CS 6220 Data Mining — Assignment 3

Face Recognition using PCA

For this assignment, you will be using a PCA to reduce the dimension of data, and use the reduced data for face recognition.

Objectives:

1. Employ data reduction techniques such as principal component analysis
2. Use PCA to generate eigenfaces from face images
3. Implement face recognition technique using eigenfaces

Submission:
Submit your ipynb via canvas.

Grading Criteria:
Follow the instructions in the pdf, and complete each task. You will be graded on the application of the module's topics, the completeness of your answers to the questions in the assignment notebook, and the clarity of your writing and code.

Dataset:
For this assignment, you will use the dataset from 'AT&T Database of Faces'. Download the dataset from here. The dataset contains a total of 400 face images of 40 different subjects having 10 images each. The image files are in PGM format. The size of each image is 112×92 pixels, with 256 grey levels per pixel. The images are organised in 40 directories (one for each subject), which have names of the form sX, where X indicates the subject number (between 1 and 40). In each of these directories, there are ten different images of that subject, which have names of the form Y.pgm, where Y is the image number for that subject (between 1 and 10).

What You Need to Do

Part 1 - Visualizing the Face Images [10 Points]
You can use the following code to read and visualize an image. Code shown for reading the first image of the first subject.

```
with open(‘face_data/s1/1.pgm’, ‘rb’) as image:
    img = cv2.imdecode(np.frombuffer(image.read(), np.uint8), cv2.IMREAD_GRAYSCALE)

plt.imshow(img, cmap="gray")
```

1. Visualize randomly selected 16 faces in a 4×4 grid [4 rows and 4 columns].
2. Report the face image size, number of images and number of classes.

Part 2 - Train Test Split [10 Points]

1. Generate Train and Test set from the dataset in the following manner.
2. Select the first nine images of each subject for the Train set.
3. Select the last image of each subject for the Test set.
4. Flatten each image into 1D vector so that the dataset size is $N \times L$, where N is the number of samples in the train/test set and L is the length of flattened image ($L = 92 \times 112 = 10304$).
5. Report the number of images in Train set and Test set.

Part 3 - Apply PCA to Get Eigenfaces [30 Points]

1. Apply PCA using scikit-learn on the Train set
2. Take first 20 principal components in the feature space. These are known as eigenfaces.
3. Visualize the first 16 eigenfaces. For visualization, reshape the flattened vector to original image shape.

Part 4 - Face Recognition [30 Points]
For face recognition, we first need to calculate weights for each sample in the Train set with respect to each principal component. For a test image, we again calculate a weight and measure euclidean distance from each sample of the Train set. The class of the sample with the minimum distance is the predicted class for the test image.

```
    train_sample_weights = np.matmul(eigenfaces, (train_facematrix - pca.mean_).T)
```

Here, $eigenfaces$ is of shape $K \times L$, $train\_facematrix$ is of shape $N \times L$, *pca.mean_* is a vector of length L, and $train_sample_weights$ is of shape $K \times N$. K denotes the number of first principal components.

1. Calculate the weights of the training samples using the given formula.
2. For each test image, calculate weights similarly.
3. Take the minimum euclidean distance between the test image weight and all the training sample weights to predict the class of the test image.
4. For each test image, Visualize the test image and the train image with minimum distance.
5. Report accuracy and total explained variance ratio by the selected components.

Part 5 - Face Recognition [20 Points]

1. Repeat Part 4 using only first two principal components instead of 20. Visualize the first two eigenfaces.
2. Repeat Part 5.
3. Compare the results using the explained variance ratio of PCA.