IDS2018 - Assignment 3

Isabela Blucher

March 10, 2018

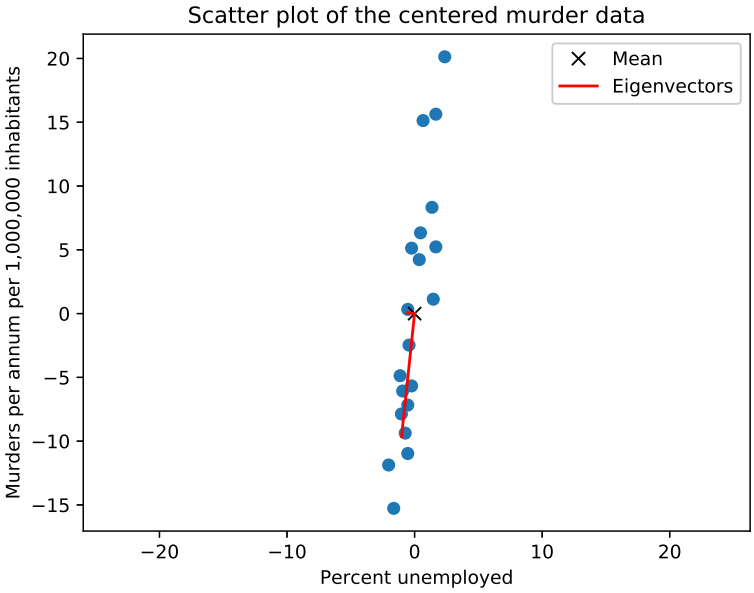
Exercise 1

(a) PCA implementation

The code for this exercise was implemented based on Jonathon Shlens' tutorial on Principal Component Analysis available on Absalon. It's important to point out that all the analysis done for this assignment was with centered, but not fully normalized data.

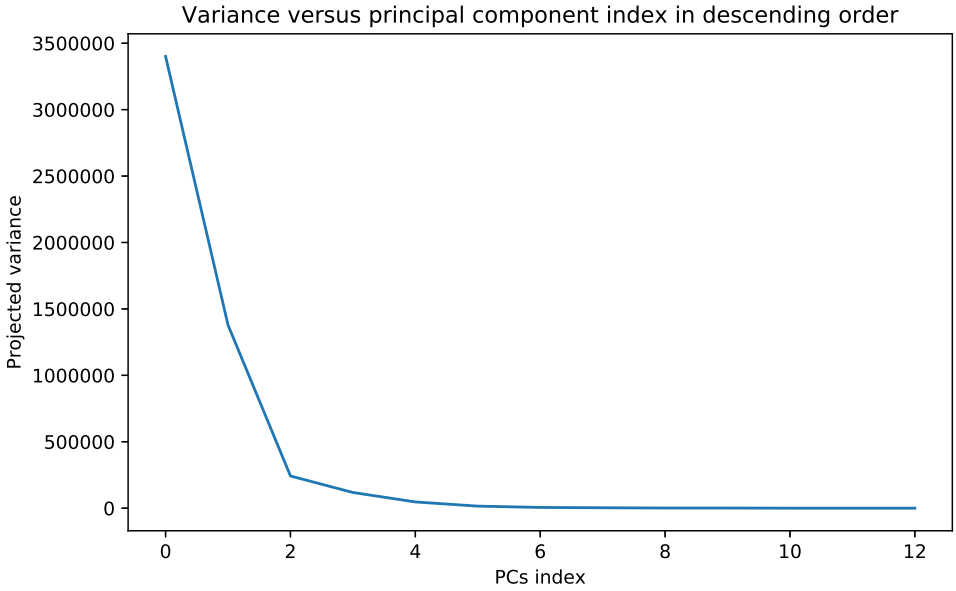
(b) PCA performance on the murder dataset

The following plot represents the centered murder dataset, which means that the dataset mean is the exact point (0, 0). In red are the principal eigenvectors pointing out of the mean. The eigenvectors have been scaled by the standard deviation of the data projected onto the respective eigenvector.

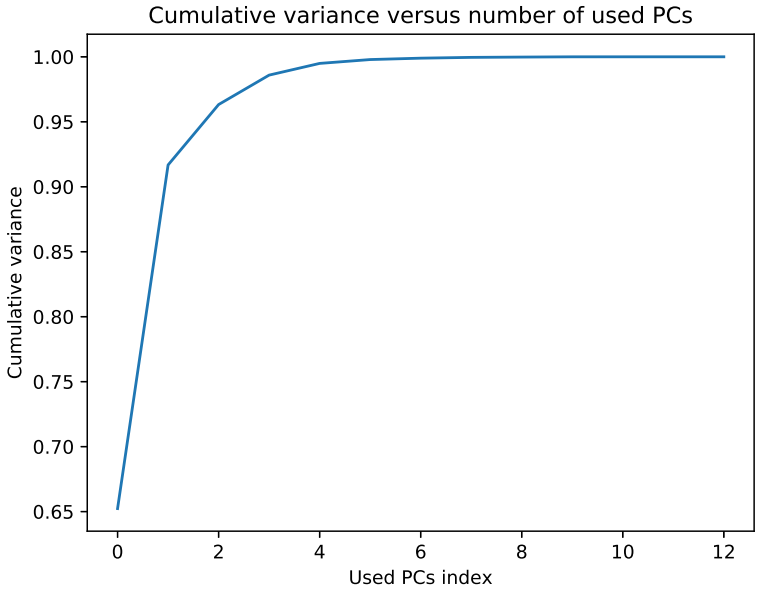


(c) PCA performance on the pesticide dataset

In the plot below we can see that because we're analyzing the principal components in descending order, the projected variances are strictly decreasing. As expected, we also see the variance value stabilizing.



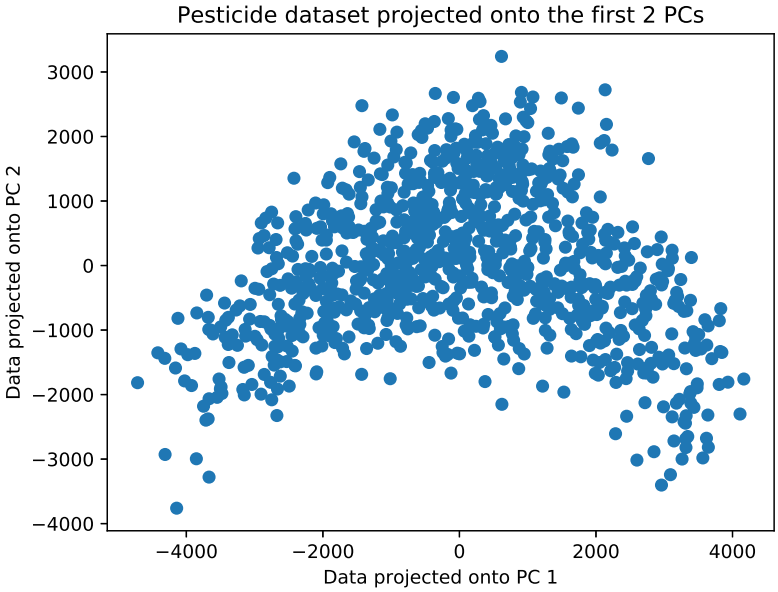
For the cumulative variance, we get the following plot for all 13 eigenvalues. By visual inspection, we see that the sum of all variances is 1, and the captured variance increases quickly for a small number of dimensions.



Looking at the array of cumulative variance values: [0.65232418459863228, 0.91676936472893444, 0.9632517112618828, 0.98591075070839918, 0.994919080719875, 0.99788133176279903, 0.9989474349760783, 0.99955753852042817, 0.99979512857059338, 0.99998843836866336, 0.99999495970389607, 0.99999999978233467, 0.99999999999999989], we can see that to capture 90% of the variance in the pesticide dataset, two dimensions are used, and for 95% captured variance, three dimensions are used.

Exercise 2 (Visualization in 2D)

The plot below was generated with the `mds.py` function, which returns the dot product of the data matrix and the first d principal components. Since only two PCs were used, the x axis represents the first eigenvector and the y axis the second.



Exercise 3 (Clustering)

For this exercise, the clustering was done with the `scikit-learn` function. The 2-means clustering was initialized with the first two data points in `IDSWeedCropTrain.csv`. Since our dataset has 13 features, we're dealing with a 13-dimensional problem, and thus, our cluster centers have 13 dimensions. The following centers were found when running the algorithm:

Cluster center 1 = [1.85326752e+00, 1.88580425e+01, 3.28324480e+02, 1.29782138e+03, 1.04398350e+02, -6.69043650e+02, -3.75582393e+02, -2.65137658e+02, -2.24878762e+02, -1.35295688e+02, -6.68472633e+01, -1.32053312e+01, -1.26738641e+00]

Cluster center 2 = [-1.65007372e+00, -1.67904310e+01, -2.92326711e+02, -1.15552716e+03, -9.29520284e+01, 5.95689147e+02, 3.34403227e+02, 2.36067745e+02, 2.00222868e+02, 1.20461756e+02, 5.95180737e+01, 1.17574877e+01, 1.12842911e+00]

What the software used does is try and find structure in our data. For our given $k = 2$, in this case, we want to separate data into 2 clusters where the points in the same cluster are similar to each other. We give the algorithm two staring centroids and it will run, until it finds new clusters and centers that are optimal and minimize our objective function (or distortion measure).