Coursework 2: Principal Component Analysis

This coursework is concerned with covariance estimation and finding principal components. It uses NumPy's linear algebra module (linalg) and its image handling capabilities. The data to be used is the *HepatitisC* data set, the set of six face images a.pgm, ..., f.pgm, and the file *PrincipalComponents.txt*. The file *CourseworkLibrary* contains three functions which will read images and turn them into data sets and vice versa.

Task 2.1: 3 marks

Complete the function *Mean* which calculates the mean vector of a data set represented (as usual) by a matrix in which the rows are data points and the columns are variables.

Task 2.2: 4 marks

Complete the function *Covariance* which calculates the covariance matrix of a data set represented as above.

Task 2.3: 3 marks

For this part, you will need to use the file named *PrincipalComponents.txt* which is on the course web page. It contains a set of ten principal components (the last having a zero eigenvalue) calculated from ten different images of the subject in image c.pgm. You can read it with the function *ReadEigenfaceBasis()* which is in the file *CourseworkLibrary.py*. It returns an array where each row is an eigenface. You will also need the mean face for this basis which is in the file *MeanImage.jpg*. You can read this in with the library function *ReadOneImage(filename).*
Complete the function *CreateEigenfaceFiles* to create image files of the principal components (eigenfaces) in the format returned by *ReadEigenfaceBasis()*. You can use the *CourseworkLibrary* file *SaveEigenface* to do the image handling. Don't use the file format .pgm (eg instead choose .jpg or .png) when creating these image files. This is to avoid problems later. You can create a series of file names for the different eigenfaces within a loop using python's powerful string concatenation feature, for example, *filename = "PrincipalComponent" + str(j) + ".jpg"*

Task 2.4: 3 marks

Complete the function ProjectFace, which reads one image file (use *c.pmg* for your example) and projects it onto the principal component basis.

Task 2.5: 3 marks

Complete the function *CreatePartialReconstructions* which generates and saves image files of the reconstruction of an image from its component magnitudes. Generate files of the mean face, the mean face plus first principal component ... and so on up to the mean face plus all principal components. (Note the function *SaveEigenface* normalizes the eigenface before writing an image file so the mean face that you generate will have more contrast than the input file *MeanImage.jpg*)

Task 2.6: 7 marks (hard)

Complete the function PrincipalComponents which performs PCA on a data set. To do this you will need to use the Kohonen Lowe method outlined towards the end of the lecture on PCA. Your function should return the orthonormal basis as a list of eigenfaces equivalent to the one returned by ReadEigenfaceBasis(). Check out the function *numpy.linalg.eig* before you try to compute the eigenvectors yourself! To check out your Principal components you should use the face images a.pgm to f.pgm. You can create an image data set using the *CourseworkLibrary*, for example: imageData = array(ReadImages()). You can find the mean image from your solution to 2.1.

Results

Create a results file as before with the following items.

1. A title giving your group members
2. The mean vector *HepatitisC* data set
3. The covariance matrix of the *HepatitisC* data set
4. The component magnitudes for image "c.pgm" in the principal component basis used in task 2.4.

Put together a file "Results2.pdf" which contains the above results together with the eigenface images you created in task 2.3 and the partial reconstructions of image c.pgm created in task 2.4. If you did Task 2.6, include the eigenfaces found in the basis you calculated from the images and the partial reconstructions of image c.pgm using that basis. Submit it with your Python code, named "Coursework2.py".