CSCE 421: Spring 2023 Homework 5

Assigned March 28, due on April 9, 11:59 PM.

Submit your assignments (written+coding) separately on gradescope. Please name your coding assignment as '**assignment5.py**'. Use the provided python template file, and complete ONLY the functions (DO NOT edit function definitions, code outside the function, or use any other libraries).

A Few Notes:

- Coding assignments should be done only in Python.
- Please start early! This includes learning how to use Latex!
- For solution please use the following template `https://www.overleaf.com/latex/templates/neurips-2022/kxymzbjpwsqx`.
- This is an individual assignment. While you are welcome to discuss general concepts together and on the discussion board your solutions must be yours and yours alone.
- **SHOW YOUR WORK.**

Problem 1: Principal Component Analysis. In this problem, we will process face images coming from the Yale Face Dataset: `https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database`. This dataset contains images of the faces of 15 individuals. For each individual there are 11 images taken under a variety of conditions e.g., the person makes a happy expression, wears glasses etc.

Download the dataset from the above URL.

(a) [Code] Processing the data and calculating eigen values.

  (1) Fill in the function *qa1_load*, which takes the folder name input, and returns the data (as a tuple). Please use **matplotlib.image.imread** to read images.

  (2) Fill in the function *qa2_preprocess* that performs a min max scaling on the faces (the X in dataset). Please use **preprocessing.MinMaxScaler**.

  (3) Fill in the function *qa3_calc_eig_val_vec*, given the dataset and integer k returns the k eigen vectors (PCA components) and the corresponding to the top k eigenvalues. *Hint: use PCA already imported from sklearn.*

(b) [Written + Code] Plot a curve displaying the first k eigenvalues $\lambda_1, ..., \lambda_k$ i.e. the energy of the first K principal components. How many components do we need to capture 50% of the energy? Report the curve and the answer to the question in the report. Fill the function *qb_plot_written* used to generate the plot. DO NOT place your code in any other function.

(c) [Written + Code] PCA and Eigen Faces

  (1) [Code] Fill in the function *qc1_reshape_images*, that returns eigen faces, given the image dimensions, and PCA object. Note: Eigen faces are re-shaped eigen vectors in the shape of the original image.

  (2) [Written + Code] Plot any 10 eigen faces for values of k = len(dataset) (as given in the starter code), and fill in the code *qc2_plot*. There is no specific format for plotting. Place the plots in the report.

(d) [Written + Code] Projection and Reconstruction

   (1) [Code] Fill in the function *qd1_project* that takes the entire dataset and the PCA objects and projects it. *Hint: Use PCA.tranform*

   (2) [Code] Fill in the function *qd2_reconstruct* that reconstructs the dataset given the projection (obtained from the previous function) and the fitted PCA object. *Hint: use PCA.inverse_transform.*

   (3) [Written + Code] Select a couple of images from the data. Use the first k eigenfaces as a basis to reconstruct the images (use functions written in previous sub-questions). Visualize the reconstructed images using 1, 10, 20, 30, 40, 50 components. How many components do we need to achieve a visually good result (report the plot and your answer in the report)? Use function *qd3_visualize* to complete this subquestion.

(e) Classification with SVM and Lasso regression post PCA on input data. We will also manually read your code for this question.

   (1) [Code] Fill in the function *qe1_svm* that splits the input data into training and testing. Use as input features the transformed feature space that resulted from PCA. Experiment with a different number of PCA components through a 5-fold cross-validation. Uniformly sample components in range [10, 100] (with a gap of 20 for the sake of homework). User 5-fold cross-validation to build predictors using support vector machines (using radial basis function kernel). The function returns the best k across folds (average over folds), and the recognition accuracy on test set.

   (2) [Code] Fill in the function *qe2_lasso* that splits the input data into training and testing. Use as input features the transformed feature space that resulted from PCA. Experiment with a different number of PCA components through a 5-fold cross-validation. Uniformly sample components in range [10, 100] (with a gap of 20 for the sake of homework). User 5-fold cross-validation to build predictors using lasso regression. The function returns the best k across folds (average over folds), and the recognition accuracy on test set.