PYU44C01 Assignment 4

Alexander Hackett
15323791
Prof. Charles Patterson

1 Question 1

The SVD script in *ahackett_Assignment4.py* enclosed in the standalone function *singularValueDecom()* takes the input matrix, A, and then computes $A^T \cdot A$ and $A \cdot A^T$. The matrix Q_1 is then constructed from the eigenvectors (sorted according to the abolute values of the corresponding eigenvalues) of $A \cdot A^T$, and Q_2 was constructed from the eigenvectors of $A^T \cdot A$ in an analogous way. Q_2^T was determined by just taking the transpose of Q_2. Both $A^T \cdot A$ and $A \cdot A^T$ have the same eigenvalues. The matrix Σ was constructed by taking the square roots of the non-zero eigenvalues of $A \cdot A^T / A^T \cdot A$ and constructing a diagonal matrix of appropriate size from them. Given the input matrix:

$$\begin{bmatrix} 1 & 3 & 3 & 2 \\ 2 & 6 & 9 & 5 \\ -1 & -3 & 3 & 0 \end{bmatrix}$$

The *singularValueDecom()* function produced the following matrices:

$$\mathrm{Q}_1 = \begin{bmatrix} 0.36397892 & 0.1848946 & -0.91287093 \\ 0.93049972 & -0.02892972 & 0.36514837 \\ 0.04110486 & -0.98233246 & -0.18257419 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 12.98482306 & 0 & 0 & 0 \\ 0 & 4.40390397 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{Q}_2^T = \begin{bmatrix} 0.16818662 & 0.50455986 & 0.7385352 & 0.41436502 \\ 0.2519055 & 0.7557165 & -0.60234761 & 0.05112296 \\ 0.95302512 & -0.28879549 & 0.02887955 & -0.08663865 \\ -0.21629469 & -0.22809957 & -0.3001978 & 0.9005934 \end{bmatrix}$$

In order to confirm that the SVD function was produce the correct result, the original matrix, A, was reconstructed as $A = Q_1 \cdot \Sigma \cdot Q_2^T$. As desired, this reproduced faithfully the original A matrix as had been input.

2 Question 3

2.1 i

The method, *PCA.centering()* in the *PCA* class written for this assignment, was utilized in order to determine the mean of each column in the dataset contained in the supplied file *iris.data*, and to subtract the column average from the entries in each column so that the mean of each column in the dataset is zero. This new, column-centered dataset was designated X.

2.2 ii

The method *coVar()* was then utilized to determine the covariance matrix of X, designated C by computing $C = \frac{X^T \cdot X}{n-1}$, where n is the sample size. The covariance matrix is, in this case, a 4 × 4 matrix describing the linear relationships between the 4 variables in the original dataset; sepal length, sepal width, petal length and petal width. The covariance matrix, C, produced was:

$$\mathrm{C} = \begin{bmatrix} 34.05611111 & -1.95033333 & 63.25955556 & 25.67288889 \\ -1.95033333 & 9.33753333 & -15.9784 & -5.85973333 \\ 63.25955556 & -15.9784 & 154.62124444 & 64.38724444 \\ 25.67288889 & -5.85973333 & 64.38724444 & 28.92657778 \end{bmatrix}$$

A positive covariance between variables indicated that one of the pair of variables has the tendency to increase when the other variable in the pair increases, and decrease when the other variable in the pair decreases, whereas a negative covariance indicates that the variable tends to decrease when the other variable in the pair increases, and vice versa. Naturally, the covariance between a variable and itself, (the diagonal entries) are large and positive. Additionally, one can see that sepal length and petal length both tend strong to increase and decrease in unison, whereas increasing petal length tends to reduce sepal width, and vice versa.

2.3 iii

The method *eignDecomC()* was used to determine the eigendecomposition of the covariance matrix C, as $C = V \cdot \Lambda \cdot V^T$, where V contain the eigenvectors of C and Λ is constructed from the eigenvalues of C. The decomposition produced the matrices:

$$\mathrm{V} = \begin{bmatrix} 0.36158968 & -0.65653988 & -0.58099728 & 0.31725455 \\ -0.08226889 & -0.72971237 & 0.59641809 & -0.32409435 \\ 0.85657211 & 0.1757674 & 0.07252408 & -0.47971899 \\ 0.35884393 & 0.07470647 & 0.54906091 & 0.75112056 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 209.83375816 & 0 & 0 & 0 \\ 0 & 12.03143072 & 0 & 0 \\ 0 & 0 & 3.90002077 & 0 \\ 0 & 0 & 0 & 1.17625701 \end{bmatrix}$$

and where V^T is just the transpose of the matrix V above. The matrix V describes the principle axes of the dataset. Helpfully thought of as the axes of a four dimensional ellipsoid being fit to the covariance.

2.4 iv

If the eigenvectors of C described by V form the principle axes of the covariance, the principle components of the dataset can be determined by projecting the centered dataset, X, onto the new basis vectors described by V, so that the principle components of the data are $PCA = X \cdot V$.

2.5 v

As noted, and commonly used, an SVD can be utilized to determine the covariance matrix C, while avoiding the computationally expensive step of constructing $X^T \cdot X$. As in Question 1, the SVD of the matrix X, representing the centered dataset was constructed as $X = U \cdot S \cdot V^T$, and hence the covariance matrix C was also constructed as $C = \frac{V \cdot S^2 \cdot V^T}{n-1}$, where, as before, n is the sample size. It was then confirmed that the same C matrix was produced each time, indicating that SVD could be utilized to more efficiently compute the covariance matrix when performing PCA.

2.6 vi

Finally, the method *plotAnalysisDiagram()* was used to produce a plot of the first two Principal Components of the dataset, reducing the four dimensional iris dataset down to a two dimensional dataset that is easy to visualize, and still allows the three different species of irises to be distinguished. This is because the first two PCs account for the most variance in the dataset, and hence "contain" the most information. The following plot was produced.

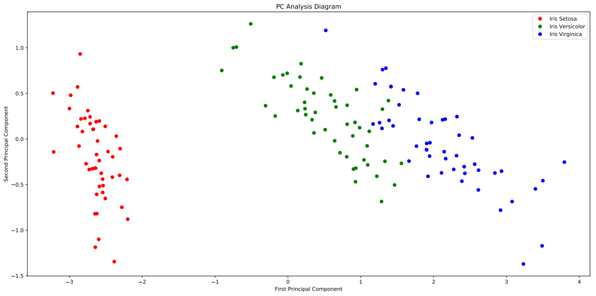

Figure 1: Principal Component Analysis Diagram of the First Two Principal Components of the iris dataset. This analysis allowed the 4-D dataset to be reduced to 2 dimensions, and the clusters corresponding to the three different species of iris, (*setosa*, *versicolor* and *virginica)* can be clearly distingushed.

In order to demonstrate the effectiveness of the PCA approach, the following plot containing the first three Principal Components was produced:

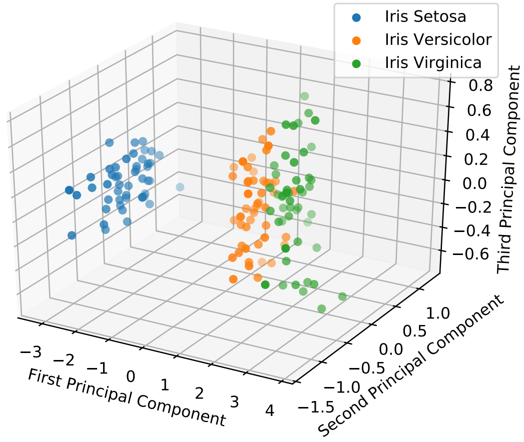

Figure 2: Principal Component Analysis Diagram of the First Three Principal Components of the iris dataset. When the dataset is reduced from 4 to 3 dimensions, i.e, one extra Principal Axes than the 2D plot, there is not very much more information present. Hence, the PCA can allow the different species of iris to be identified using only the first two Principal Components, since, as discussed, these contribute the most to the varience of the dataset.

3 Conclusion

The Principal Component Analysis technique (PCA) was investigated with regards to its application to the iris dataset. The dataset was centered, and the covariance matrix was computed directly. It was also noted that the covariance matrix could have been initially constructed from the singular value decomposition of the centered dataset, and this in general would be a less computationally expensive method of doing so. The eigendecomposition of the covariance matrix was utilized in order to determine the principal directions of the dataset, which were the eigenvectors of the covariance matrix, and the Principal Components of the dataset were determined by projecting the centered dataset onto the basis formed by the principal direction vectors. A plot of the first two Principal Components was produced, and the three clusters corresponding to the three separate species of iris could be clearly identified, although the differences between the *setosa* iris and the other two was considerably greater than between the *versicolor* and the *virginica* irises. A plot of the first three Principal Components of the dataset was also produced in order to demonstrate that very little extra information is gleaned by including the next most significant Principal Component direction. This is because the first two Principal Components contribute most to the variance of the dataset, and hence, contain the most information about the variance of the dataset. Hence, it was demonstrated successfully that PCA allows this 4D dataset to be reduced to and analyzed successfully in two dimensions, without significant loss of information and in a manner considerably more conductive to simple visualization.