Report for Sheet 1

Lab Course Machine Learning and Data Analysis

Henri Bunting (358718) Malte Seimers (378120)

May 5, 2016

Implementation comments

The goal of the assignment was to implement and play with two forms of dimensionality reduction: PCA and LLE, an evaluation method for classifiers: AUC / ROC, and a distance measure: the gamma index.

We passed all tests.

Unfortunately, we did not have time to finish Assignment 8.

Exercise 1 - Implement PCA

Exercise 1 was pretty straight forward. We got a bit confused about the orientation of each of the different matrices, but got it sorted out in the end. Initially, we used np.linalg.eig to compute the eigen vectors / values, but got imaginary results later on so we switched to np.linalg.eigh.

Exercise 2 - Implement gamma index

Gamma index went quickly. We made our own K-Nearest neighbors algorithm.

Exercise 3 - Implement AUC

We came up with two different implementations for AUC. One is with a for loop, iterating over each point as a bias, and the over was using cumsum and

taking a dot product. The non-loop implementation was of course faster, about 80% of the speed of the loop version.

Exercise 4 - Implement LLE

This was the most difficult of all the implementations. We got a bit confused what that the tolerance was the regularization parameter.

Exercise 5 - Application PCA

No noise

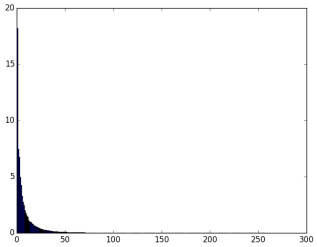

Figure 1: All PCs

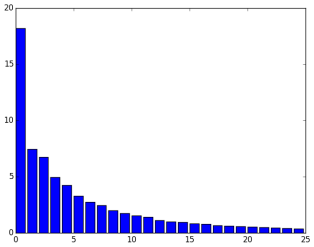

Figure 2: First 25 PCs

Figure 3: First 6 PCs

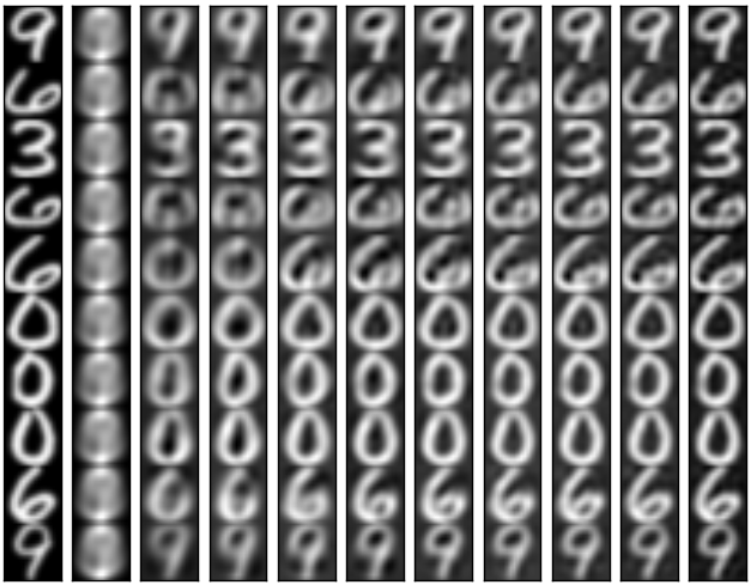

Figure 4: Orginal 10 images, followed by 0, 5, 10, ... PCs

Low noise (X + .5*noise)

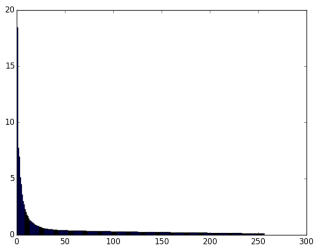

Figure 5: All PCs

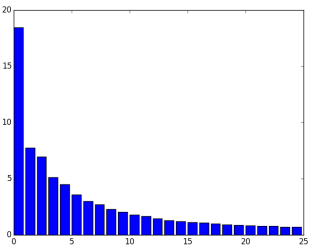

Figure 6: First 25 PCs

Figure 7: First 6 PCs

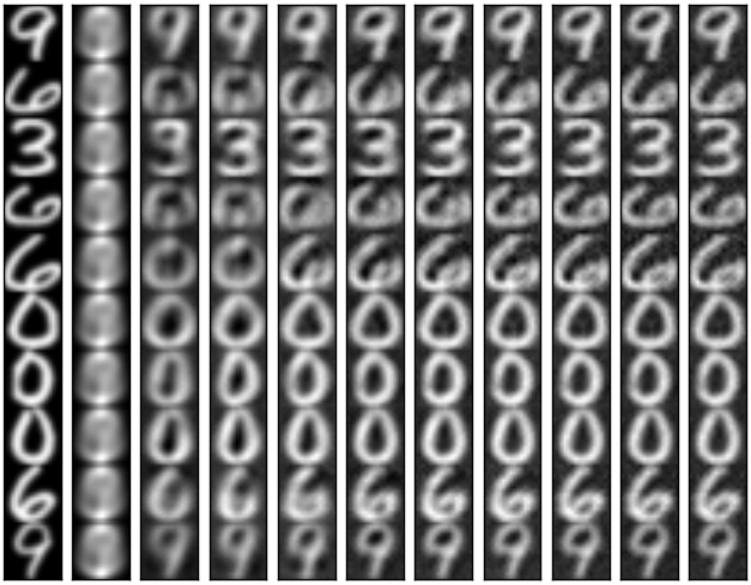

Figure 8: Orginal 10 images, followed by 0, 5, 10, ... PCs

High noise (X + 3*noise)

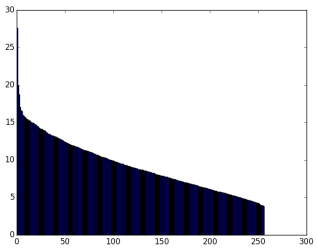

Figure 9: All PCs

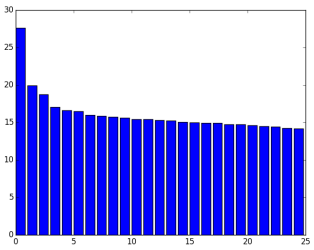

Figure 10: First 25 PCs

Figure 11: First 6 PCs

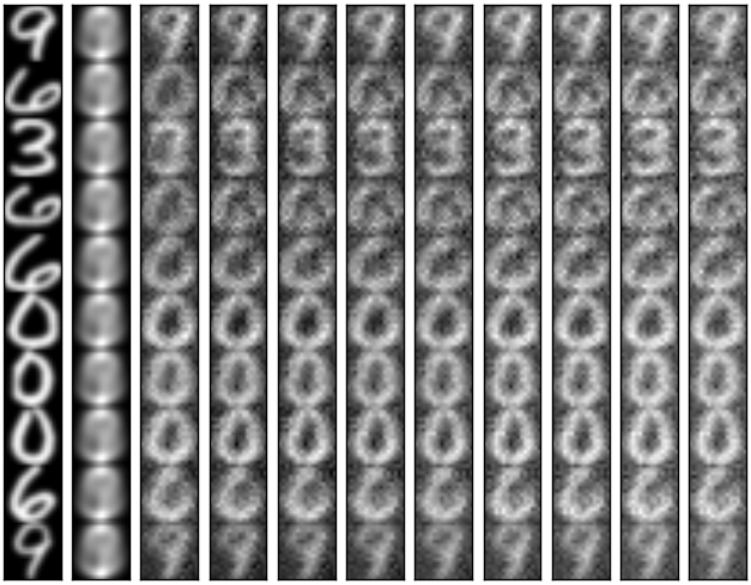

Figure 12: Orginal 10 images, followed by 0, 5, 10, ... PCs

Outliers

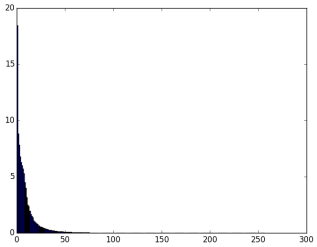

Figure 13: All PCs

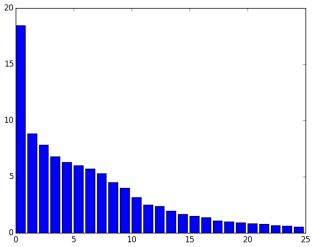

Figure 14: First 25 PCs

Figure 15: First 6 PCs

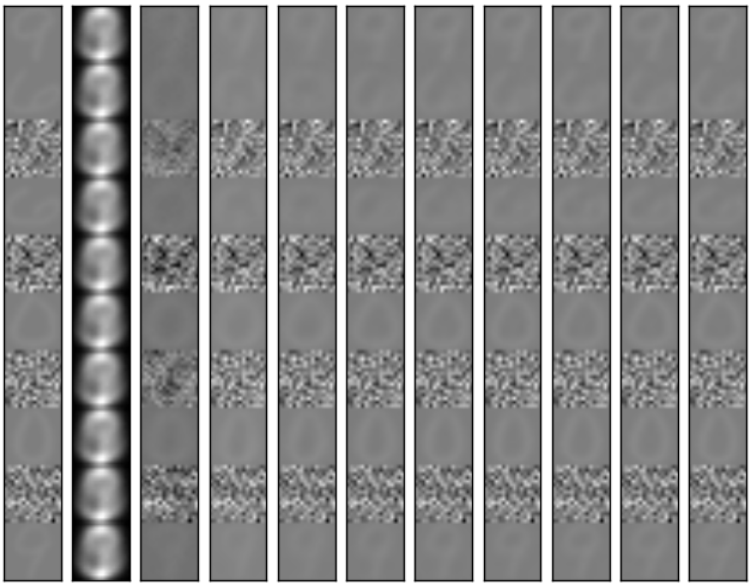

Figure 16: Orginal 10 images, followed by 0, 5, 10, ... PCs

Exercise 6 - Application AUC / Gamma Function

We had big speed issues with this one. It took 580 seconds to run each set of outliers at 100 trials each. Almost all of the time was spent in the argsort method of our k nearest neighbors algorithm. Our idea for speeding it up would be to first precompute the k nearest neighbors for all, then subset that, depending on which outliers were chosen. We didn't have time to implement this though.

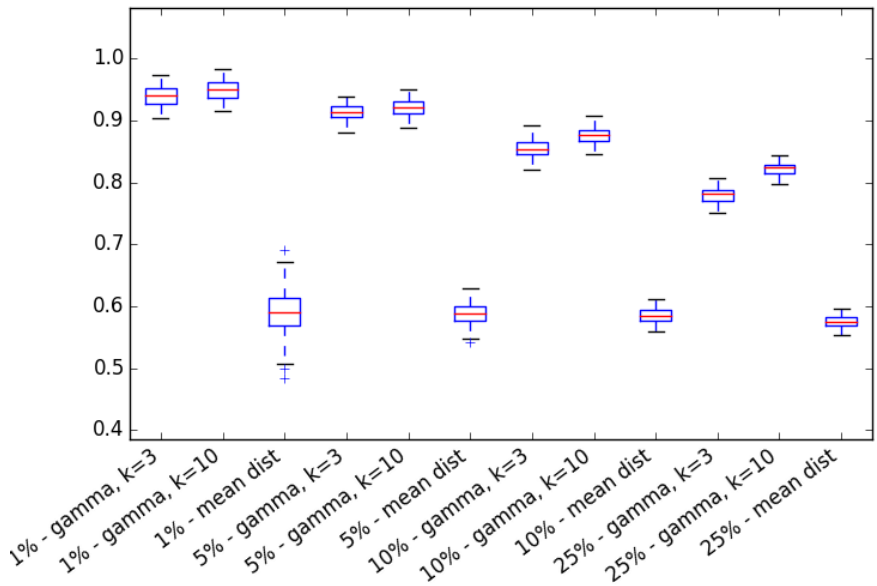

Figure 17: Outlier % and method vs AUC

Exercise 7 - Application AUC / Gamma Function

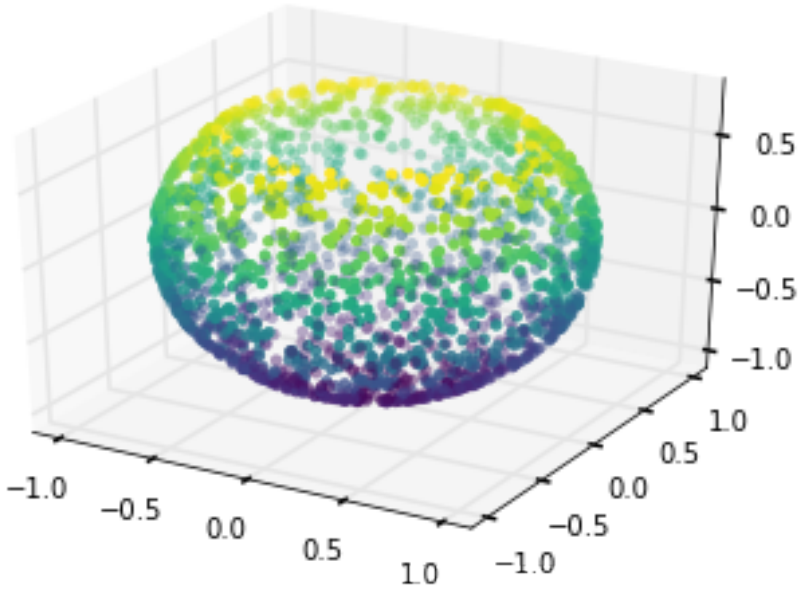

Figure 18: 7.a.0

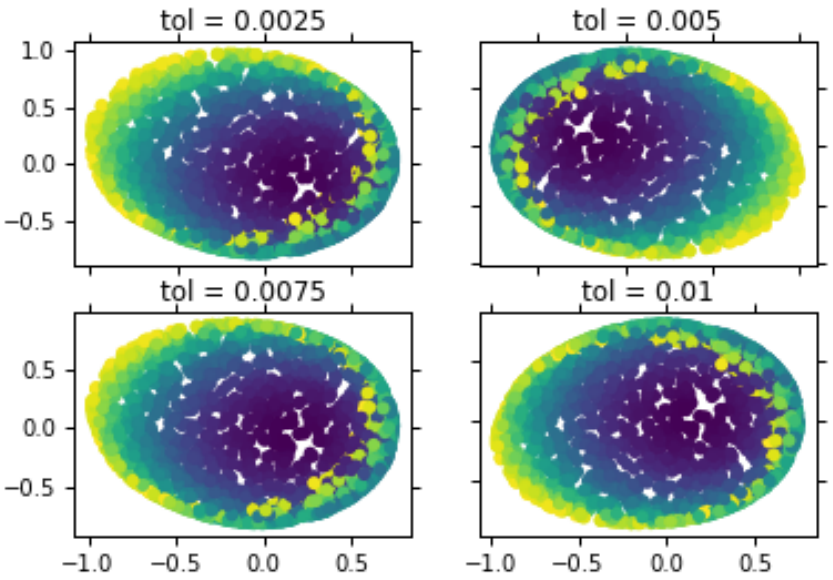

Figure 19: 7.a.1

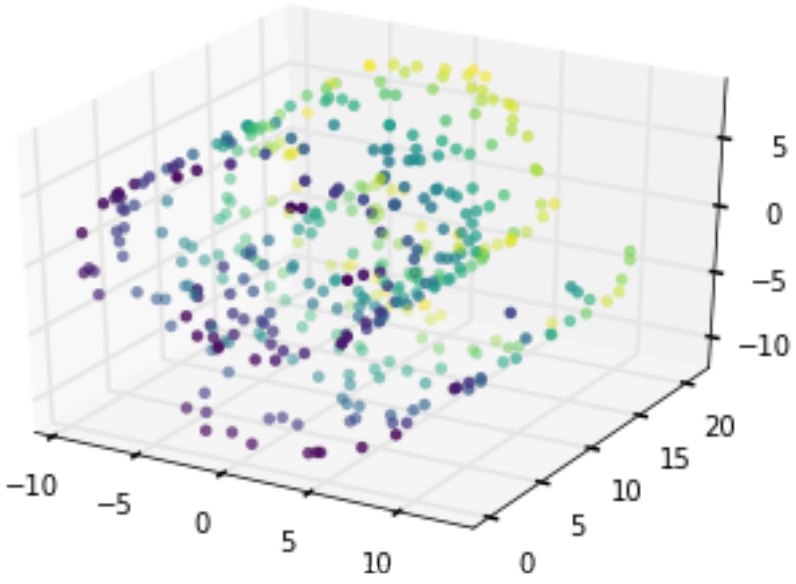

Figure 20: 7.a.2

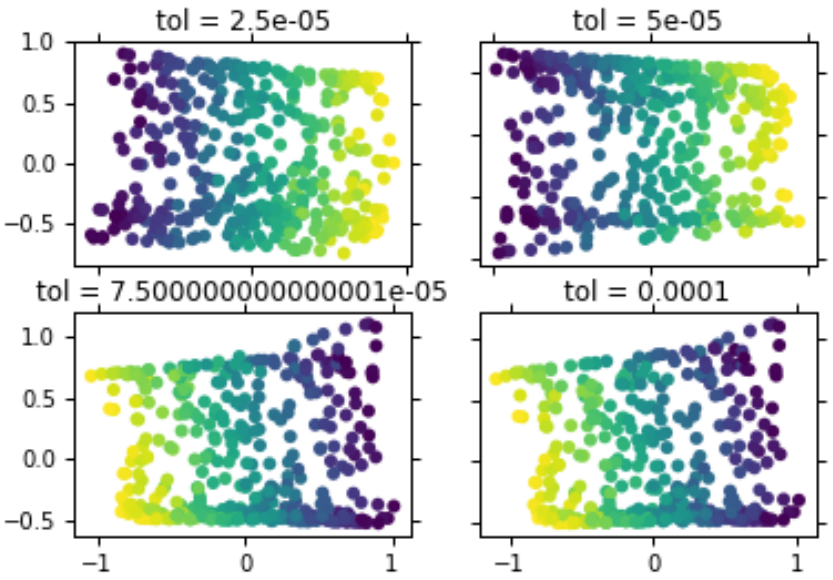

Figure 21: 7.a.3

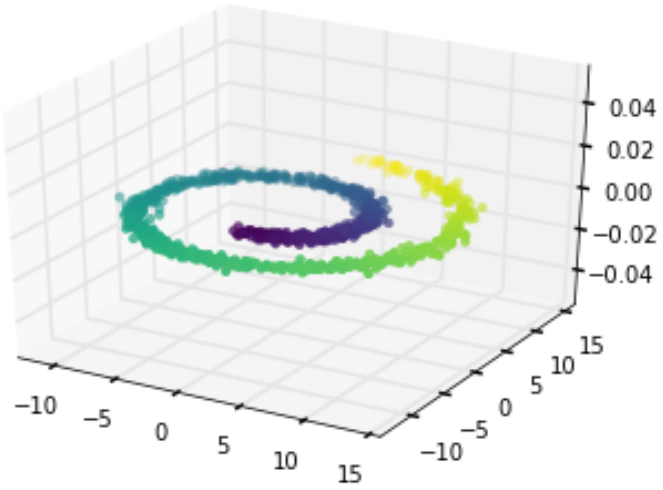

Figure 22: 7.a.4

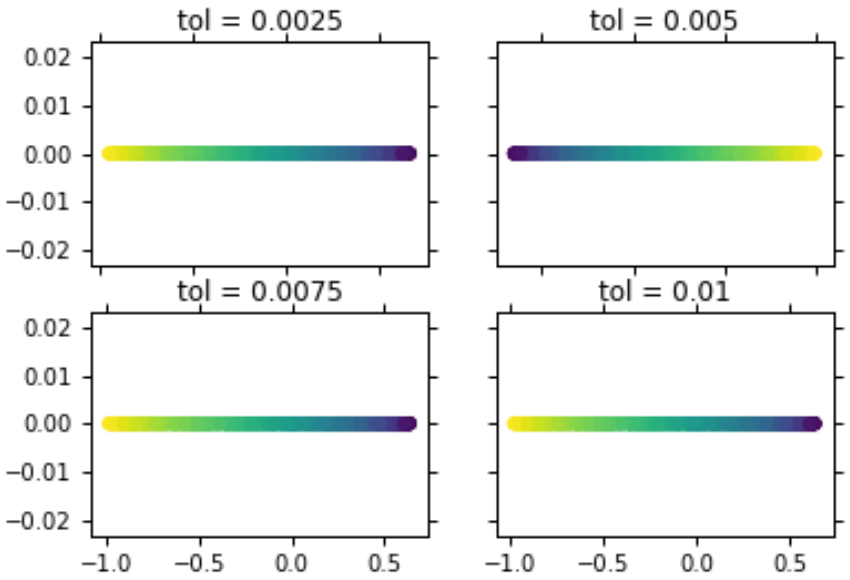

Figure 23: 7.a.5