24-787 Homework 10

Due date: 04/24/2021 @ 11:59 pm EST

Note: In case a problem requires programming, it should be programmed in `Python`. In Programming, you should use plain `Python` language, unless otherwise stated. For example, if the intention of a Problem is familiarity with numpy library, it will be clearly noted in that problem to use numpy. Please submit your homework through Gradescope.

Submissions: There are two steps to submitting your assignment on Gradescope:

1. **HW10_Writeup**: Submit a combined pdf file containing the answers to theoretical questions as well as the pdf form of the FILE.ipynb notebooks.

- To produce a pdf of your notebooks, you can first convert each of the .ipynb files to HTML.
- To do this, simply run: `ipython nbconvert -to html FILE.ipynb` for each of the notebooks, where FILE.ipynb is the notebook you want to convert. Then you can convert the HTML files to PDFs with your favorite web browser.
- If an assignment has theoretical and mathematical derivation, scan your handwritten solution and make a PDF file.
- Then concatenate them all together in your favorite PDF viewer/editor. The file name (FILE) for naming should be saved as HW-assignmentnumber-andrew-ID.pdf. For example for assignment 1, my FILE = HW-1-lkara.pdf
- Submit this final PDF on Gradescope, and **make sure to tag the questions correctly!**

2. **HW10_Code**: Submit a ZIP folder containing the FILE.ipynb notebooks for each of the programming questions. The ZIP folder containing your iPython notebook solutions should be named as HW-assignmentnumber-andrew-ID.zip

Q1: PCA (50 pts)

(a) Principle Component and Transformed space (20 pts)

You are given a two-dimensional dataset `q1adata.txt`:

$$X = \begin{bmatrix} 2.5 & 2.4 \\ 0.5 & 0.7 \\ 2.2 & 2.9 \\ 1.9 & 2.2 \\ 3.1 & 3.0 \\ 2.3 & 2.7 \\ 2.0 & 1.6 \\ 1.0 & 1.1 \\ 1.5 & 1.6 \\ 1.1 & 0.9 \end{bmatrix}$$

1. Compute the first and second principal components (e_1 and e_2) of this dataset. Show all your work. Using `Python`, plot the data in the original feature space and show the principal components. Submit both calculation and the plot.
2. Transform the data using both principal components (i.e. compute a_1 and a_2 for each data point) and plot this new representation on the e_1, e_2 plane. Submit both calculation and the plot.
3. What is the PCA-optimal one-dimensional representation of the data? In this reduced dimension, what is the range of the data (the distance between the minimum and maximum points)? Submit reasoning for your answer and the range.

(b) PCA (30 pts)

Now, suppose your data is as given as `q1bdata.txt`. In this case, the number of samples – 4 is much less than the number of dimensions – 19, so you will need to use an approach similar to the face recognition problem discussed in lecture slides.

1. Compute the 4×4 inner product matrix and find all non-zero eigen vectors. How many such vectors are there? Show all steps in your derivation. Submit derivation and your answer.

2. Calculate the projection of each data point onto a reduced 3D space. Show all steps in your derivation. Report the 4×3 matrix. Submit derivation and your answer.

3. Determine the mean-squared error between each original sample and its reconstructed version using the three eigenvectors. Note that the MSE should be computed in the original space. Submit calculation and your answer.

4. Repeat part (3), but this time assume the dimensionality of the data is reduced to two instead of three. Submit calculation and your answer.

5. What is the Euclidean distance between the new data vector given below and each of the four samples in the reduced three-dimensional space? Which of the four samples is most similar to this new vector? Submit calculation and your answer.

$$Y = [1\ 3\ 0\ 3\ \text{-}2\ 2\ 4\ 1\ 3\ 0\ \text{-}2\ 0\ 1\ 1\ \text{-}3\ 0\ 1\ \text{-}2\ \text{-}3]$$

6. Perform the same analysis as part (5) (*i.e.* nearest-neighbor classifier), but this time in the original dimension space. Do the results match? Does this make intuitive sense? Why or why not?

Q2: Eigenface with PCA (50 pts)

In this question, you will be developing a simple classifier to recognize human faces with the help of **PCA** to reduce dimension. The image dataset `data.mat` (from Yale face database) is provided with the assignment. It consists of 165 grayscale images (15 subjects, 11 images each), each of size 231×195. To load `data.mat` dataset, you can do the follows and the loaded `mat` will be in the form of a dictionary with the keys corresponding to the provided information variables.

```
import scipy.io
mat = scipy.io.loadmat('data.mat')
print(mat.keys())
```

In order to access the values of specific keys, for instance X, you can get it by just calling the key name: `mat['X']`. The provided information variables include:

- **X**: the data, encoded as a 45045×165 matrix, in which the rows represent dimensions (pixels) and the columns indicate examples. To view an image, simply reshape the appropriate column vector into an array with size 195×231 and then do a transpose. For instance: `sample_face = np.transpose(X[:,102].reshape(195,231))`
- **Y**: the labels, given as a 165×1 vector. Each label takes an integer value from 1-15, indicating which subject is shown in the corresponding image.
- **testimages**: a vector of indices indicating which images to withhold for testing. For example, if the number 10 is included in this vector, do not use the 10th image when computing principal components.
- **trainimages**: the complement of **testimages**, provided for completeness.

(a) (5 pts) Load the dataset and plot the $0^{th} - 4^{th}$ faces, you should see the same person on these five images if you load the dataset correctly.
Submit your code together with the plotted human faces.

(b) (25 pts) Reduce the dimensionality of the data using PCA such that 90% of the variance is preserved. That is, find the number of eigenvectors **E** such that the sum of the first **E** eigenvalues accounts for 90% of the total sum of all eigenvalues. What is the new number of dimensions **E**? In your submission, provide visualizations of the first five eigenfaces (those eigenvectors with the highest eigenvalues) and include a bar graph showing the percentage of variance explained by each principal component in descending order. You could use the `sklearn` library to run the PCA algorithm in this question.
Submit your code, number of eigenvectors E, plot the first 5 eigenfaces as well as the above-mentioned bar graph.

(c) (10 pts) Try reconstructing the first (0^{th}) training example with varying number of eigenfaces. In your report, juxtapose the original image with its reconstructed versions using {10, 20, 30, 40, 50} eigenfaces respectively. You should be able to see the reconstructed versions getting clearer as you increase the number of eigenfaces.
Submit your code, plot of the original image and five reconstructed images.

(d) (10 pts) Implement the k-nearest neighbor (KNN) algorithm with $k = 1$ in the 30 dimensional reduced space to classify each of the test images. Report the overall test accuracy. Repeat the KNN on test images, but in the original dimension space. Does the accuracy change? Is this the result you would expect? Why or why not? You could use the `sklearn` library to run the KNN algorithm in this question.
Submit your code and the accuracy of your classifier in thee dimensional reduced space as well as the original dimension space; submit the descriptive answers.