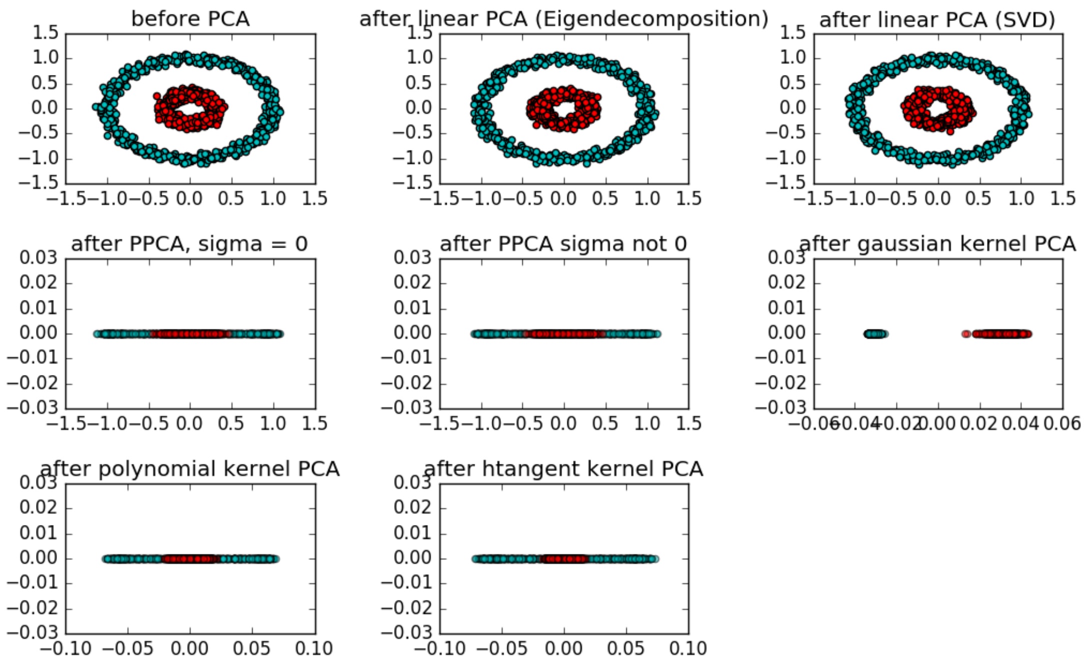
PCA project

Theoretical

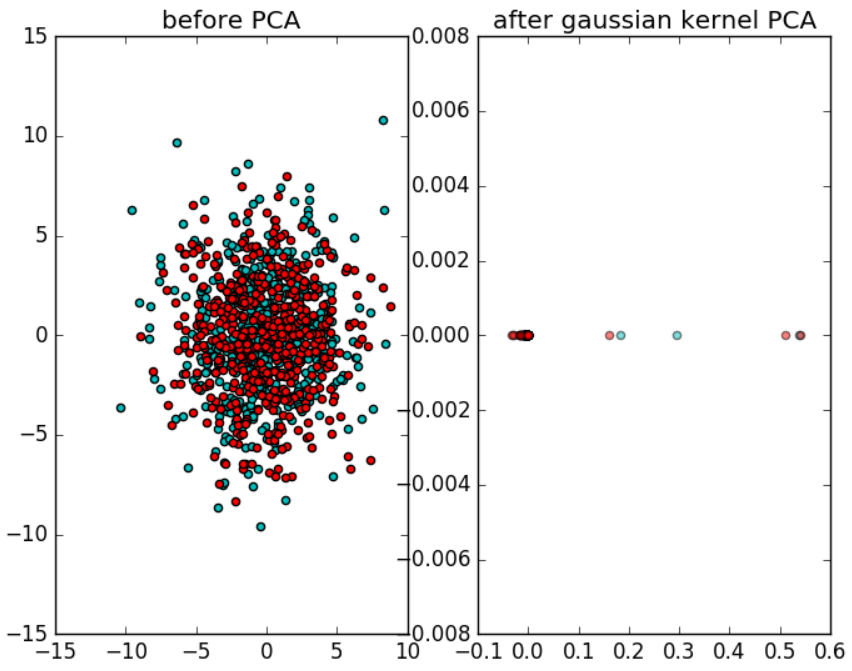
In the PCApackage there are three different versions of PCA with variations. These are PCA (One that calculates the covariance matrix and another that performs SVD on the original data), Probabilistic PCA (one where sigma is equal to zero and one where the sigma is not zero but calculated through maximum likelihood with the EM algorithm) and lastly kernel PCA (with 3 different kernel functions, gaussian, polynomial and hyperbolic tangent).



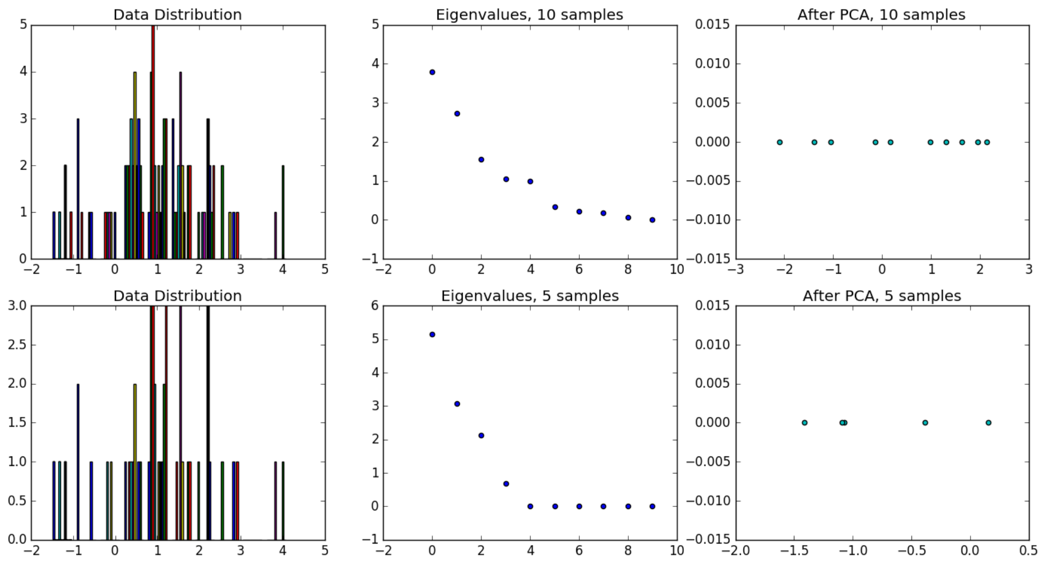
Clearly the circles are not linearly separable so the linear classifiers fail to separate the data. Projecting the data onto a higher dimensional space where the classes become linearly separable solves this problem, hence kernel PCA is the only one to succeed in separating the two classes and the function that seems to do the job is Gaussian.

For Linear PCA it’s not even attempted to project the data on a 1 dimensional space because the eigenvectors of the covariance matrix for the two classes have the same power. That makes sense considering these eigenvectors are basically the lines of maximum projected variance. The eigenvectors are orthogonal to each other (because the covariance matrix is symmetric) and they pass through the center of the circle, so they have almost equal projected variance.

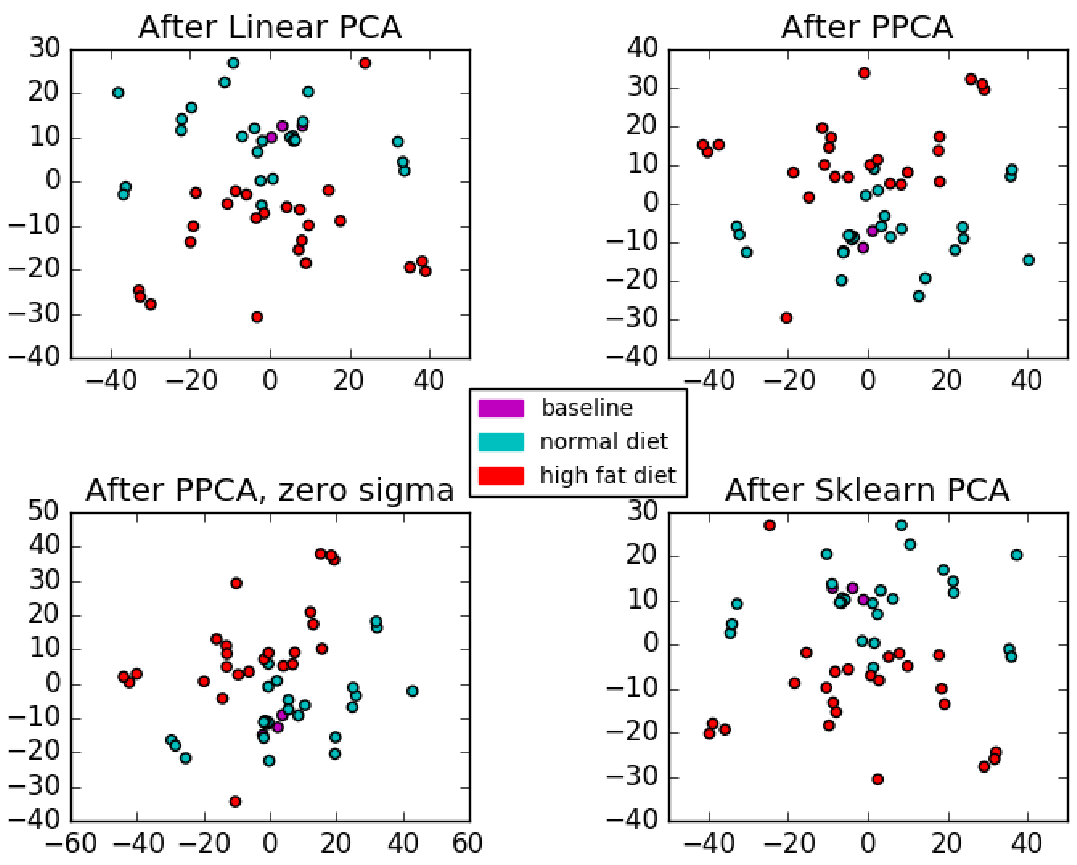
Increasing the noise of the data to 3 basically makes the data one big cluster as is obvious in the plot below. Gaussian kernel can't separate the data, because it's not separable.



Lastly an artificial 10-dimensional dataset with mean of ones was made. PCA was then performed on it twice, once with 10 samples and the second time with 5. Since there are less samples than dimensions on the second run, computation of the covariance matrix is considered better than SVD of the original data. The resulting eigenvalues were plotted.



Lastly, PCA was performed on gene expression data from livers of C57BL/6J mice who were on different diets. The results are plotted below in 2 dimensions



The picture we see is of similar pattern on all plots. The built-in sci-kit learn PCA function gives this pattern as well which proves the functions work well. The apparent detail is that in my PPCA the picture is reversed, which is something I can’t explain.