Computational Physics HW6

Ben Johnston

October 2023

1 Problem 1

This problem involved performing a simple Principle Component Analysis (PCA) on a data set consisting of the central optical spectra of 9,713 nearby galaxies from the Sloan Digital Sky Survey. The raw spectral data for the first galaxy in the data set is shown below in *Figure 1*:

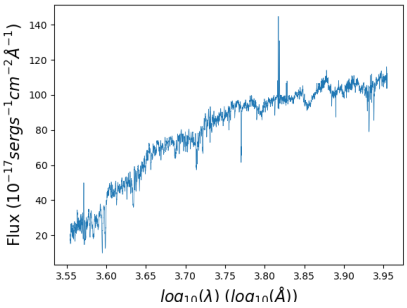

Figure 1: Un-normalised flux as a function of wavelength.

To make the PCA more meaningful, there were two processing steps that were undertaken. As the galaxies are at different distances, their fluxes span a very large dynamic range, so therefore the first step was to normalise these fluxes so that their integrals over wavelength were the same. The integral was estimated by summing over the spectral values for each galaxy. Second, the mean flux at every wavelength is positive, meaning that a PCA will spend an eigenvector to explain the mean offset from zero. Instead, the mean $\vec{f}_m$ of the normalised spectra was subtracted, leaving residuals of all galaxies i varying around zero of the form:

$$\vec{r}_i = \vec{f}_i - \vec{f}_m \tag{1}$$

The normalised spectra for the first galaxy in the data set is shown in *Figure 2* below alongside a check to make sure the flux is correctly normalised:

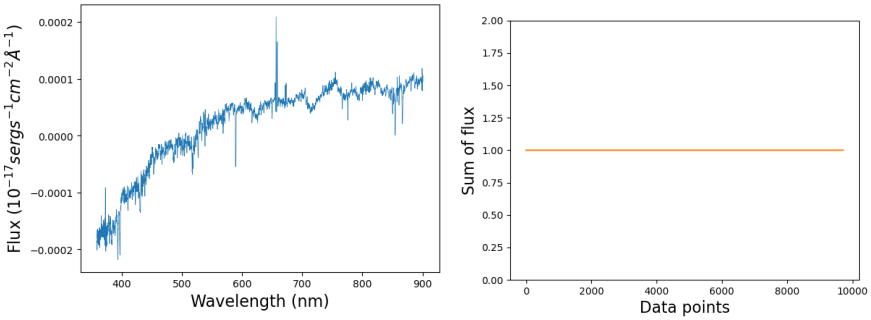

Figure 2: Left: Normalised flux as a function of wavelength. Right: Normalisation check.

Following on from this, the PCA was then performed - in essence the PCA aimed to find the eigenvectors of the covariance matrix of the distribution. This covariance matrix can be calculated using the following equation:

$$\mathbf{C} = \frac{1}{N_{gal}} \sum_{ij} \vec{r}_i \vec{r}_j \tag{2}$$

where i and j index the galaxies. The residuals were then recast as a matrix R_{ij} we can write the covariance matrix as:

$$\mathbf{C} = \mathbf{R} \cdot \mathbf{R}^T \tag{3}$$

The above matrix was then calculated, which was then checked to confirm that the dimensions were $N_{wave} \times N_{wave}$; the dimensions of $\mathbf{C}$ were indeed 4001×4001. The eigenvectors and eigenvalues were calculated using NumPy's linalg module which took 27.83855390548706s, following which the first 5 eigenvectors were plotted as seen in *Figure 3*:

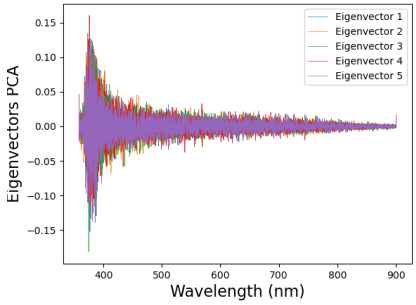

Figure 3: PCA Eigenvectors as a function of wavelength.

It is also possible to find the eigenvectors directly from $\mathbf{R}$ using Singular Value Decomposition (SVD), which was performed using NumPy's linalg module - this calculation took 37.904186725616455s. This calculation therefore took longer to compute than teh corresponding PCA calculation. The first 5 eigenvectors calculated using SVD are plotted in *Figure 4*:

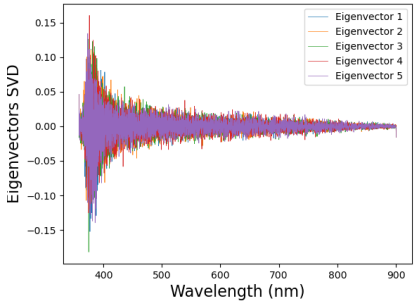

Figure 4: SVD Eigenvectors as a function of wavelength.

In order to confirm that these eigenvectors are the same, graphs of SVD eigenvectors vs PCA eigenvectors and SVD eigenvalues vs PCA eigenvalues as seen in *Figure 5*:

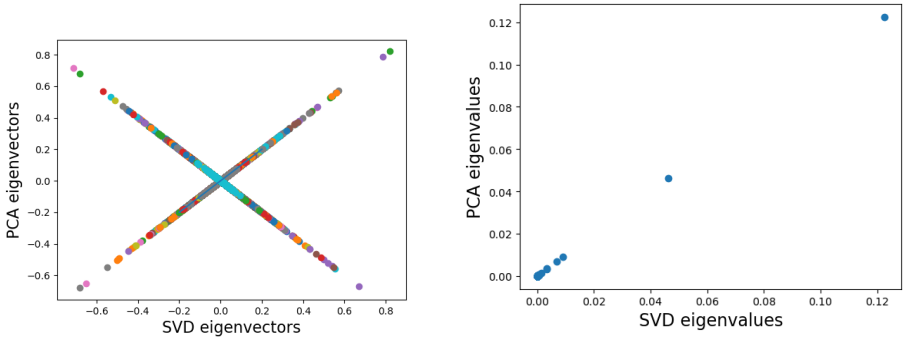

Figure 5: Left: Eigenvector plot. Right: Eigenvalue plot.

Considering first the eigenvector plot it can be seen that some of the SVD eigenvectors are reflected about $x = 0$. This is because eigenvectors found through SVD are unique up to a sign change between the left and right singular vectors. However, considering the eigenvalues it can be seen that the eigenvalues match between PCA and SVD and so it can be concluded that everything looks ok.

The next part of the problem involved comparing the condition numbers of the covariance matrix $\mathbf{C}$ used for PCA and the residuals matrix $\mathbf{R}$ used for the SVD calculation. The condition number of $\mathbf{C}$ is 62384247000.0 while the condition number of $\mathbf{R}$ is 6561841.5. Both of these numbers are very high, however the condition number of $\mathbf{R}$ is 4 orders of magnitude smaller than that of $\mathbf{C}$, meaning that $\mathbf{R}$ is a more viable matrix to perform calculations with.

Following this, the problem involved reducing the dimensionality of the PCA calculation in order to model the data. This was done using the following steps:

1. Get the eigenvalues and eigenvectors of the correlation matrix C
2. Project $\mathbf{R}$ onto the l eigenvectors with the largest l eigenvalues.
3. Multiply each eigenvector by its weight to "filter" the original signal.

Figure 6 below shows this method for the first 5 eigenvectors and all 4001 eigenvectors in order to check this fully reproduces the data set.

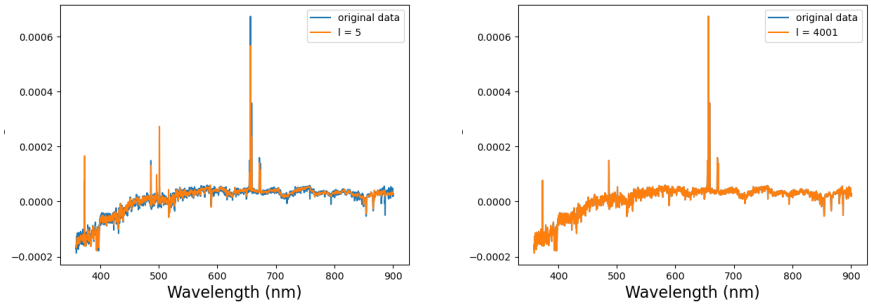

Figure 6: Left: 5 eigenvectors. Right: 4001 eigenvectors.

Next, plots of c_1 vs c_0 and c_2 vs c_0 were produced and can be seen in *Figure 7* below:

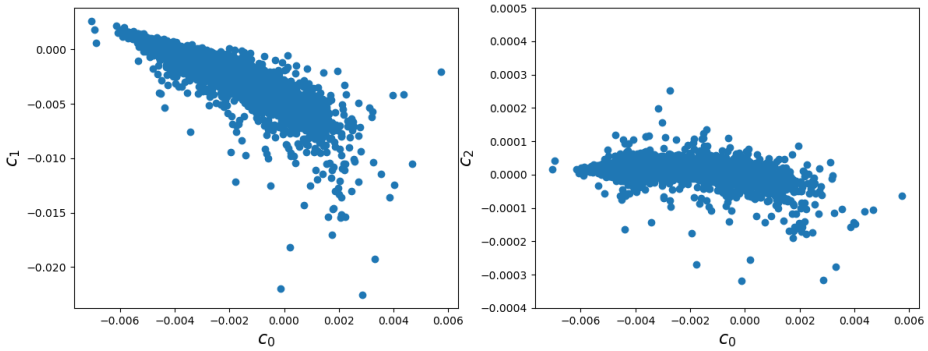

Figure 7: Left: c_1 vs c_0. Right: c_1 vs c_0

Finally, the number of eigenvectors used to model the data was varied from $1 \leq N_c \leq 20$, from which the squared residuals were calculated and plotted as a function of N_c as seen in *Figure 8*:

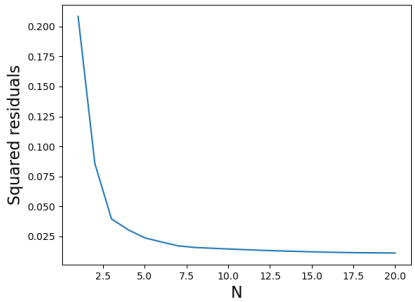

Figure 8: Squared residuals as a function of N_c

It can be seen that as the number of eigenvectors increases the squared residuals decrease, which is expected and so it can be concluded that everything is working as it should be.