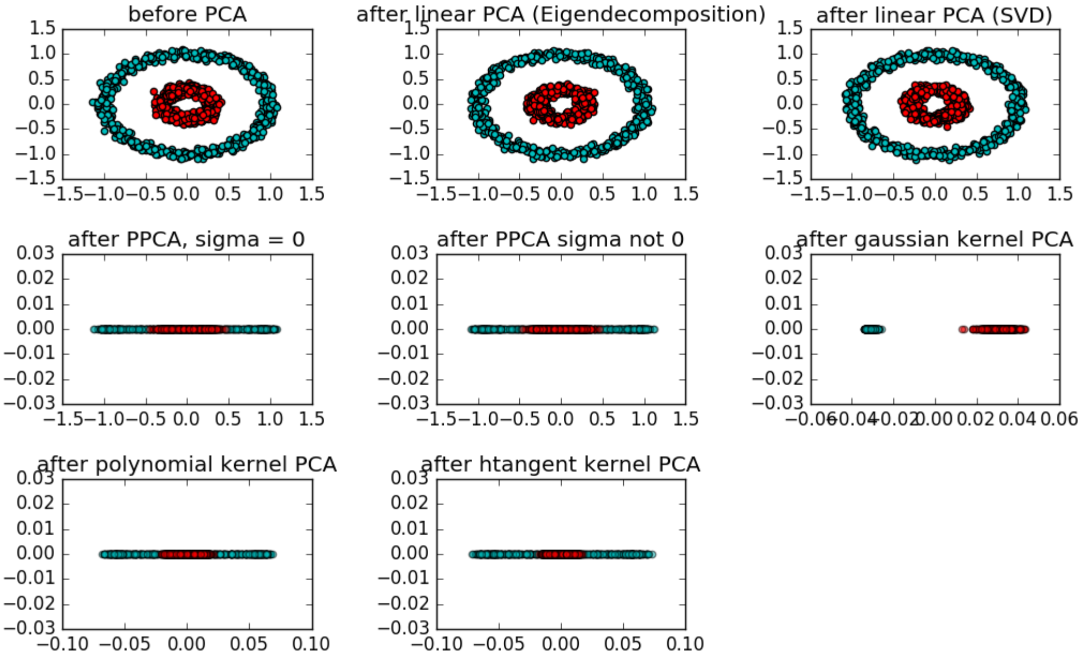
PCA project

Theoretical

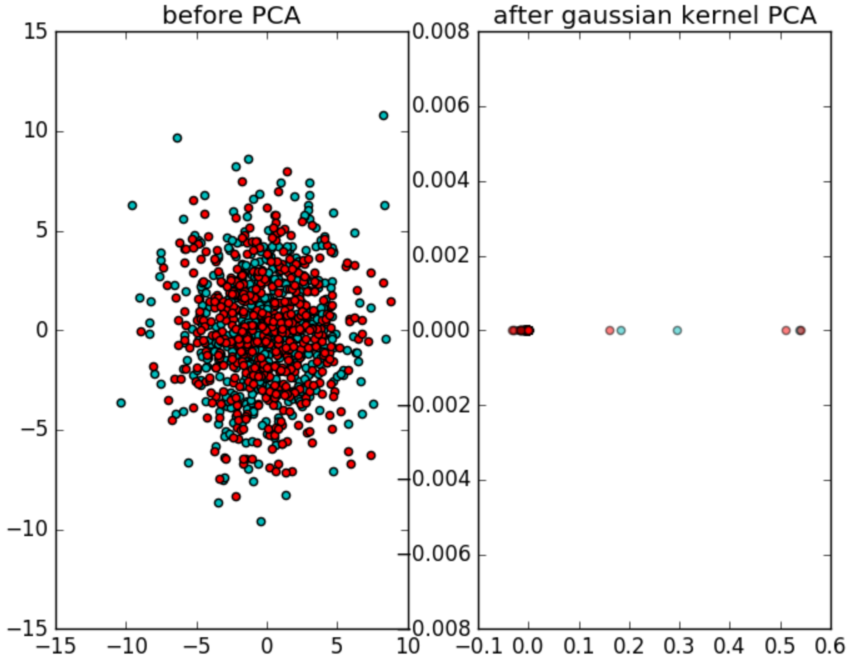
In the PCApackage there are three different versions of PCA with variations. These are PCA (One that calculates the covariance matrix and another that performs SVD on the original data), Probabilistic PCA (one where sigma is equal to zero and one where the sigma is not zero but calculated through maximum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



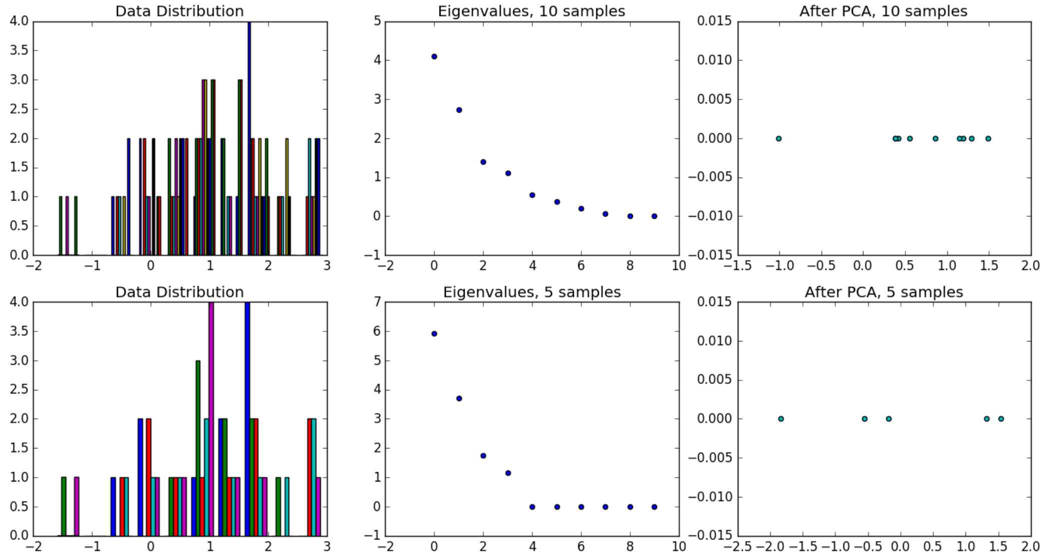
Clearly the circles are not linearly separable so the linear classifiers fail to separate the data. Projecting the data onto a higher dimensional space where the classes become linearly separable solves this problem, hence kernel PCA is the only one to succeed in separating the two classes and the function that seems to do the job is Gaussian. It makes sense that the Gaussian distribution is the one to do the job considering that in 2-D space it is represented by contours, perfect separators for circles. For the 3-D space we can imagine that the small circle will lie below the bell shaped curve while the big circle will surround it.

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.

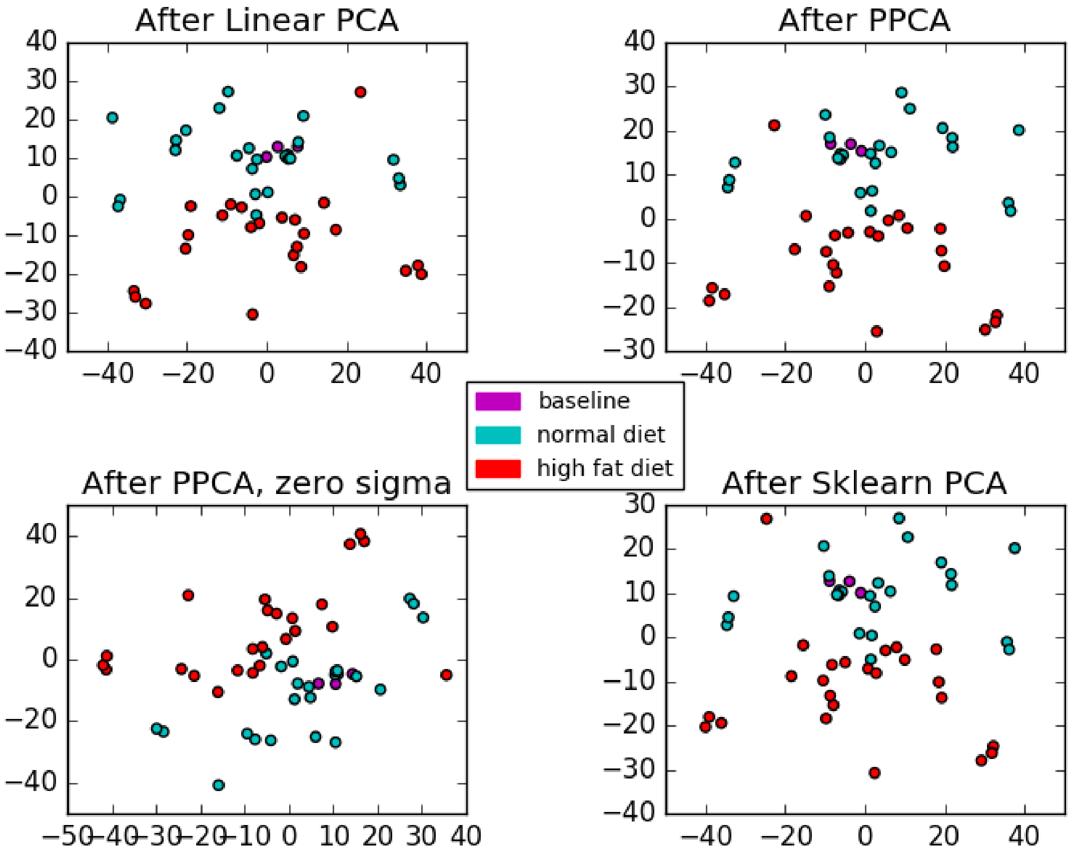


For the last part of the theoretical exercise, an artificial 10-dimensional dataset with mean of ones was made. PCA was then performed on it twice, once with 10 samples and the second time with 5. Since there are less samples than dimensions on the second run, computation of the covariance matrix is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



By subtracting 5 samples the square matrix becomes non-square. Eigenvalues refer by definition to square matrices, but for non square matrices we calculate the singular values which are the positive square roots of the eigenvalues. It holds true that the distinct singular values cannot be more than the minimum dimension (in this case we reduced the minimum dimension to 5), so all the eigenvalues after 5 are zero. The 5th eigenvalue seems to be 0 as well however that is probably because the dataset is so simple it will be perfectly described by 1-2 principal components.

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. (zero sigma is not a good estimation so let’s not bother with that)

We originally had the expression of 45.281 genes and after running PCA this is reduced to 2 principal components that are completely uncorrelated, given the apparent randomness in the scatter plots. The principal components summarize the gene expression tendency. Two clusters seem to be perfectly distinguishable in the diagrams, one for the normal-diet and baseline samples and one for the high fat diet samples. Both seem to have the same distribution over the first principal component (X axis) which probably describes the majority of genes but greatly differ on the the second (Y axis). High fat diet seems to greatly affect the expression of the genes described by the second PC, as it greatly deviates from the normal diet and baseline tendency.