CS 391L: Machine Learning **Summer 2023**

Homework 3 - Programming

Lecture: Prof. Adam Klivans

Keywords: SVD, PCA

1. In this problem we will look at image compression using SVD, following the lines of the well-known "Eigenfaces" experiment. The basic concept is to represent an image (in grayscale) of size $m \times n$ as an $m \times n$ real matrix M. SVD is then applied to this matrix to obtain U, S, and V such that $M = USV^T$. Here U and V are the matrices whose columns are the left and right singular vectors respectively, and S is a diagonal $m \times n$ matrix consisting of the singular values of M. The number of non-zero singular values is the rank of M. By using just the largest k singular values (and corresponding left and right singular vectors), one obtains the best rank-k approximation to M.

   We will use the Olivetti faces dataset built into scikit-learn, consisting of 400 64×64 grayscale images. You have two tasks:

   (a) [**10 points**] Given an $m \times n$ image M and its rank-k approximation A, we can measure the reconstruction error using mean ℓ_1 error:

   $$\text{error}_{\ell_1}(M, A) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} |M_{i,j} - A_{i,j}|.$$

   For $k = 1, \ldots, 30$, take the average rank-k reconstruction error over all images in the dataset, and plot a curve of average reconstruction error as a function of k.

   (b) [**6 points**] Pick any image in the dataset, and display the following side-by-side as images: the original, and the best rank-k approximations for $k = 10, 20, 30, 40$. You will find the `imshow` method in matplotlib useful for this; pass in `cmap='gray'` to render in grayscale. Feel free to play around further with higher resolution images to see the progression more clearly.

2. [**10 points**] In this problem we visualize the Wisconsin breast cancer dataset in two dimensions using PCA. First, rescale the data so that every feature has mean 0 and standard deviation 1 across the various points in the dataset. You may find `sklearn.preprocessing.StandardScaler` useful for this. Next, compute the top two principal components of the dataset using PCA, and for every data point, compute its coordinates (i.e. projections) along these two principal components. You should do this in two ways:

   (a) By using SVD directly. Do not use any PCA built-ins.

   (b) By using `sklearn.decomposition.PCA`.

   The two approaches should give exactly the same result, and this also acts as a check that you are doing the right thing. (But note that the signs of the singular vectors may be flipped in the two approaches since singular vectors are only determined uniquely up to sign. If this happens, flip signs to make everything identical again.)

   Your final goal is to make a scatterplot of the dataset in two dimensions, where the x-axis is the first principal component and the y-axis is the second. Color the points by their diagnosis (malignant or benign). Do this for both approaches. Your plots should be identical. Does the data look roughly separable already in two dimensions?