CMPE 462 Machine Learning
Spring 2020 Project III
Due: June 14 by 11.59pm

Project Description

In this project, you are going to implement two unsupervised learning techniques. In the first task, you are asked to implement k-means clustering algorithm using the data provided in `kmeans_data.zip` folder. In the second task, you need to implement PCA and apply dimensionality reduction on the data provided in `USPS.mat`.

Task 1 (50 pts)

Please download `kmeans_data.zip`. In this problem, ground truth cluster assignments are given in `labels.npy`. Please do the following.

1. (10 pts) Plot the data using scatter plot. Assign different colors to different classes.
2. (30 pts) Implement k-means clustering algorithm by yourself using the number of iterations as the stopping condition. You can use built-in functions only for side-tasks such as norm computation, minimum element search and mean calculation, not for the clustering itself.
3. (10 pts) Run k-means 9 times with number of iterations $N = \{1, 2, ..., 9\}$. Plot the final clustering assignments as a scatter plot for each run as 3x3 `matplotlib subplot`. Visually investigate the effect of the number of iterations on obtaining the optimal clustering and find the convergence point by comparing the plots with the one in Task 1.1. If the model does not converge at 9 iterations, you can select 9 other N to effectively show the progress of the clustering.

   For a fair comparison, start each run with the same initial random assignments. You can use `np.random.seed(1)` to this purpose.

Task 2 (50 pts)

Please load the whole dataset in `USPS.mat` into Python using the function `loadmat` in `Scipy.io`. The matrix A contains all the images of size 16 by 16. Each of the 3000 rows in A corresponds to the image of one handwritten digit (between 0 and 9). Please do the following.

1. (30 pts) Implement PCA and apply it to the data using $d = 50, 100, 200, 300$ principal components. You are not allowed to use an existing implementation. You can use existing packages for eigen-decomposition. Do not forget to standardize the data before eigen-decomposition.
2. (15 pts) Reconstruct images using the selected principal components from Task 2.1

3. (5 pts) Visualize the reconstructed images for the images at indices $i = 0, 500, 1000, 2000$ for $d = 50, 100, 200, 300$. Create a 4x5 subplot where the rows correspond to images at each index, first four columns correspond to reconstructed images using each d and the last column is the raw image, i.e. before PCA. Comment on your results.

Submission Guidelines

- Download the provided .zip file.
- Prepare a comprehensive report in pdf format. Include your tables, and comments for each task. Follow academic writing rules. Prefer a concise and clear language. You do not need to include code snippets in your report.
- Submit the completed .ipynb and .pdf files through the assignment *Project 3* on Moodle. Name your submission files with the student IDs of the group members (i.e. 2015400XXX_2015400YYY.ipynb). Please submit your pdf report through Turnitin assignment.
- The submitted .ipynb file should run from scratch without errors when accompanied with the provided data. Note that iPython kernel of Jupyter stores variables in the kernel over time and if your code depends on any such variable, it will not run on new sessions. Therefore, make sure that you can run your code from scratch (restart kernel & run all) before submitting the notebook. If your code does not run for any reason, you will be deducted 10 pts.
- Please fairly share the tasks among the group members. Write the student ID/IDs next to each task/code cell to specify who completed which task. If you do not specify any IDs, the instructor will assume each student put equal effort in that task.