Homework 2

Daria Laslo

March 29, 2022

Exercise 1. **(a).** In the first part of this exercise, we obtain the principal components following the approach in the previous homework, namely, calculating the eigenvalues and eigen vectors using the covariance matrix. We can see in Figure 1 that two principal components are sufficient for a relatively clear separation of the three classes.

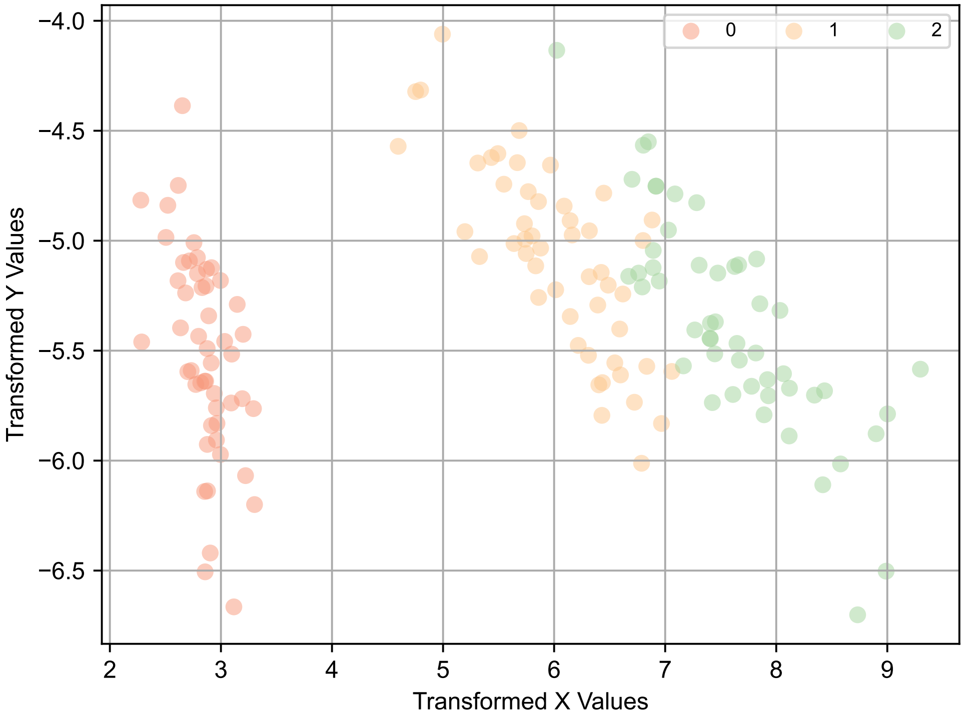

Figure 1: Scatter plot for the first two principal components(obtained using the covariance matrix) on the original Iris dataset.

In Figure 2 we can observe the cumulative variance explained by an increasing number of components. Two components are enough to explain 98% of the variance in the dataset.

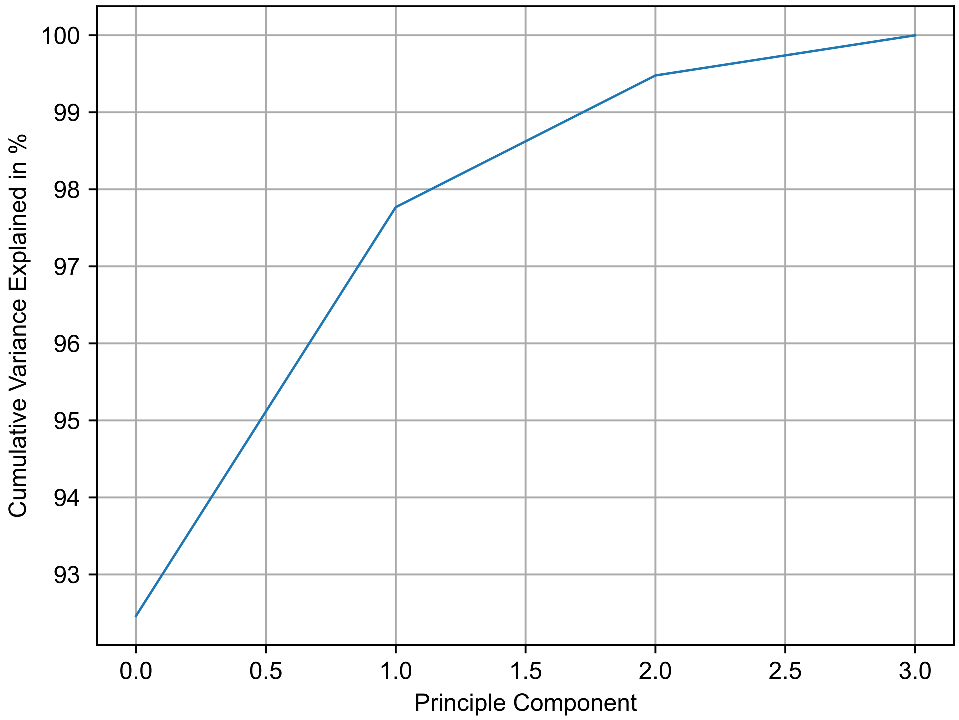

Figure 2: Cumulative plot of the variance explained by all principal components (obtained using the covariance matrix) on the original Iris dataset.

(b). The following shows the PCA pseudocode using SVD:

```
Input: Data matrix $D \in R^{nxd}$ , with n samples and d features
Return: Eigen vectors (and eigen values)
Algorithm:
begin
[L, D, S] ← $SVD(D)$
eigen_values ← $D^2/n$
eigen_vectors ← $L$
sort eigen_values in decreasing order
sort eigen_vectors based on the order of the sorted corresponding eigen_values
return sorted eigen_vectors (and eigen_values if needed for calculating the explained variance)
end
```

(d). The following figures show the results obtained by performing PCA using SVD on the mean centered Iris data. In Figure 3 we can see a very similar representation for the data set using the first two principal components with a relatively clear separation of the three classes. The main difference we can observe compared to the approach showed in (a) is the different scale for the axis which comes as a result of centering the data.

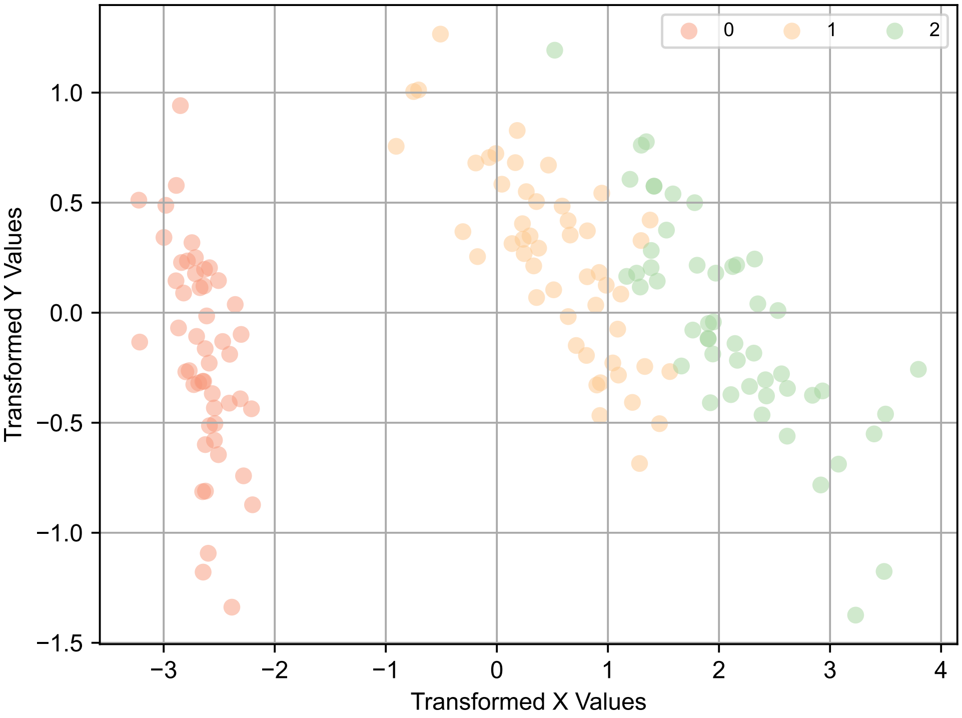

Figure 3: Scatter plot for the first two principal components(obtained using SVD) on the centered Iris data set.

In Figure 4, the cumulative explained variance is shown. These results are matching the ones obtained in (a), having the same percentage of variance explained with each additional principal component.

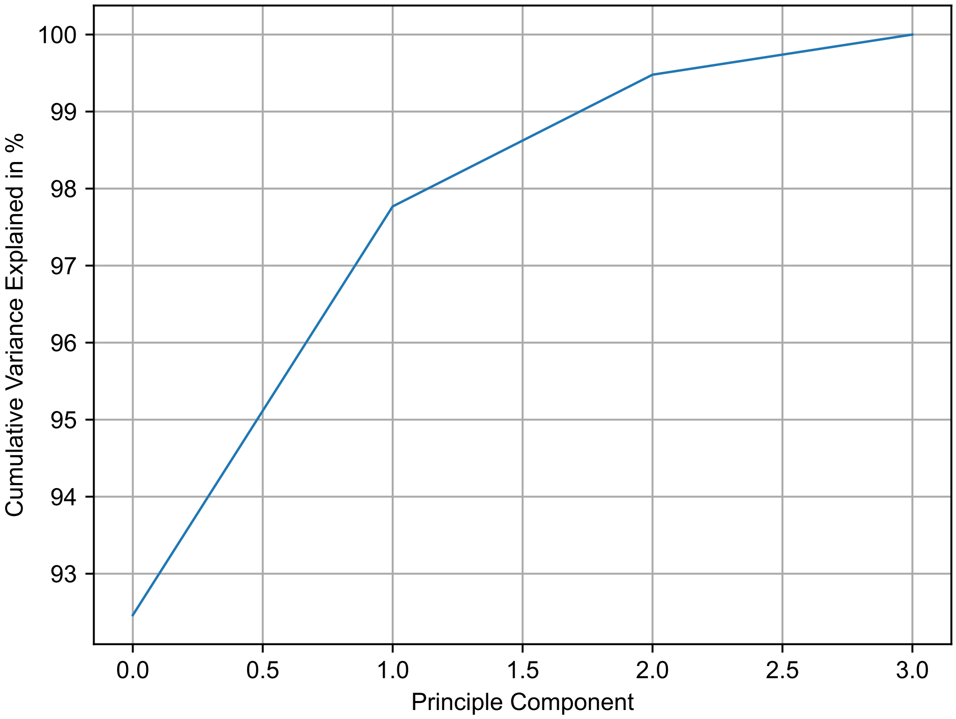

Figure 4: Cumulative plot of the variance explained by all principal components (obtained using the SVD) on the mean centered Iris data set.

(e). In the previous subpoints, we have shown that the results obtained using the eigen-value decomposition (a) are the same as the ones obtained using SVD for the given data set. However, there may be advantages in using one method over the other related to run time and numerical issues.

When performing the eigen-value decomposition, we first need to calculate the covariance matrix X^TX which implies a complexity of $\mathcal{O}(nd^2)$ in a data set with n samples and d features, when n>d (in the other case, we would have a complexity of $\mathcal{O}(dn^2)$ given we would opt for a speed up).However, using SVD can be more efficient by working directly with the data set and obtaining the eigen values and eigen vectors without an intermediate step (that of calculating the covariance matrix (or kernel matrix, when d>n)).

Additionally, numerical issues related to the precision of the machine may arise when working with the covariance matrix. In the case of very low values (threshold given by the precision of the machine), they will be set to 0. If we are interested in very small singular values, then working with the data set directly using SVD may be really advantageous.

Exercise 2. The two criteria are satisfied by the computed pseudo-inverse (see py files).

Exercise 3. **(b).** In Figure 5 we can see the denoised image using a threshold of 1000. The amount of noise seem to be visibly reduced with a the photo looking smoother than before. The black and white dots comprising the noise disappear and at a closer look they transformed into 'cross-like' shapes borrowing the color of the neighbouring pixels which contributes to the smoother look of the image.

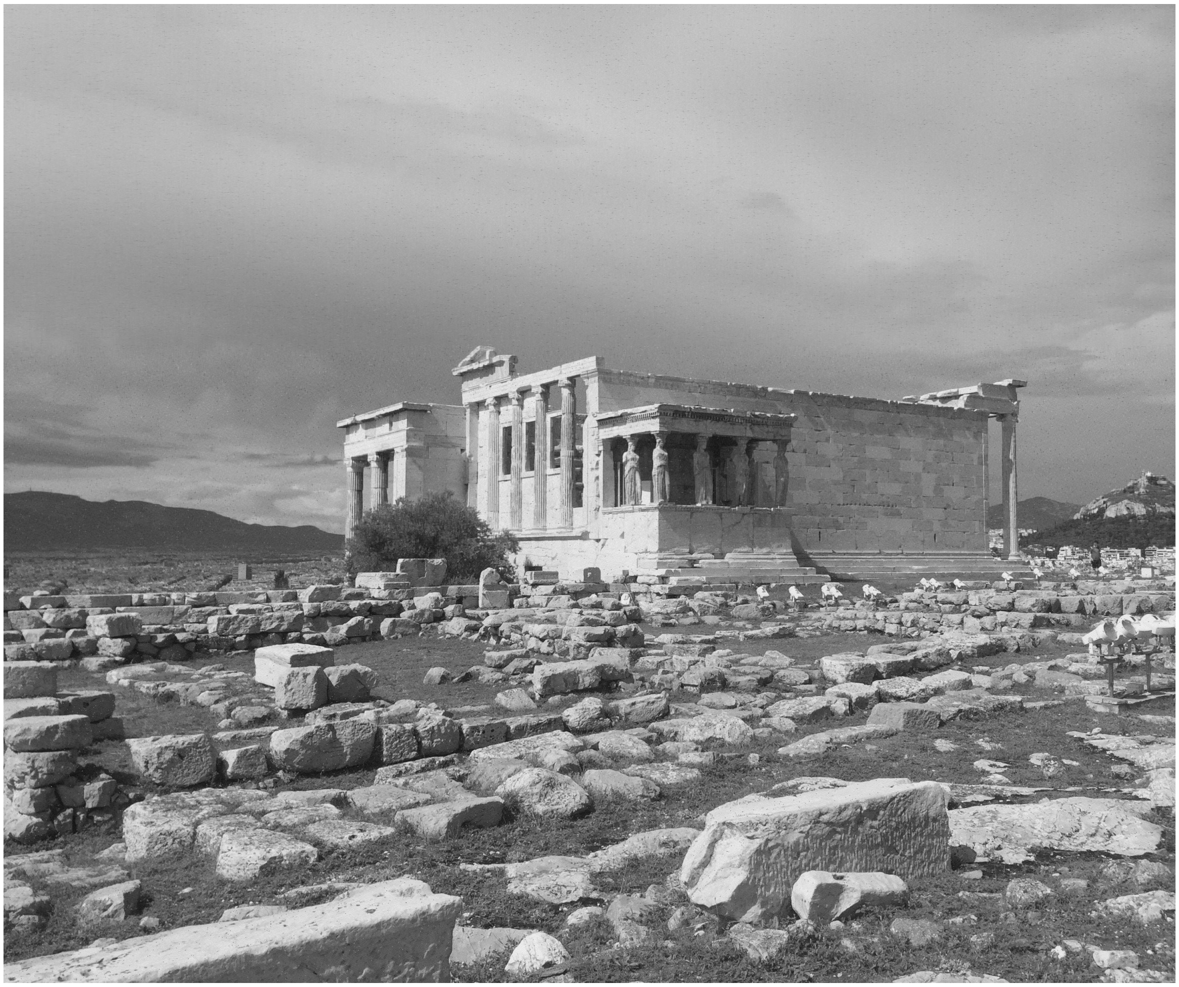

Figure 5: Denoised image using threshold of 1000.

Figure 6 shows the denoised image using a much larger threshold: 3000. In this case the image looks way too smooth, as if it were painted, with the edges fading and a decreased contrast. The photo loses details and quality when applying such a high threshold.

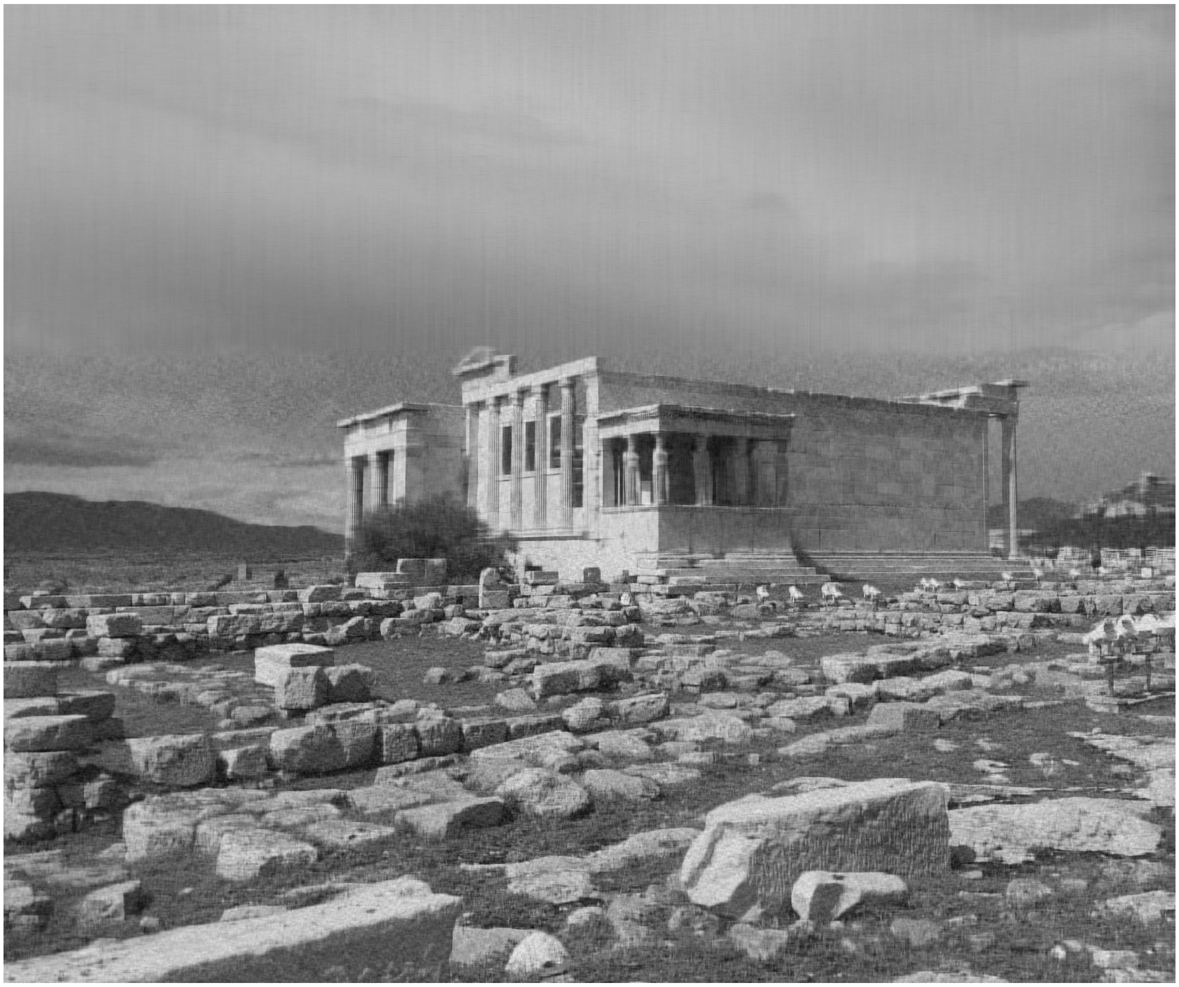

Figure 6: Denoised image using threshold of 3000.

Figure 7 shows the denoised image using a smaller threshold: 500. In this case the image looks almost identical to the original. This suggests that there are not many singular values below the threshold, if any. Therefore, there is no reduction in the noise. This threshold is not enough for noise reduction.

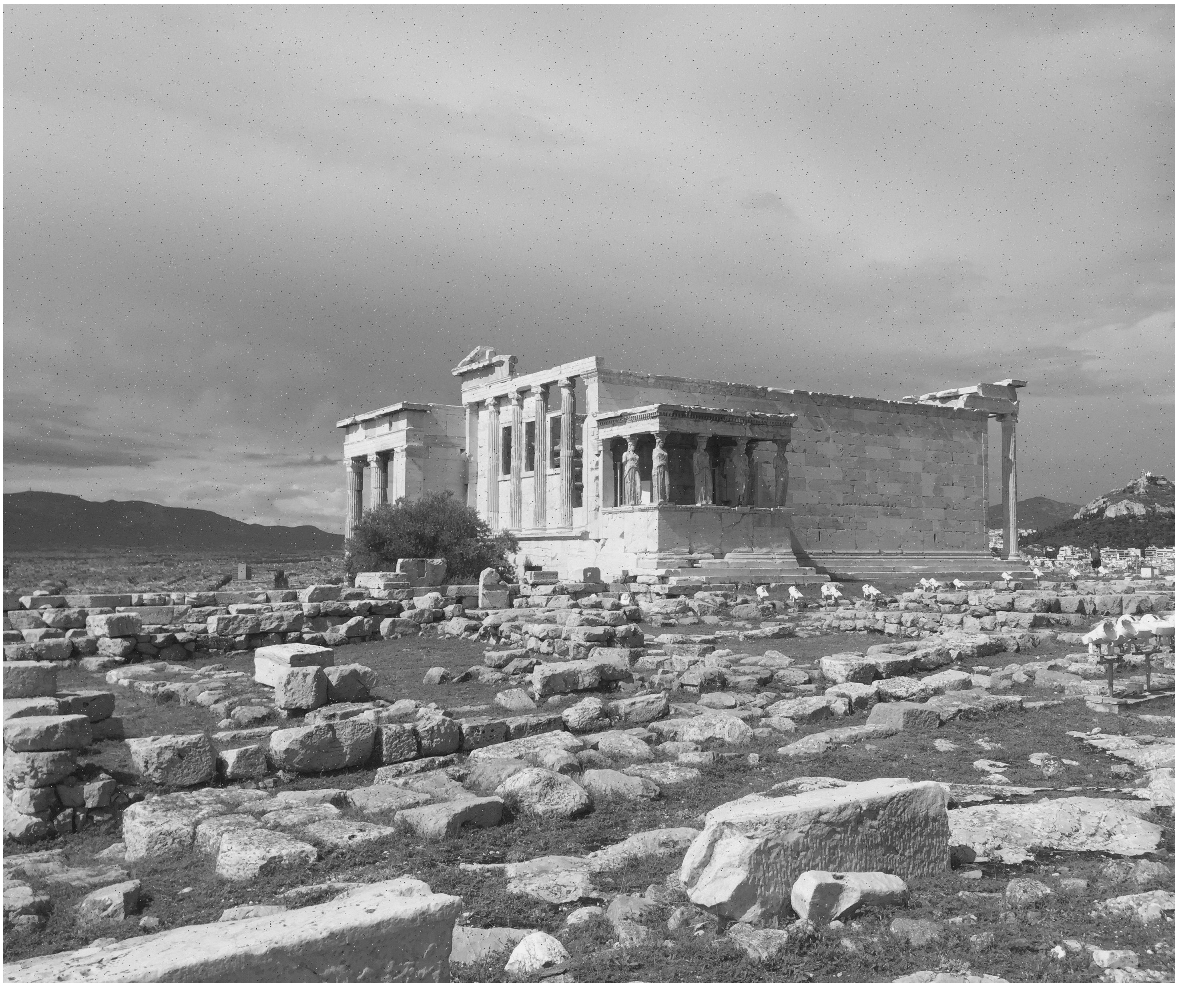

Figure 7: Denoised image using threshold of 500.