

Machine learning lab course

Problem set 1: PCA, LLE, outlier detection

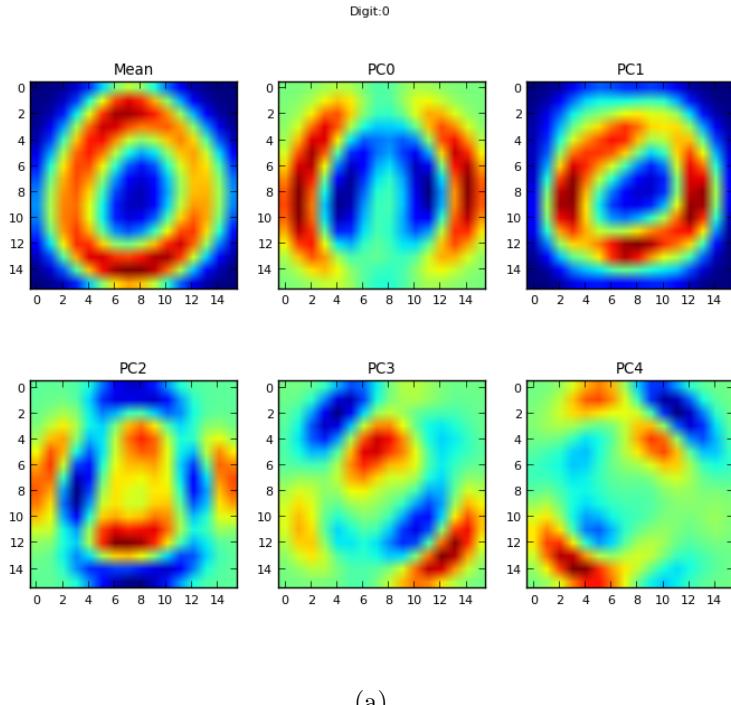
ROHRMANN Till, Matrnr: 343756

April 27, 2013

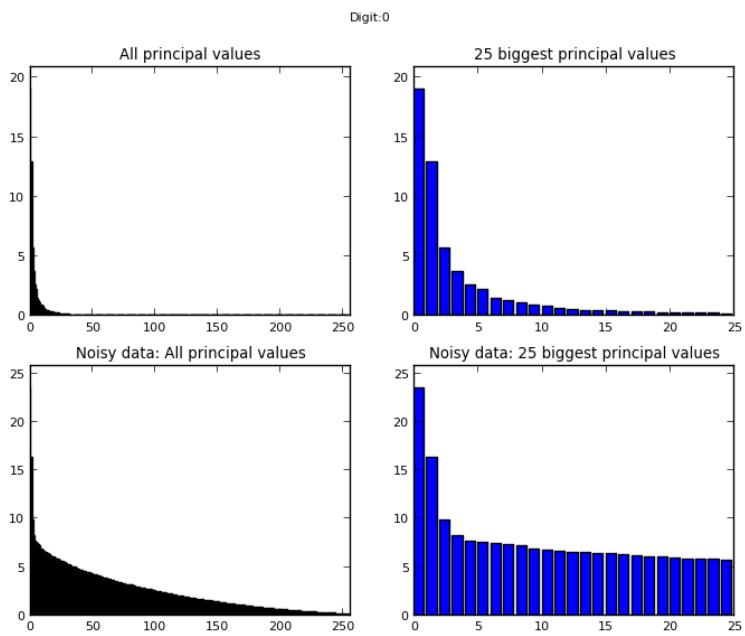
2 Application

Assignment 4

Original data

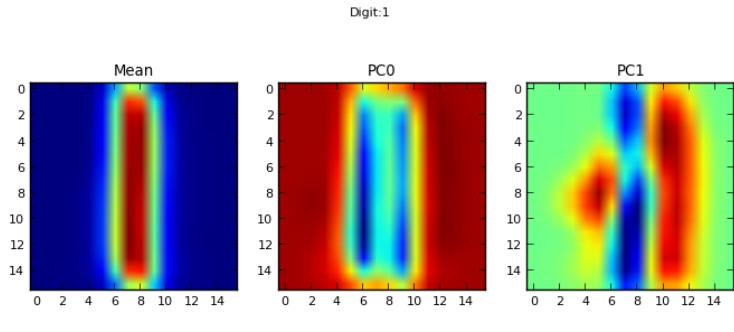


(a)

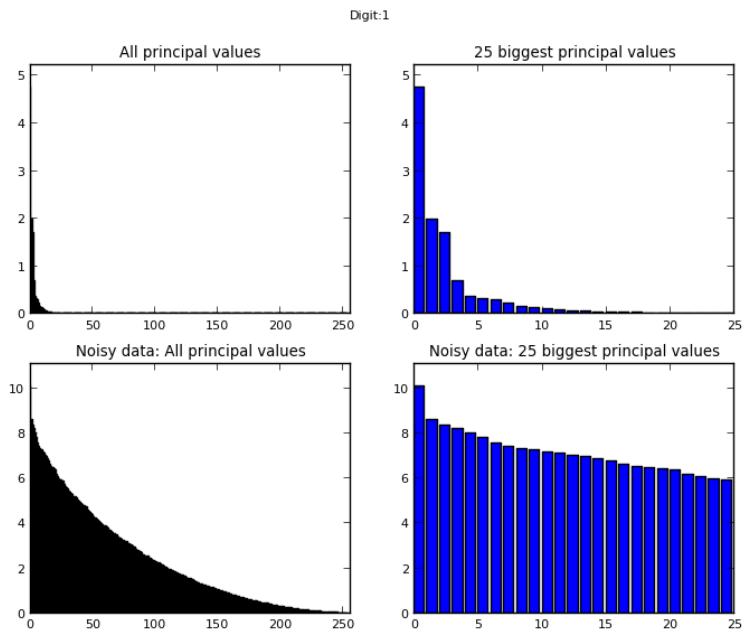


(b)

Figure 1: (a) Mean and the first 5 principal components of the digit 0. (b) Top row: All principal values and the 25 biggest principal values of the digit 0. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 0.

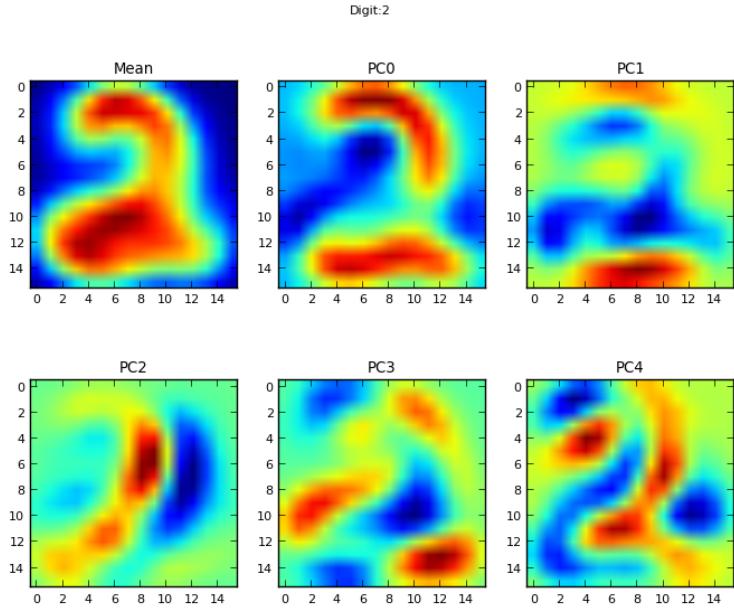


(a)

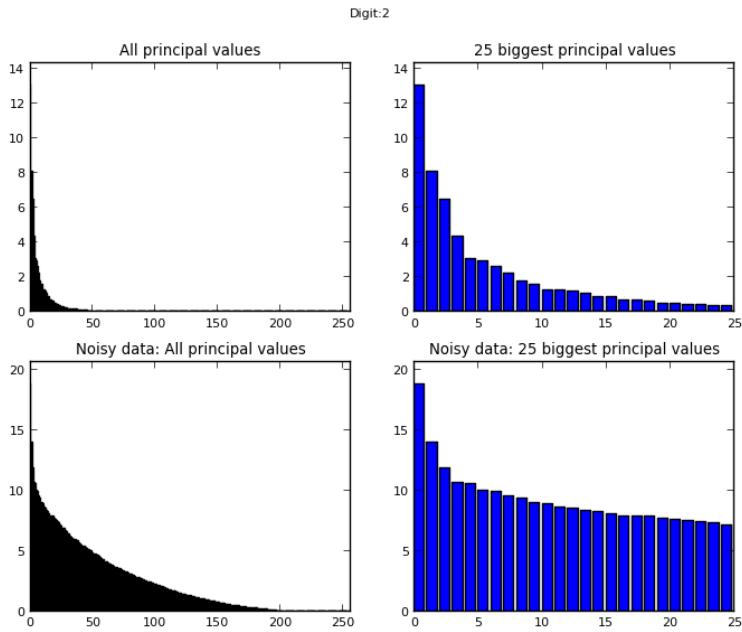


(b)

Figure 2: (a) Mean and the first 5 principal components of the digit 1. (b) Top row: All principal values and the 25 biggest principal values of the digit 1. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 1.

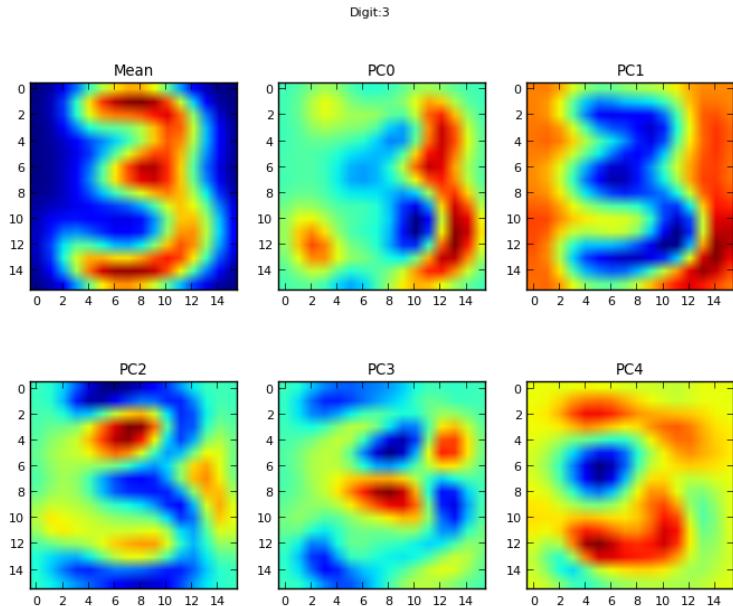


(a)

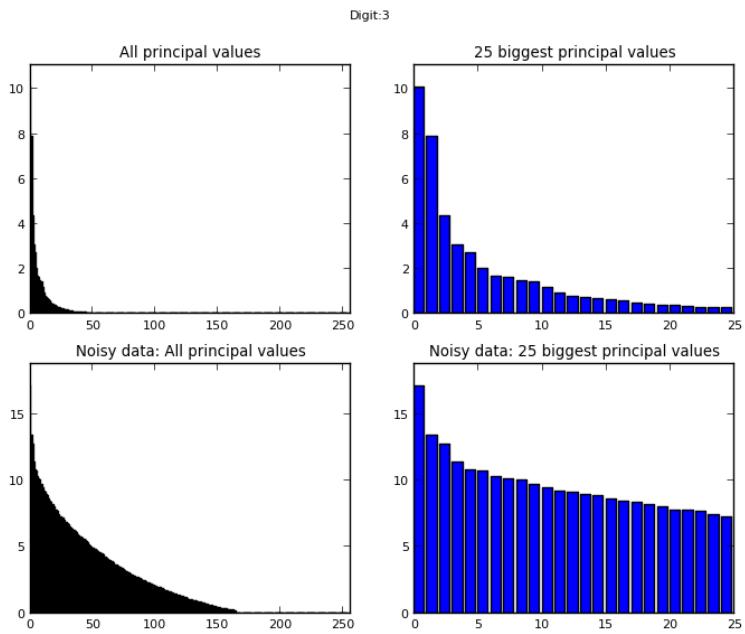


(b)

Figure 3: (a) Mean and the first 5 principal components of the digit 2. (b) Top row: All principal values and the 25 biggest principal values of the digit 2. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 2.

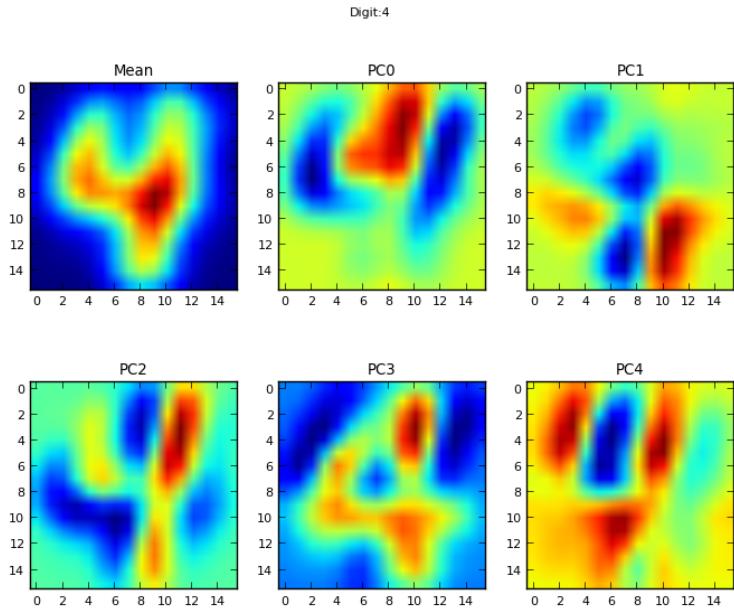


(a)

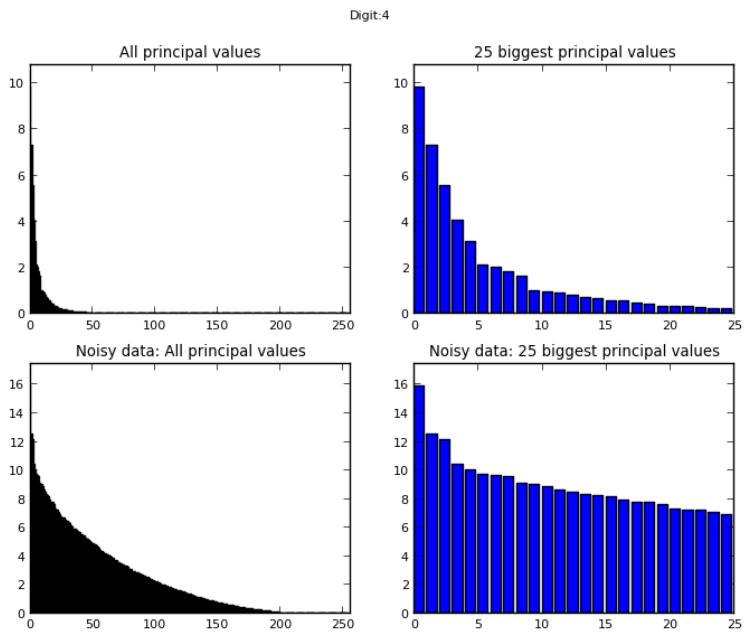


(b)

Figure 4: (a) Mean and the first 5 principal components of the digit 3. (b) Top row: All principal values and the 25 biggest principal values of the digit 3. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 3.

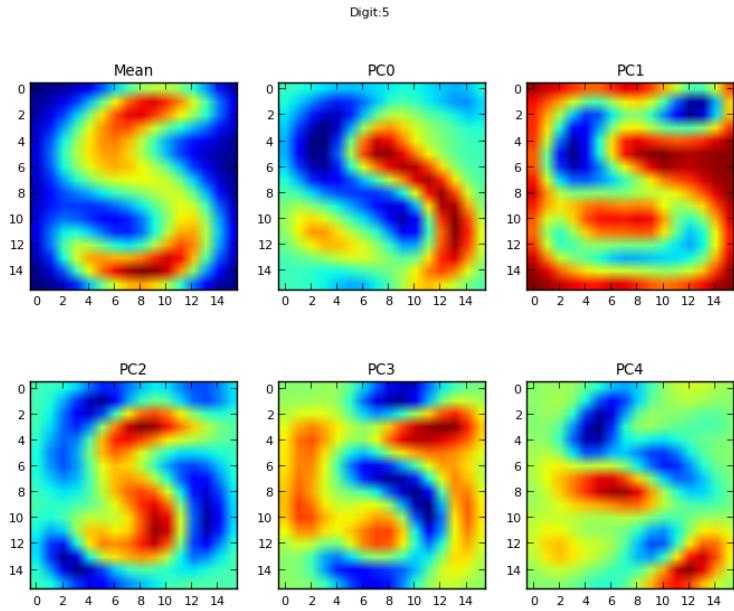


(a)

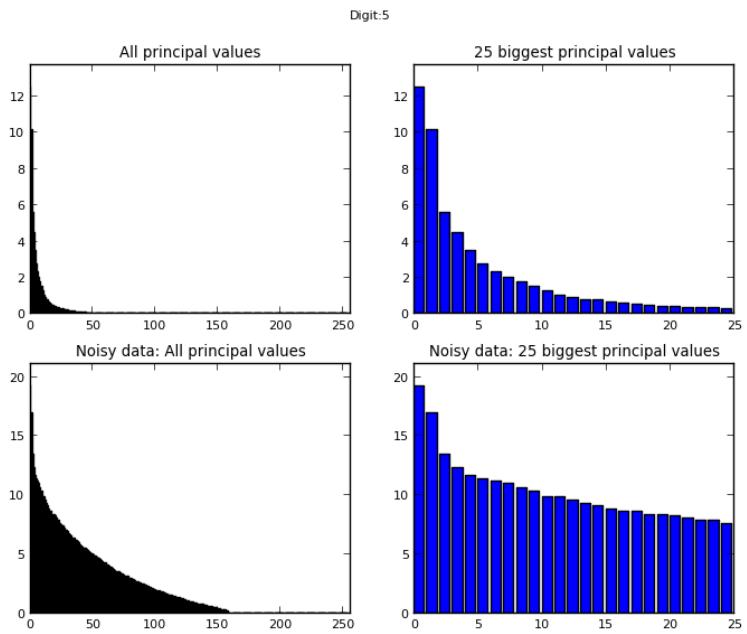


(b)

Figure 5: (a) Mean and the first 5 principal components of the digit 4. (b) Top row: All principal values and the 25 biggest principal values of the digit 4. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 4.

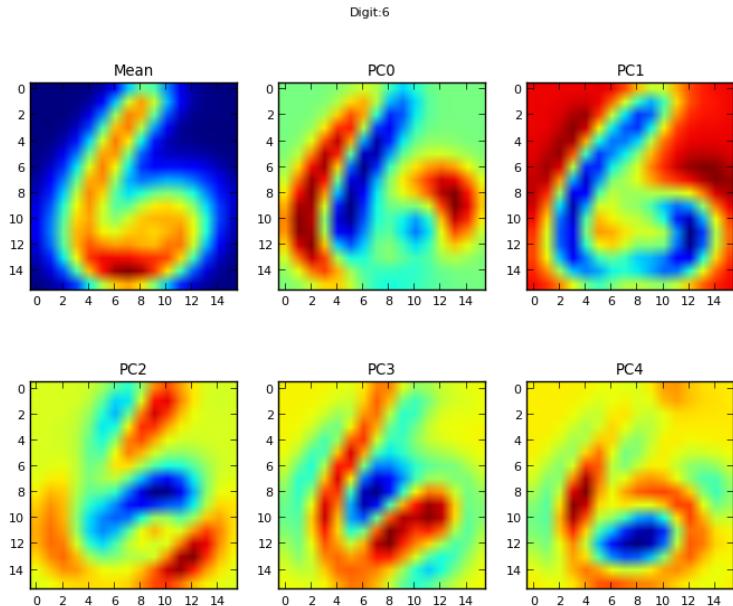


(a)

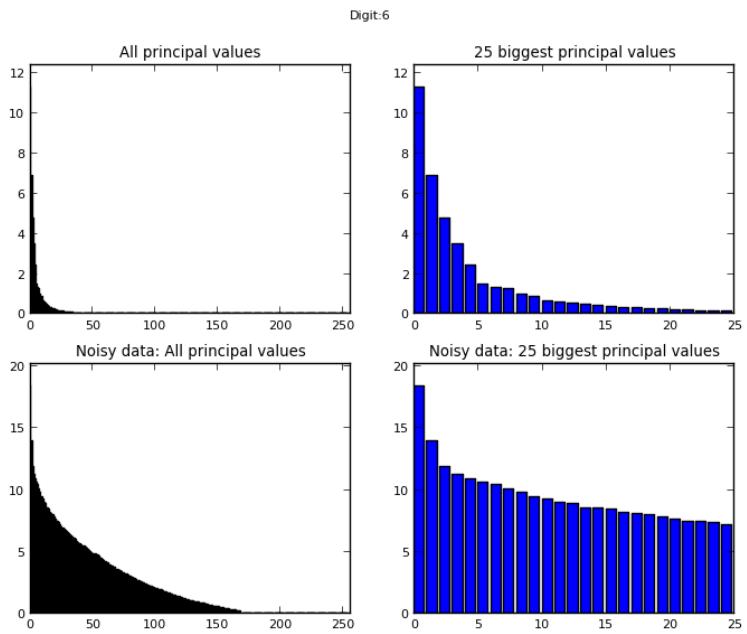


(b)

Figure 6: (a) Mean and the first 5 principal components of the digit 5. (b) Top row: All principal values and the 25 biggest principal values of the digit 5. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 5.

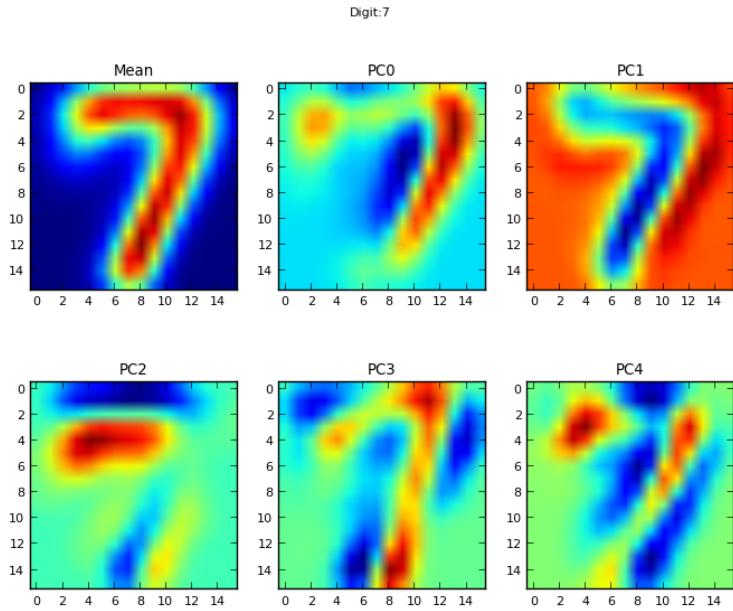


(a)

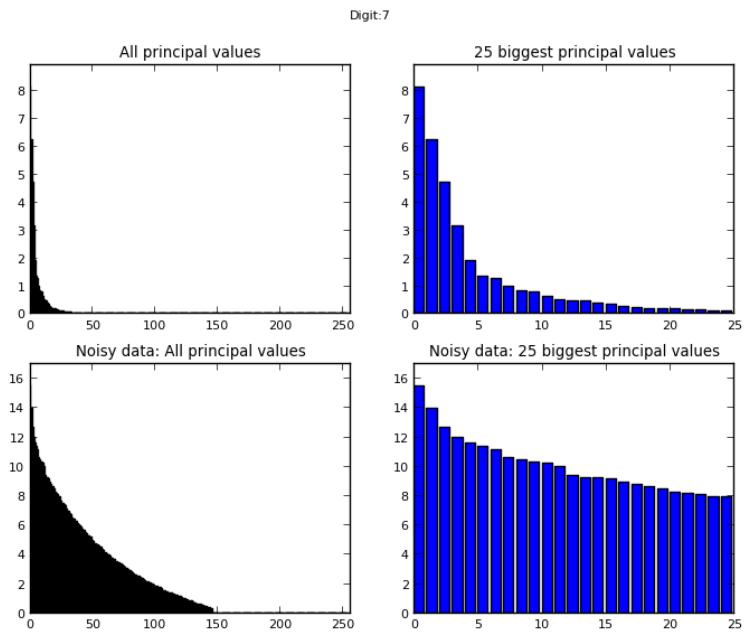


(b)

Figure 7: (a) Mean and the first 5 principal components of the digit 6. (b) Top row: All principal values and the 25 biggest principal values of the digit 6. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 6.

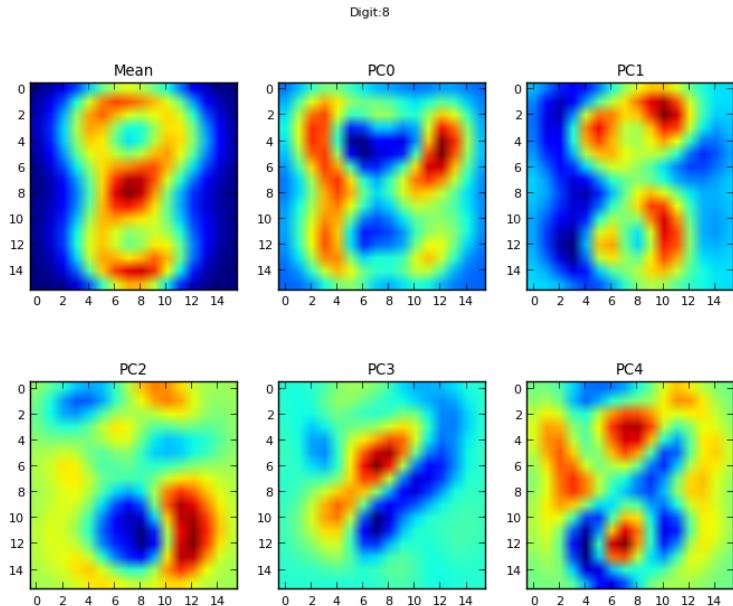


(a)

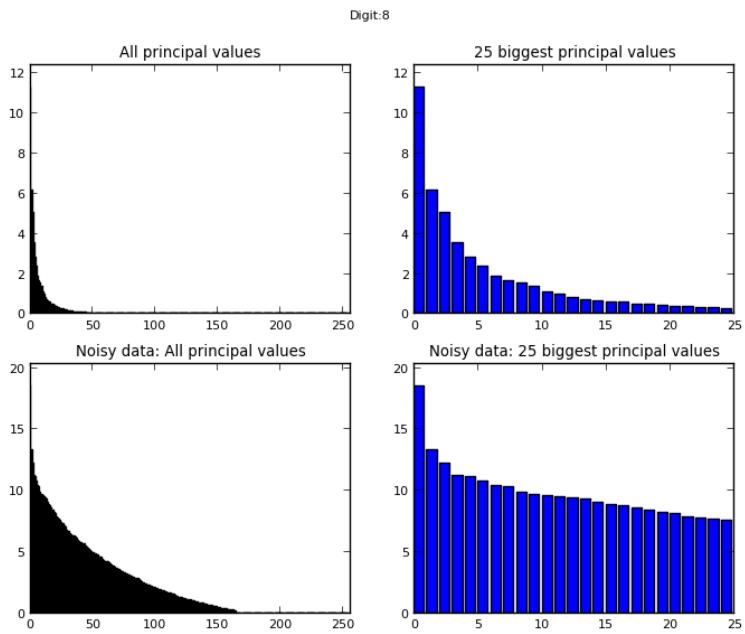


(b)

Figure 8: (a) Mean and the first 5 principal components of the digit 7. (b) Top row: All principal values and the 25 biggest principal values of the digit 7. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 7.

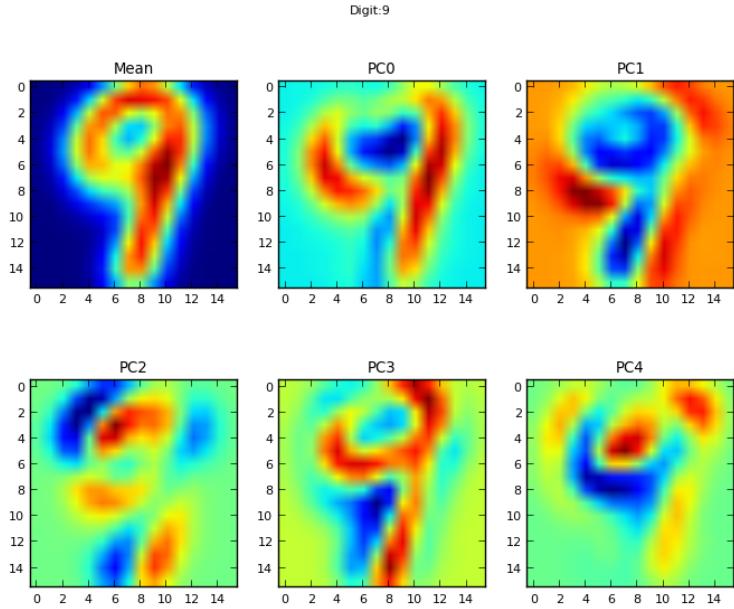


(a)

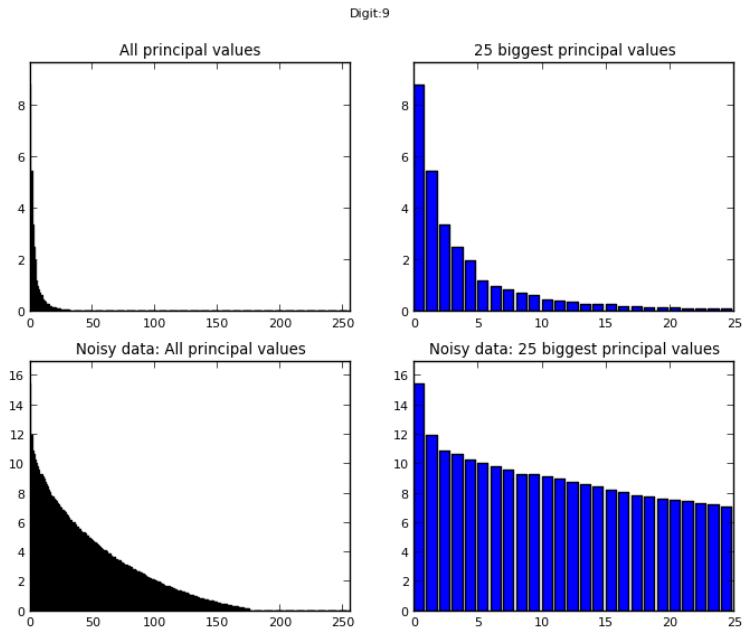


(b)

Figure 9: (a) Mean and the first 5 principal components of the digit 8. (b) Top row: All principal values and the 25 biggest principal values of the digit 8. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 8.



(a)



(b)

Figure 10: (a) Mean and the first 5 principal components of the digit 9. (b) Top row: All principal values and the 25 biggest principal values of the digit 9. Bottom row: All principal values and the 25 biggest principal values of the noisy data of digit 9.

Noisy data

In the following a Gaussian noise with a variance $\sigma^2 = 2.25$ has been added to the original data. For the noisy data we observe that the principle values increase. This is not surprising because what we have done is the following: We added to each dimension X_d a Gaussian white noise W_d with $0 \leq d < 256$ denoting the dimension. All W_d are independent random variables and Gaussian. Thus they maintain this property under rotations as well. Therefore adding independent white noise to the original data is the same as adding independent white

noise to the projected data in the coordinate system spanned by the principle components. Let \tilde{X}_d denote the coordinates in the new coordinate system. Then is the perturbed value $\hat{X}_d = \tilde{X}_d + W_d$. This means for the variance $Var(\hat{X}_d) = Var(\tilde{X}_d) + Var(W_d)$ where $Var(\tilde{X}_d)$ is the d th principal value and $Var(W_d) = \sigma^2$. Adding white noise does not affect the covariance though: $Cov(\hat{X}_i, \hat{X}_j) = Cov(\tilde{X}_i, \tilde{X}_j) = 0$ with $i \neq j$. To make a long story short: Adding Gaussian white noise which is independent, simply increases the principal values and thus the variances in the directions of the principal components by the variance of the white noise σ^2 . However, since we have only a comparatively small number of sampling points of our data in a highly dimensional space, the empirical covariance matrix of the white noise shows still some correlations between pairs of components and the variance of the individual components might differ as well. That is the reason, why we still can observe that the principal values seem to converge against 0 and not against variance of the white noise. Furthermore this explains also why some of the components are influenced by a greater summand than others.

Denoising

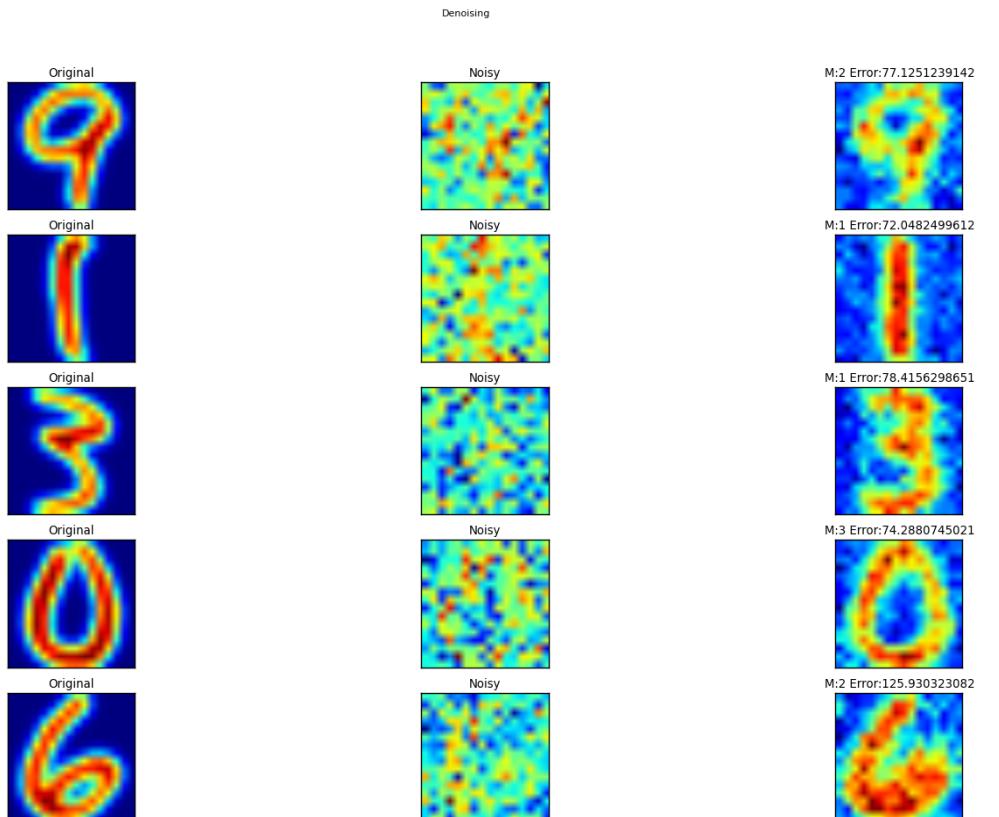


Figure 11: Left column shows the original images. Middle column shows the noisy images. Right column shows the denoised images with M denoting the number of principle components used for the reconstruction.

Assignment 5

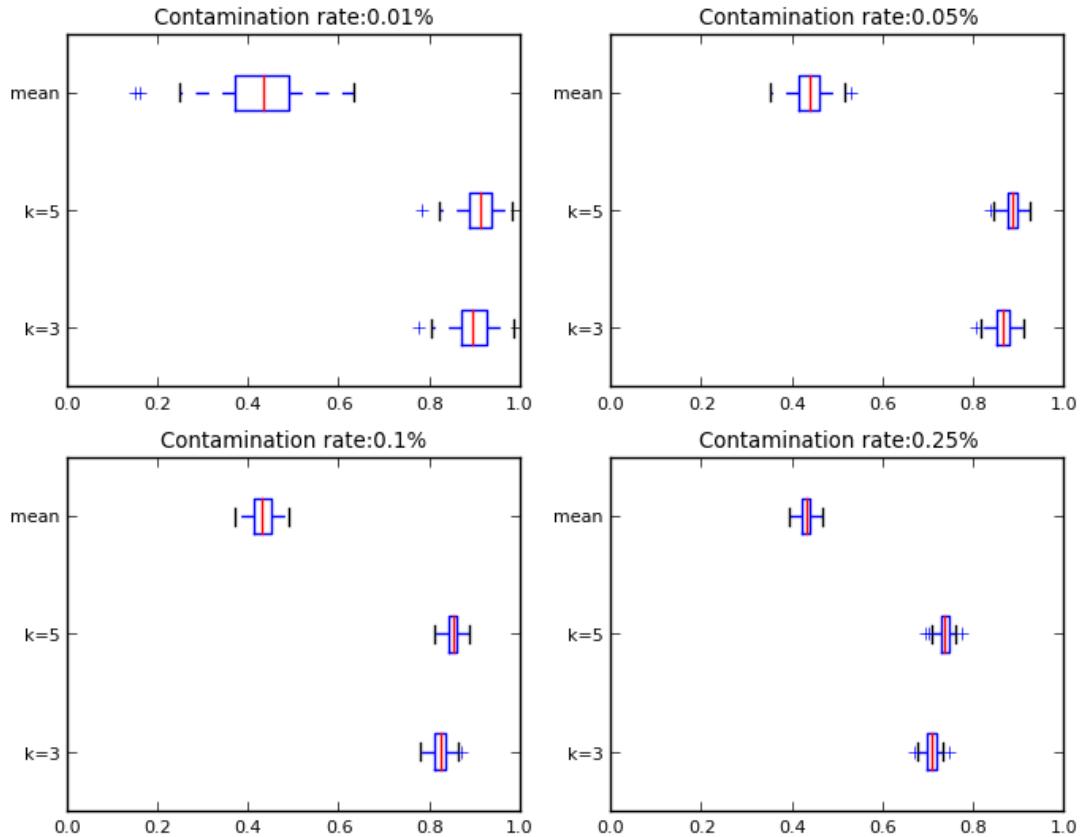


Figure 12: Box plot showing the distribution of the AUCs values for different rates of contamination of the different outlier detection methods: Gamma index with $k = 3$, Gamma index with $k = 5$ and mean distance.

One clearly sees that the mean distance method has an AUC value whose mean is close to 0.5 which means that this method is not well suited for the outlier detection in our case. In contrast to that the gamma index methods perform far more better. For moderate contamination rates, 0.01 and 0.05, they have an AUC value whose mean is close to 0.9. For higher contamination rates, their performance degrades slightly. It is noteworthy that the gamma index method using 5 neighbors performs marginally better than the same method using only 3 neighbors.

Assignment 6

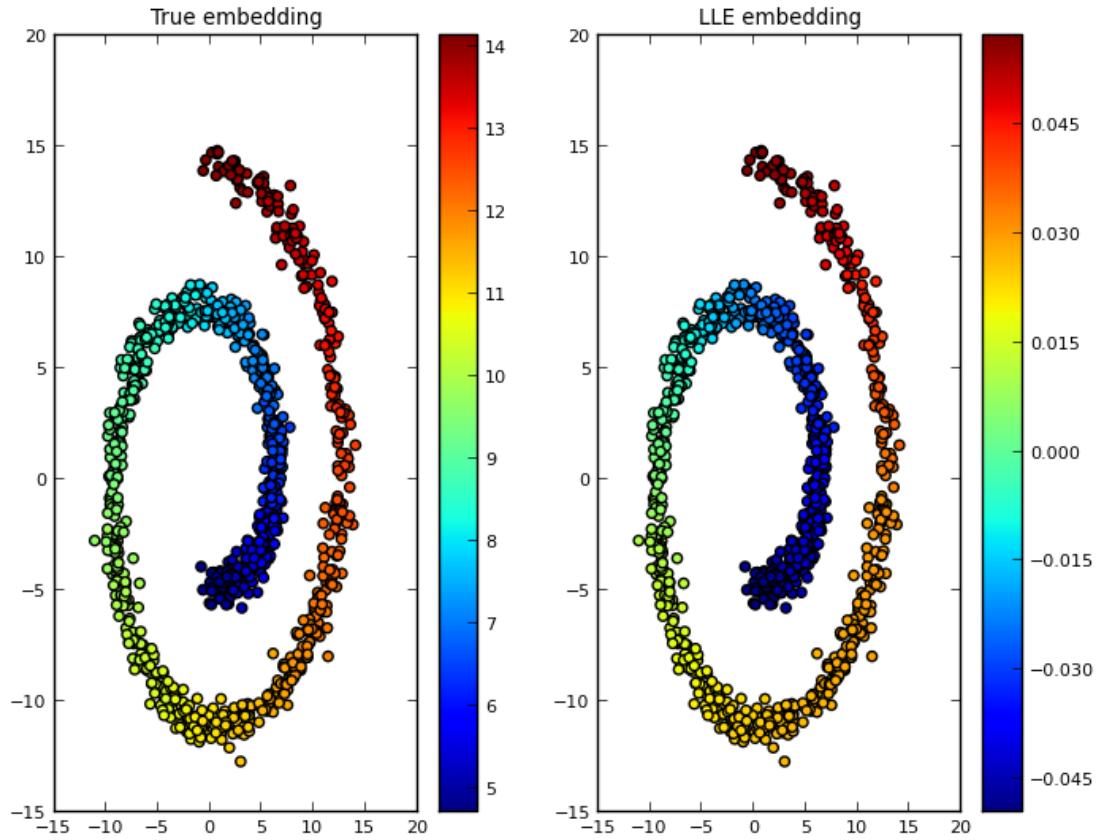


Figure 13: Left: Original data with true embedding indicated by the coloring. Right: Original data with the calculated LLE embedding indicated by the coloring.

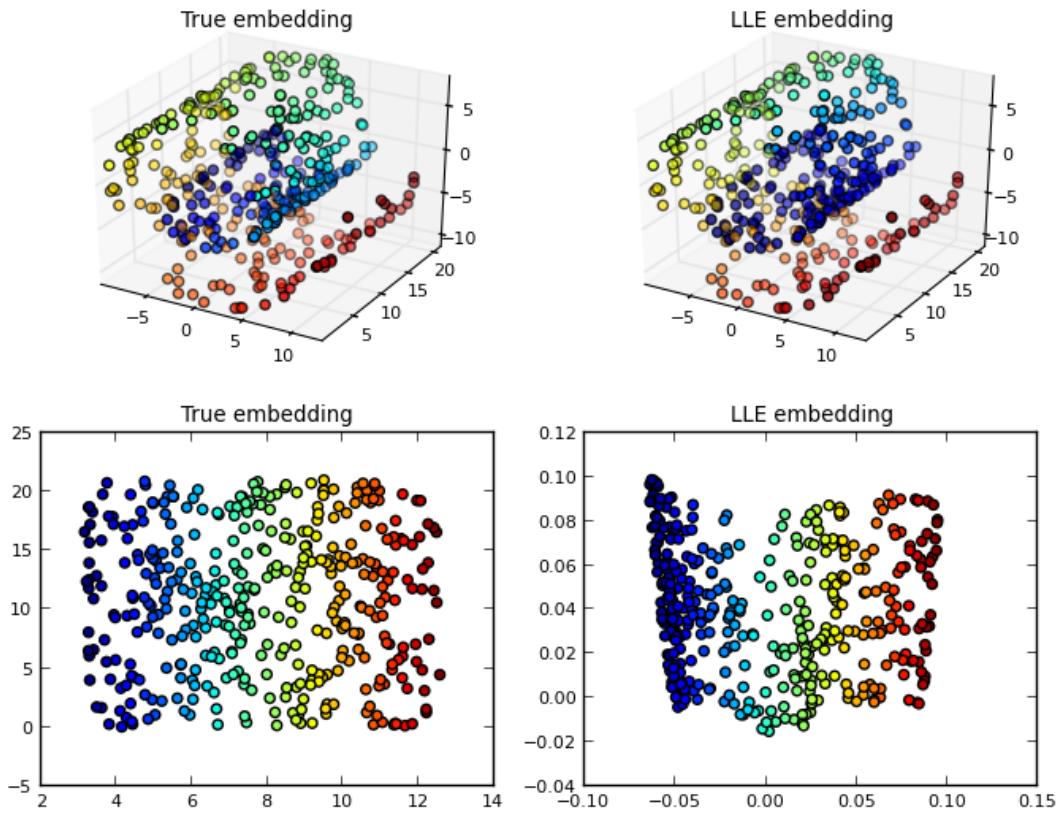


Figure 14: Top left: Original data with true embedding. Bottom left: True 2D manifold with the same coloring as in the top left plot. Top right: Original data with calculated LLE embedding. Bottom right: Calculated 2D manifold with the same coloring as in the top right plot.

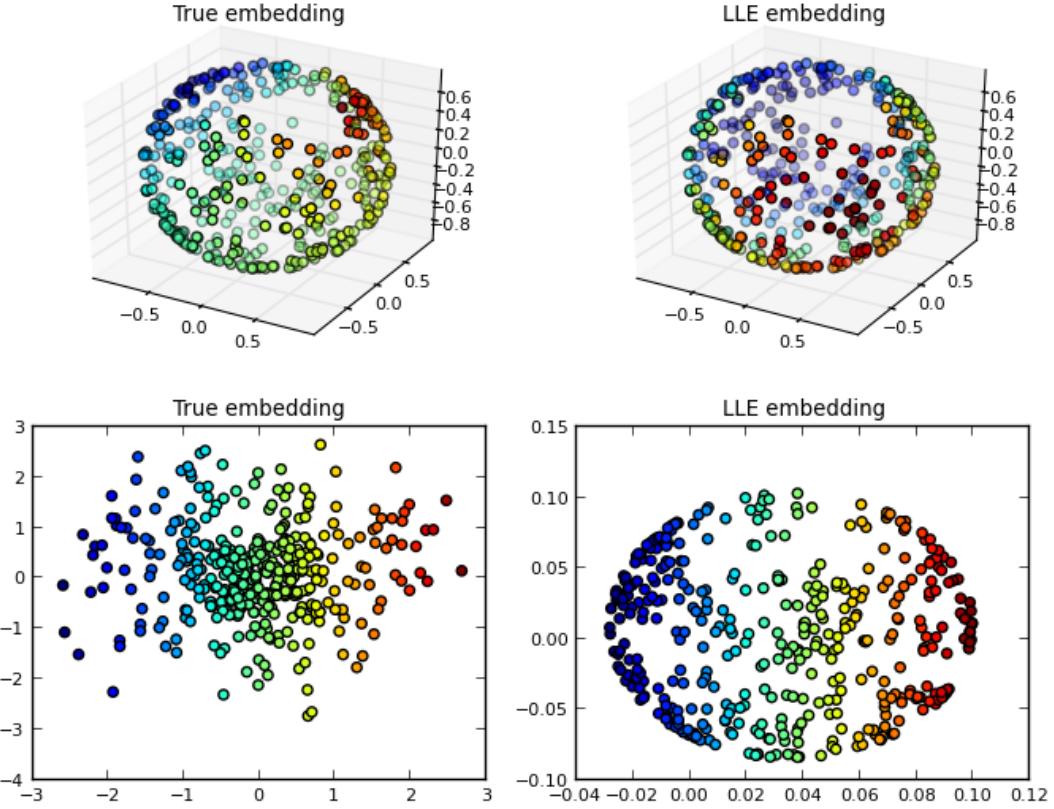


Figure 15: Top left: Original data with true embedding indicated by the coloring. Bottom left: True 2D manifold with the same coloring as in the top left plot. Top right: Original data with calculated LLE embedding. Bottom right: Calculated 2D manifold with the same coloring as in the top right plot.

Why is it difficult to obtain an embedding in which the brink of the fishbowl is on the fringe of the embedding?

In order to embed the fishbowl into a 2-dimensional map, one has to widen the brink of the fishbowl, that is to say one has to stretch it to the outside. This causes inevitably a change in the neighborhood relations of those points (see the 2-dimensional plot of the true embedding where the brink points are far less densely packed than in the LLE embedding). Since LLE maintains the local neighborhood relations, this method has some problems finding a good embedding.

Why does the $\text{eps} - \text{ball}$ criterion appear to work better than the knn -criterion?

With the $\text{eps} - \text{ball}$ criterion the brink points have fewer assigned neighbors than the points not lying close to the brink. Consequently, they also have fewer constraints to fulfill in the final 2-dimensional embedding and thus they can be more easily placed on the fringe of the resulting embedding.

Nonetheless, I obtained the LLE-embedding with the knn criterion and the result is also quite satisfying.

Assignment 6

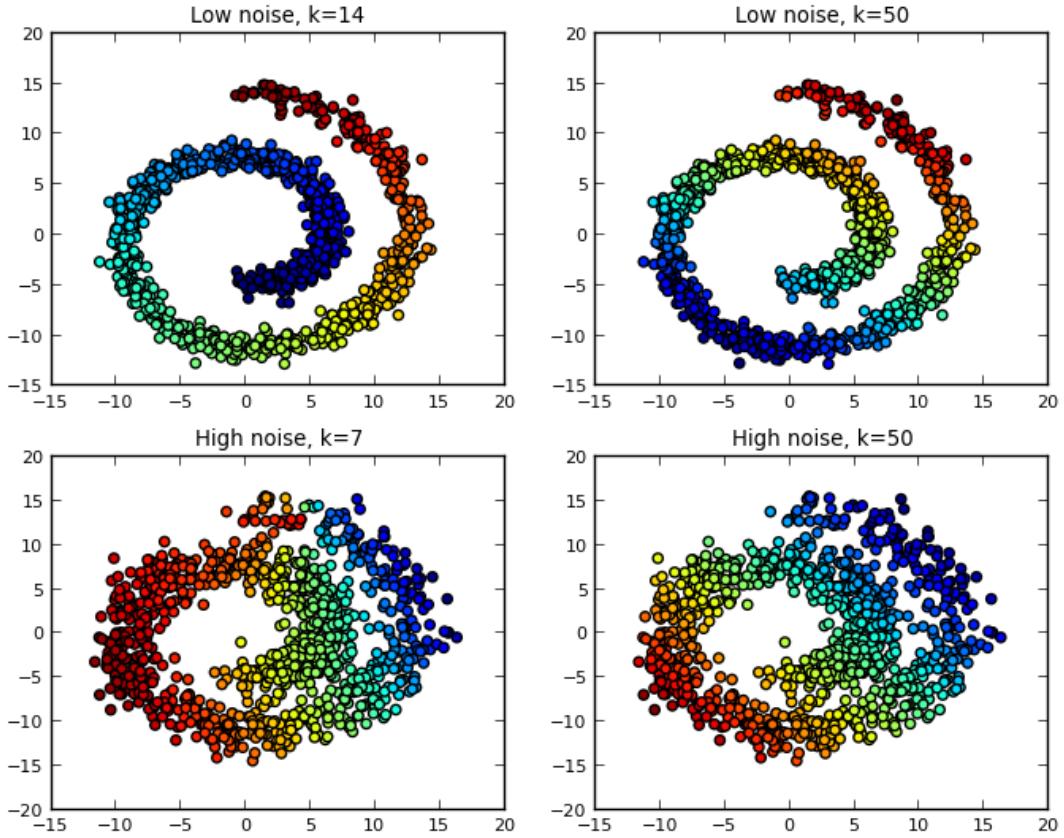


Figure 16: Top left: LLE embedding with $k = 14$ of flat roll with low noise. Top right: LLE embedding with $k = 50$ of flat roll with low noise. Bottom left: LLE embedding with $k = 7$ of flat roll with high noise. Bottom right: LLE embedding with $k = 50$ of flat roll with high noise.

The figure shows that LLE can successfully retrieve the embedding of a flat roll which is affected by low noise. However if one sets the number of neighbors too high, the embedding won't be found. The same holds if the noise becomes too high. In the bottom left image the chosen number of neighbors is the minimum such that the graph is still weakly connected. But since the noise is so high that the roll starts to admix, the neighborhood structure cannot be retrieved correctly anymore. Thus, rendering the lle method unusable in this case.