University of Copenhagen

Faculty of Science

Introduction to Data Science

Assignment 3

Dimitrios Galinos (bst265)

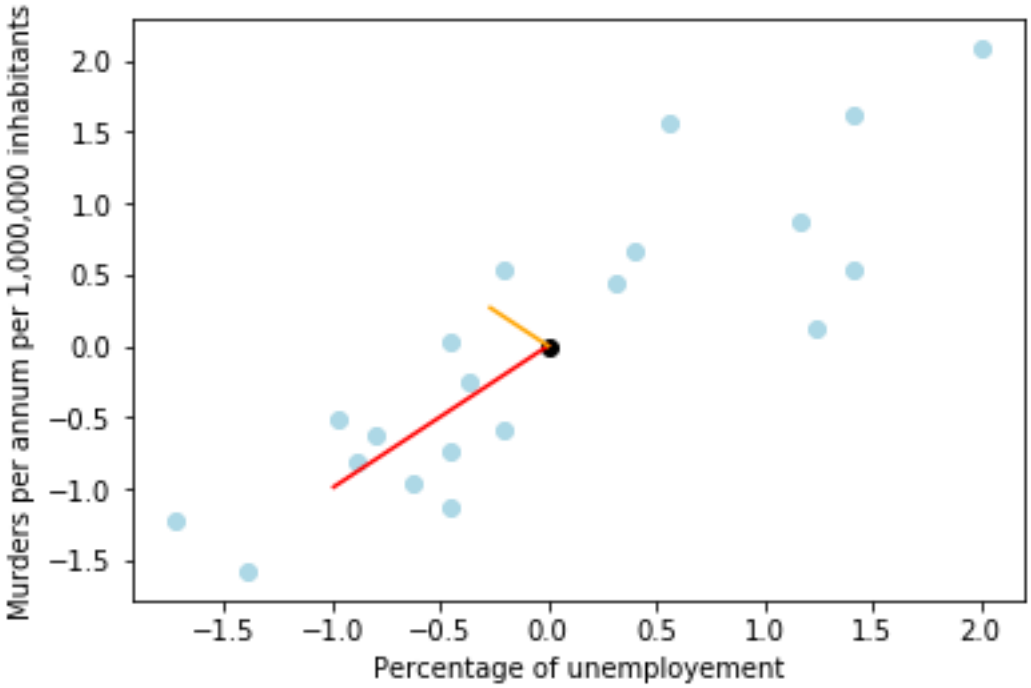
March 3, 2020

1 Performing PCA

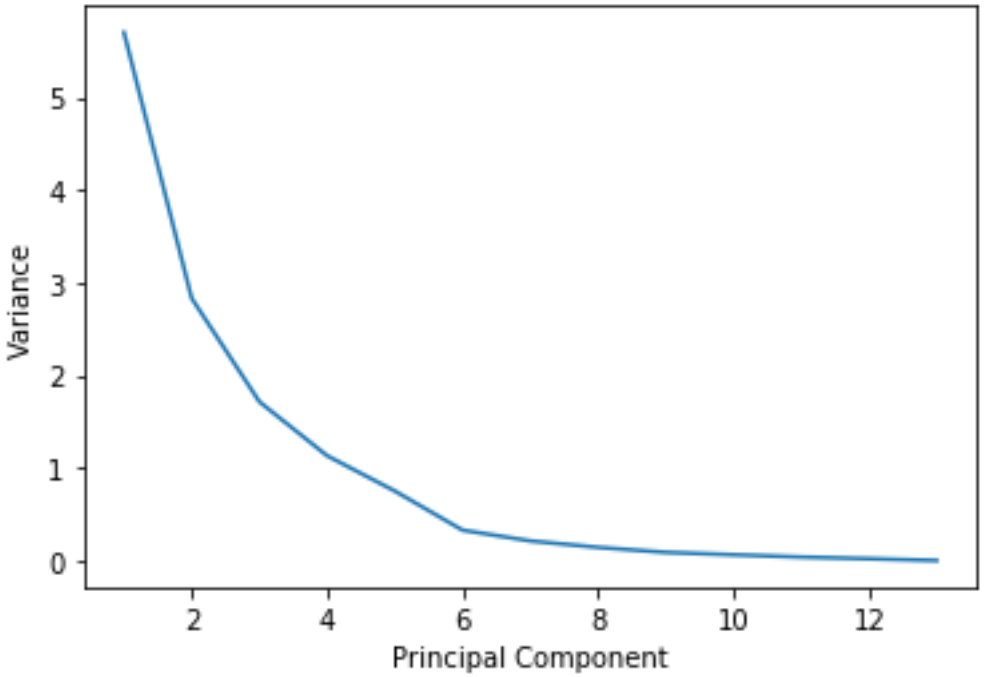
The code is included inside the dimitrios.galinos.zip and is named "assignment3.py" for the main part of the execution while also having the "mds.py" for the multidimensional scaling and the "pca.py" for my own PCA implementation. Furthermore whenever I mention the pesticide dataset please note that I have used only the training part and not the XTrain test part (I made this assumption since no extra details where provided and in a TA session I was told it is fine). Finally I want to mention that all the data I use have been standardized (zero mean and unit variance), the reason for this is that I have read in a lot of PCA papers that PCA is affected by scale so the data should always be standardized before passing them through the PCA in order for it to work efficiently. Please also read the README file.

a). I have implemented my own PCA using the supplied template in the file "pca.py". A quick description of what I am doing there is centralizing the data, finding the covariance matrix, calculating the eigenvalues and eigenvectors sorting those eigenvalues and eigenvectors in a descending eigenvalue order and finally return them along with the index order of the sorting (which I thought might be a useful value to return as well).

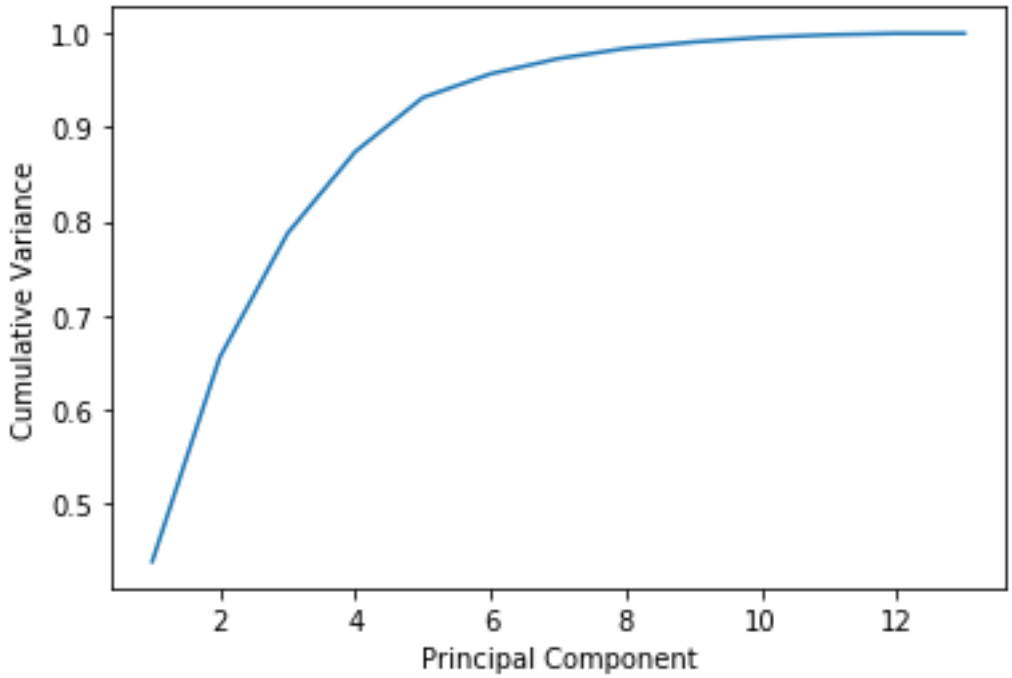
b) I performed my own implementation of PCA in the murder dataset. The plot of the murder dataset along with the mean and the principal eigenvectors pointing out of the mean, each eigenvector with a length scaled by the standard deviation of the data projected onto that eigenvector can be found bellow.



c) I performed my own implementation of PCA in the pesticide dataset. The plot of the pesticide dataset's variance against principal components is:



The plot of the cumulative variance which was normalized along all PCs such that the sum of all variances is 1 versus the PCs.



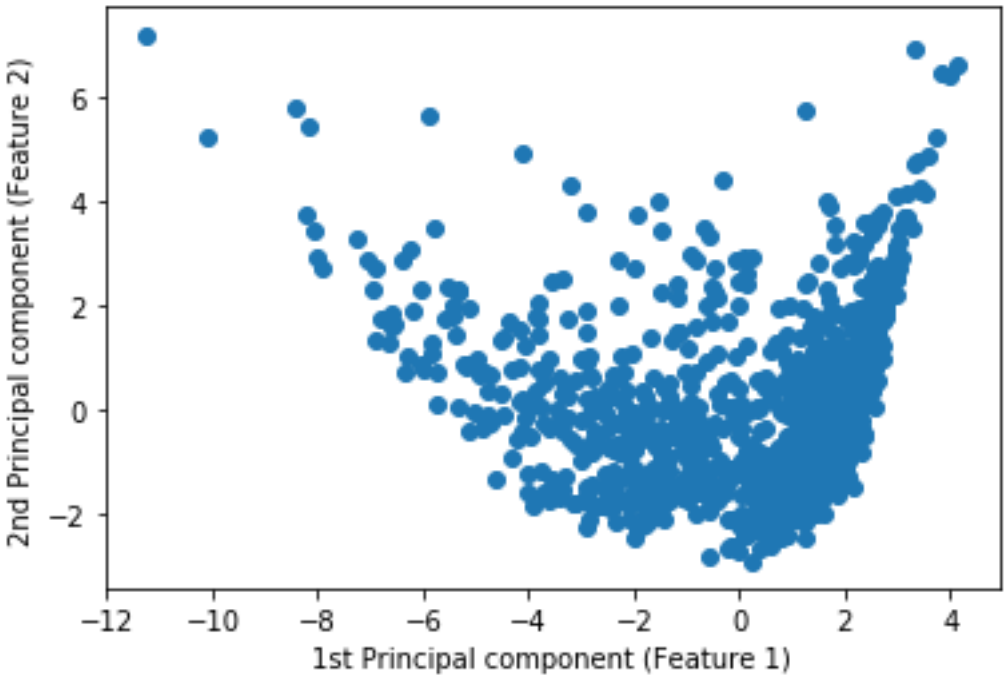
Using the cumulative variance we can easily calculate how many PCs we need to explain a certain percentage of the variance. In our case we need:

5 PCs in order to capture 90% of the variance

6 PCs in order to capture 95% of the variance

2 Visualization in 2D

I implemented my multidimensional scaling using the supplied template in the file "mds.py". A quick description of what I am doing there is perform PCA (using my own "pca.py" but it works exactly the same using sklearn's PCA) on the dataset given in order to find the eigenvectors in descending order, I centralize the dataset and find the new d-dimension dataset which is the projection of the original dataset onto the first d principal components. The plot of the 2-D projection of the pesticide dataset is below:



Note: I have noticed that in my PCA implementation vs sklearn's PCA sometimes the signs are opposite, after some research and a long talk with a TA I came to the conclusion that it is not a problem if the signs are inversed because the variance relations remain the same.

3 Clustering

I performed a 2-means clustering exactly as requested. I used sklearn.cluster.KMeans, I initialized the two starting points to be the first two data points of XTrain they way it is shown in the exercise text, run the KMeans algorithm with the following parameters: n_clusters=2 (2 cluster centers), n_init=1 (running only 1 trial), init=start (my starting point of the first two data points of XTrain) and algorithm=full in order to run the classical EM-Style algorithm. Finally the results where the two final cluster centers are:

```
[[0.10697804  0.15743643   0.24443175  0.42630232   0.36461238  (1)
−0.28733807  −0.48392646  −0.52535925  −0.49566457  −0.43337993  (2)
−0.37703005  −0.26288248  −0.17762998]                            (3)
[−0.26064409  −0.38358222  −0.59553991  −1.03865412  −0.88835112  (4)
 0.70007797   1.17905107   1.27999901   1.20765011   1.05589819  (5)
 0.91860586   0.64049373   0.43278231]]                            (6)
```

4 Bayesian Statistics

Q: How is probability interpreted differently in the frequentist and Bayesian views?
A: In the Bayesian view the probability is a degree of belief or a measure of certainty. In the frequentist view, a probability is a frequency, the frequency of observing that event in a large number of trials.

Q: Cheap, efficient computers played a major role in making Bayesian methods mainstream. Why?
A: The frequentist approach is plagued by inconsistencies and limitations. Bayesian models are often analytically intractable and thus require methods based on simulation. Cheap and fast computers, and general-purpose software resolved this issue.

Q: What is the difference between a Bayesian credible interval and a frequentist confidence interval?
A: For the Bayesian credible interval the parameter is random but the data is fixed while the opposite holds true for the frequentist confidence interval where data is random but the parameter is fixed.

Q: How does a maximum likelihood estimate approximate full Bayesian inference?
A: It assumes the prior is uniform and a zero-one error, then obtains the maximum likelihood estimate.

Q: When will point estimates be a good approximation of full Bayesian inference?

A: When the data are not sparse and the uniform prior does not induce strong and unsuited prior beliefs.