <- LowPctPopUnderPov (92.0455)
Rule: LowmedIncome <- HighPctPopUnderPov (87.232)
Rule: LowPctPopUnderPov <- HighmedIncome (87.182)
Rule: HighracePctWhite <- Lowracepctblack (83.299)
Rule: Highracepctblack <- LowracePctWhite (83.1601)
Rule: HighViolentCrimesPerPop <- LowracePctWhite Highracepctblack (82.125)
```

Some of which are not of much interest (for example, that a community with low median income is likely to also have a high percent of population under poverty is not a big surprise). What is somewhat interesting, however, is the association rule that communities that a low percentage of white population combined with a high percentage of black population is likely to also has a high number of violent crimes per capita.

The reason we chose to extract only a few features, was that the number of found rules quickly grew out of hand when using the full dataset.

2.5 Outlier and anomaly detection

Since the preprocessed dataset by [1] already had some outlier handling done, this segment was performed on the a version of the dataset that has simply been standardized (0 mean and unit variance).

Anomaly detection was performed, by ranking objects in terms of the Gaussian Kernel density (see figure 6), K-Nearest Neighbours density (see figure 7), K-Nearest Neighbours Average Relative density (see figure 8) and distance to Kth nearest neighbour for K=5 (see figure 9). When all four methods agree, there is a higher probability that the anomaly is the result of some other mechanism. We also ran the outlier scoring methods in multiple iterations, by removing the object with the worst outlier score if the ranking methods all agreed.

In order to cope with missing values, we removed any objects containing such values. We also considered using the means or medians across each attribute as replacements for the missing values, but we figured that could hide potential anomalies (for example an object with extreme values in one attribute, but missing values for others which would be set to the mean or median).

object/method	Gaussian	KNN	KNN avg. rel.	5th-nearest
worst	4	4	4	4
2nd worst	17	17	17	17
3rd worst	8	291	127	291

Table 1: First iteration of anomaly ranking

The first iteration gave the results seen in table 1. All four ranking methods agreed that object 4 ("Bemidjicity") and 17 ("Glendalecity") were the most anomalous objects. However, there is some

disagreement as to the third worst object. Looking at all the graphs over outlier scores except for that of the Gaussian Kernel density estimater, we can see that the two worst objects have significantly worse scores, compared to the relative changes between the remaining 8 worst objects.

object/method	Gaussian	KNN	KNN avg. rel.	5th-nearest
worst	17	17	17	17
2nd worst	8	291	127	291
3rd worst	251	8	117	8

Table 2: Second iteration of anomaly ranking (after removal of object 4)

object/method	Gaussian	KNN	KNN avg. rel.	5th-nearest
worst	8	291	127	291
2nd worst	251	8	117	8
3rd worst	89	251	101	117

Table 3: Third iteration of anomaly ranking (after removal of object 4 and 17)

The second and third iterations were made in order to investigate whether there would be a difference between the top worst ojects found by a single rank computation, versus iteratively ranking and removing the worst object. This was not the case, though.

In conclusion, there was agreement from all four ranking methods that objects 4 and 17 were anomalous, less agreement that object 8 was anomalous, and even lesser agreement that objects 117, 251 and 291 were anomalous. As such, it might potentially be useful to remove these anomalous objects, perhaps even during preprocessing.

On a side note, running the ranking methods on the preprocessed dataset, not even the first iteration agreed on an anomalous object, which was to be expected since outiers had already been handled in that dataset.

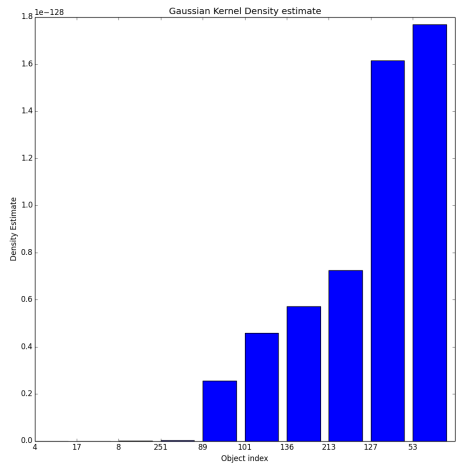

Figure 6: figure
Plot of 10 objects with worst outlier score in terms of the Gaussian Kernel density estimater

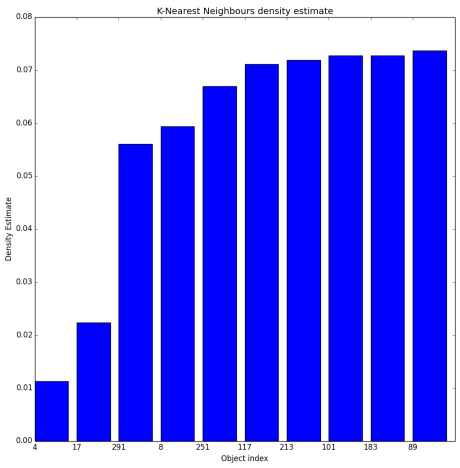

Figure 7: figure
Plot of 10 objects with worst outlier score in terms of the K-Nearest Neighbours density estimater

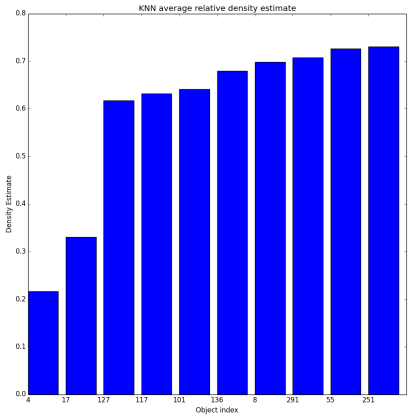

Figure 8: Plot of 10 objects with worst outlier score in terms of the K-Nearest Neighbours average relative density estimater

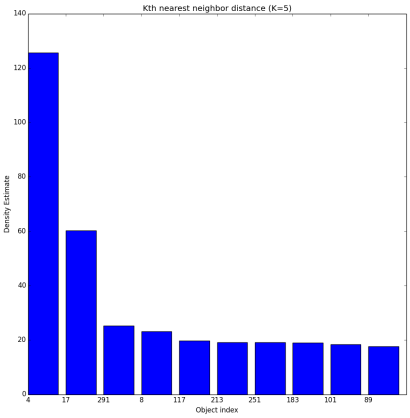

Figure 9: Plot of 10 objects with worst outlier score in terms of the Kth Neighbours distance

References

[1] Michael Redmond, *Communities and Crime*. Computer Science; La Salle University; Philadelphia, PA, 19141, USA, 2009