Danmarks Tekniske Universitet

02450 Introduction to Machine Learning and Data Modeling

Assignment 3

Clustering, association mining and anomaly detection

Poul Kjeldager Sørensen
s093294

Martin Kasban Tange
s093280

May 7th 2013

Contents

1 Introduction

This report is made in regard to the course 02450 Introduction to Machine Learning and Data Modeling at DTU (Technical University of Denmark). In this third assignment we tackle the problems of finding clusters in our data set as well as trying to cluster them based on their class. We also look at anomaly/outlier detection to see if we can find some digits which are just not normal in the way they are drawn. We have not shown any association mining as we did not see how this fit to our features at all, since they are so random.

All the work carried out in this report are done by Tange, M. K. and Sørensen, P. K. and the code can be found on github[1]. Do note that the code have only been used to generate plots and illustrations for the report and some playing around. There have been done no effort into readability or re-usability of the code.

1.1 Data

The same data as from assignment one have been used, a feature representation of the MNIST dataset. The representation consist of 272 features calculated from vertical, horizontal and radial histograms together with two profiles; in-out, out-in. Refer the first assignments for the full description.

[1] https://github.com/mktange/IntroMachineLearning

2 Clustering

2.1 Clustering within classes using PCA

We decided to first try see if any of the digits would get separated using a PCA. We found that the digit 3 and 5 was somehow separated into two classes with a few intermediate clusters.

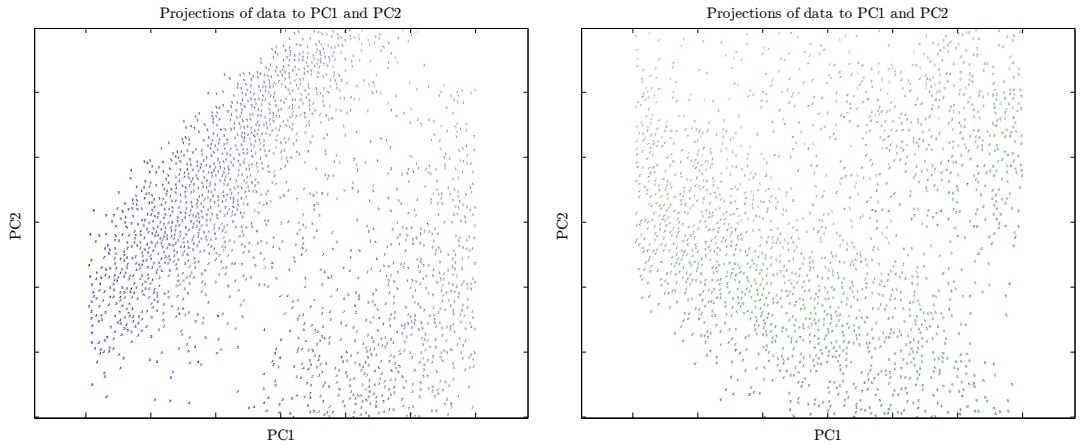

Figure 2.1

In both cases this turned out to be a separation of whether the digits was bold or thin. In figure 2.1 the clusters can be seen separated by a low density region in the middle. We made the same plot with larger digits to show what meaning the PCA components had, but this means the clusters fade due to spacing between the digits. You can always open the PDF to inspect the numbers.

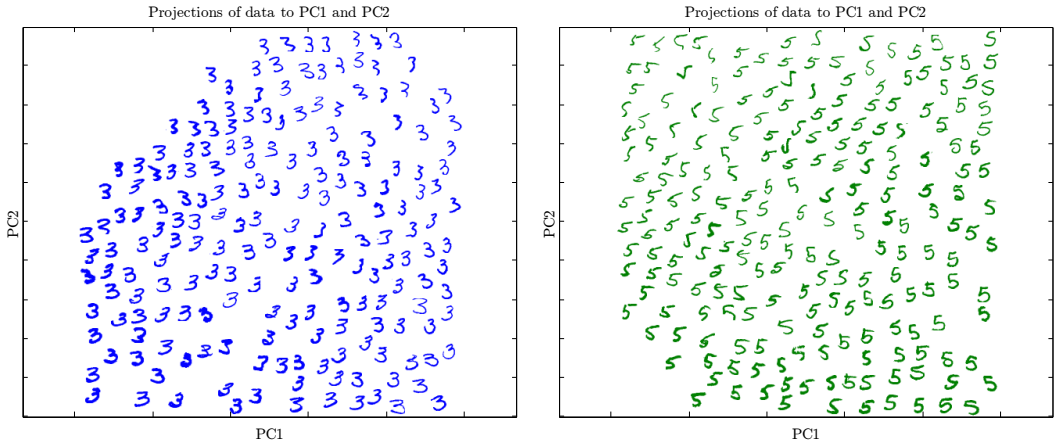

Figure 2.2

2.2 Gaussian Mixture Model (GMM) cross-validation

As asked in the assignment we have clustered our data by the Gaussian Mixture Model and used cross-validation to determine the how many clusters produces the best results.

Since our data set (again) is too big such that we are unable to run this in a decent time, we have had to take some measures to bring the run-time down. As such we have decided to use the 40 first principal components from a PCA over the digits, instead of the 272 original attributes. We have also cut the observation size down from 60.000 to 10.000, and even with all this it is still incredibly compute heavy and takes ages to run.

The result using different amount of clusters (K) is shown below.

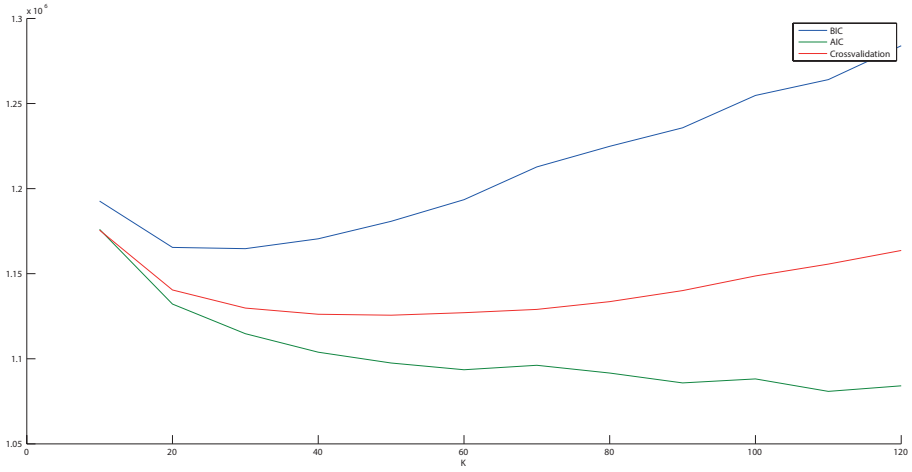

Figure 2.3: The y-axis is the negative logarithm-likelihood for each amount of clusters K. Besides cross-validation, AIC and BIC are also plotted on the graph. One can see that the maximum likelihood is around K=50 for CV.

From this we can conclude that when using this clustering method with cross-validation on all 10 of our digits, we need to use 50 clusters to get the best result.

The centers for the clusters do not hold any significant meaning because of the nature of the data set, which is why we have not done any further analysis on this.

A GMM run-through (replicates 10) with 50 clusters gave an average success rate for the clusters of 82.9% of getting the the mode of the cluster (the most frequent class in the cluster).

2.3 Hierarchical clustering

Every time we try to run the functions to do any hierarchical clustering it errors out because of too high recursion or some indexing errors. It seems that it is simply too much data for the function(s) to handle, and because of that we have not been able to get any result using this clustering technique.

2.4 GMM with matching amount of clusters

To test for actual clusters which correspond to our digit classes in the data set we set out to use exactly the amount of clusters needed depending on how many classes were used. First off we started with the digits 0 and 1, and using only the first 10 principal components we got the following clustering from GMM:

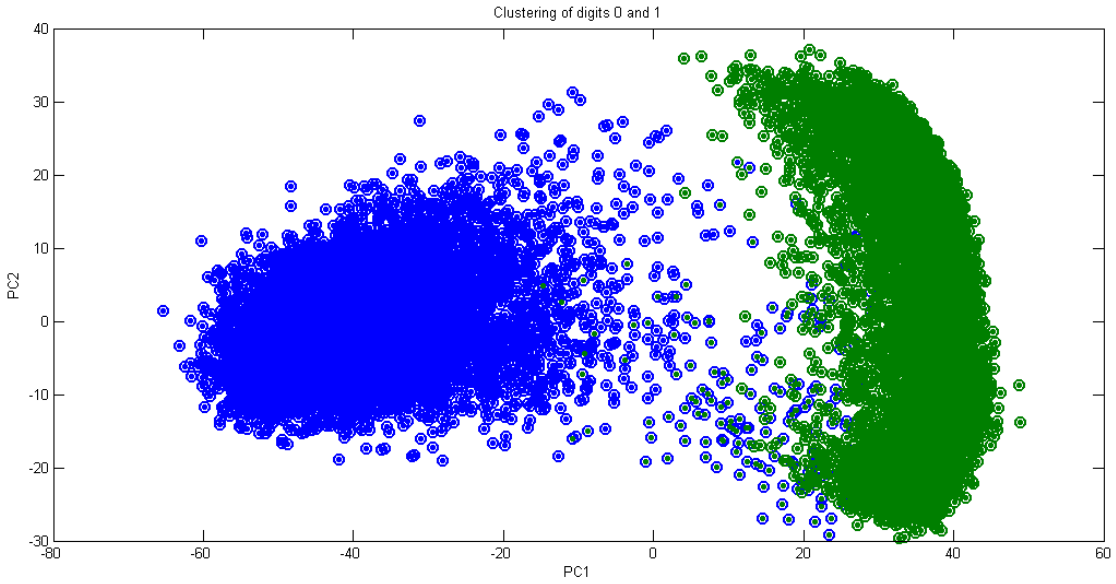

Figure 2.4: Clustering of the digits 0 and 1 shown using the two first principal components.

```
Cluster 1 has mode 0. Success rate: 5922/6081 (97.3853%)
Cluster 2 has mode 1. Success rate: 6583/6584 (99.9848%)
OVERALL: 98.7367%
```

A pretty good success rate, and it can be seen from the plot that the two first PCs split the two digit classes pretty well, which makes it ideal for clustering.

Following this we continued with more digits classes and clusters. Following $K = 3$ GMM clustering with an extra digit, namely 2. Again we used only the 10 first principal components to get this clustering and success rate:

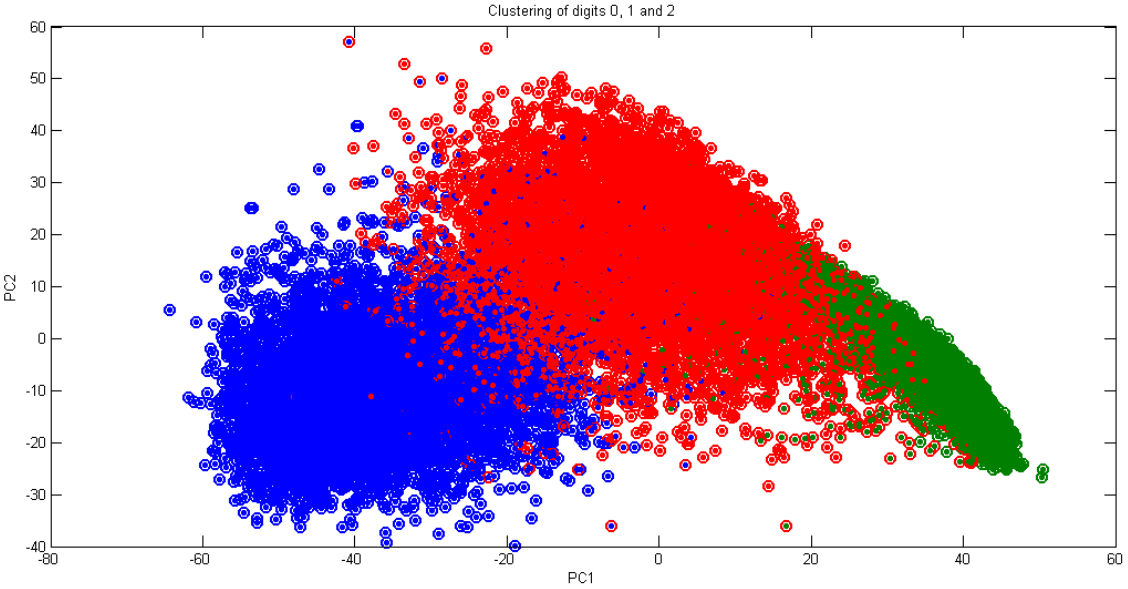

Figure 2.5: Clustering of the digits 0, 1 and 2 shown using the two first principal components.

```
Cluster 1 has mode 0. Success rate: 5679/5693 (99.7541%)
Cluster 2 has mode 1. Success rate: 6365/6365 (100.0000%)
Cluster 3 has mode 2. Success rate: 5944/6565 (90.5407%)
OVERALL: 96.5902%
```

Finally we attempted with all 10 digits and 40 PCs instead of just 10 in order to achieve a better result. This is the result of doing it with 40 PCs:

```
Cluster 1 has mode 6. Success rate: 5102/5137 (99.3187%)
Cluster 2 has mode 4. Success rate: 5310/6164 (86.1454%)
Cluster 3 has mode 8. Success rate: 2208/6233 (35.4244%)
Cluster 4 has mode 1. Success rate: 5934/5939 (99.9158%)
Cluster 5 has mode 0. Success rate: 5131/5147 (99.6891%)
Cluster 6 has mode 2. Success rate: 4117/5503 (74.8137%)
Cluster 7 has mode 8. Success rate: 3052/8561 (35.6500%)
Cluster 8 has mode 9. Success rate: 4416/4633 (95.3162%)
Cluster 9 has mode 3. Success rate: 2440/7871 (30.9999%)
Cluster 10 has mode 7. Success rate: 4711/4812 (97.9011%)
OVERALL: 70.7017%
```

The plot with PC1/PC2 is just one big glob where nothing is distinguishable, so we have left it out. From the above results we can see that some clusters capture a single class very well, namely cluster 1, 4, 5, 8 and 10 are all above 95%. This means that we are pretty good at determining the class of an observation in those areas. The rest seems to be a worse mixture of classes which 10 clusters are not good at distinguishing.

Using 50 clusters from the cross-validation part of this chapter had a better success rate, but still not a super good one.

3 Outlier/Anomaly Detection

In this chapter we looked into the topic of outlier detection. The difficulty of this topic with our data set is that people have been asked to write a digit, i.e. 5. One could argue that there are no outliers as no faulty generations of digits have been made and people simply have different ways to write a digit and many of these are similar.

Taking the definition : *An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism.* it is our hypothesis that we won't find an actual outlier, as this would mean we would have to find one single digit that distinct it self from the rest and with 60000 digits, this becomes finding a needle in a haystack.

3.1 KNN

We did an experiment with KNN and five neighbours, to show the five digits in each class that had the largest outlier score. We expected to see something we would classify as being odd digits, but still being able to predict the actual class. Evaluation about them being truth outliers cant be done, but the figure speak for it self.

Figure 3.1: The top five outliers from each class found with knn and 5 neighbours.

We didn't follow through with the GMM solutions for finding outliers since this data set is just not suitable for that kind of task. Furthermore it takes significant amount of time to fit GMM over the data set, and by filtering only some data we might enforce some outliers that are not truly outliers. Instead we did like this to see what digits in each class that scored the highest.

4 Discussion

In this report we have only solved some of the tasks asked, but we have somewhat been limited (again) by some tasks as they are not trivial on our data set, or greatly time consuming and therefore just not viable.

First off we wanted to see some clusters within each class, so we did PCA on a single digit at a time and found a few of them which had clusters. We were interested in why these were clustered as they were, so we illustrated this by putting the digit images into the plot for a nice visual effect.

Fitting the GMM was not that trivial as it required significant amount of CPU time. We tried to evaluate by only fitting a few classes and clustering them, as well as limiting it to the first few principal components, and in some cases only a subset of the observations.

In GMM clustering we also spent some time figuring out any proper way of visualizing and showing clusters. One of the tests that would make most sense with clustering would be to cluster the digits base on the classes, so we added a separate section for this which showed quite promising results.

We didn't manage to get the data transformed to binary and doing the associative mining, since we did not see this conclude anything at all with our features.

We decided it would be fun to see what digits got the highest outlier score based on KNN with 5 neighbours, expecting it to find some numbers we would as humans would classify as being a little odd. We show that this was truth, and there was some digits with some funny shapes that we would have a hard time telling to be its correct digits. But the problem about outliers is to tell when its an outlier when there are multiply of these digits that look a bit funny.

5 Conclusion

We found some nice clusters within classes and seen what exactly separates these with our features. Clustering was also done over all the classes in several different ways in order to achieve and show some results.

Afterwards we discovered some of the outliers for each class. These clearly show that they can be named an 'outlier' since they are far from the 'norm' of how that specific digit should look.

This assignment has room for improvements, since there were some sections that have been skipped out due to huge computer processing time, poor time management, and lack of relevance to our data set. But overall we tried to cover the topics as best as we could and saw fit.