ECE 445 (ML for ENGG): Mini Jupyter Exercise #4

Waheed U. Bajwa (waheed.bajwa@rutgers.edu)

Objective: In this exercise, we will engage in classification of images of handwritten digits '0' and '1' using k-nearest neighbor (k-NN) classification.

Training Dataset

The training dataset for this problem corresponds to the images of handwritten digits '0' and '1' that come prepackaged with the `sklearn` package. There are a total of 360 images of digits '0' and '1'; we will divide them into 300 images for training purposes, while we will evaluate the performance of k-NN classification on the remaining 60 images, which we will refer to as the "test" set.[1] You can use the following code to obtaining the training and the test sets.

```
from sklearn.datasets import load_digits
images, labels = load_digits(2, return_X_y=True)

# Labeled training set
training_images = images[:300]
training_labels = labels[:300]

# Labeled test set
test_images = images[300:]
test_labels = labels[300:]
```

k-NN Classification Using 2-D Features

1. Carry out *principal component analysis* (PCA) of images in the training set and compute two-dimensional PCA features of training images.
   - Display the two-dimensional features of training images as points on a two-dimensional scatter plot. Color all points corresponding to digits '0' as *red* and all points corresponding to digits '1' as *green*.
2. Classify each image in the test set by first transforming it to the two-dimensional PCA domain using the principal components obtained above and then using k-NN classification with $k = 5$ and the distance metric being $\|\cdot\|_2$.
   - Display the two-dimensional features of test images as points on a two-dimensional scatter plot. Color all points that are correctly classified as *blue* and all points that are incorrectly classified as *black*.
3. Compute and display the average classification error for the test set, defined as $\frac{1}{N}\sum_{i=1}^{N} 1_{\{\widehat{y}_i \neq y_i\}}$; here, y_i denotes the true label of the i-th image, $\widehat{y}_i$ denotes the label returned by k-NN, and $N = 60$ in this particular problem.

[1] Principled splitting up of a given dataset into *training*, *testing*, and *validation* sets is an important aspect of machine learning that will be covered in a future lecture.

k-NN Classification Using Higher-dimensional Features

1. Carry out PCA of images in the training set and compute r-dimensional PCA features of training images such that the top-r principal components capture 95% of variation within the training data.
2. Classify each image in the test set by first transforming it to the r-dimensional PCA domain using the principal components obtained above and then using k-NN classification with the distance metric being $\|\cdot\|_2$ and k being an odd integer from 1 to 9.
    - Provide a labeled plot of the average classification error for the test set as a function of k.
    - Based on the plot, what value of k will you recommend be used for future k-NN classification of digits '0' and '1'?