CS 506 FALL 2019 - HW2

Classification and Dimensionality Reduction

Due date: November 6, 2019

1 Logistic Regression and kNN Classification

The goal of this assignment is to perform classification on the famous **MNIST** dataset.
We have already preprocessed a sample of this dataset (30% of the original dataset), that you can find here: Download from Google Drive in the format of NumPy arrays.
File *mnist_data.npy* contains an array of the data -each row corresponds to a $36x36$ digit picture vectorized to create 36*36=1296 features, while the file *mnist_labels.npy* contains the respective labels of the images.

a) [0pts.] Randomly **split the dataset**, using 20% of the samples as your test set and the remaining 80% as the train set that you will use to fit your models.

b) [1pt.] Try to classify the images using **Logistic Regression**. Have in mind that the dataset contains more than 2 labels, hence is a **multinomial classification** problem. What is your **train accuracy and test accuracy**?

c) [2pts.] Now, try to classify the dataset using a **k-Nearest Neighbor classifier**. **Plot the train and test accuracy** as you **vary** k from 1 to 25 with a step size of 2.

d) [1pt.] Explain your results.

2 PCA - Dimensionality Reduction

The original dataset contains $36 * 36 = 1296$ features. Therefore, we will try to reduce the dimension by using PCA.

a) [1pt.] Perform PCA decomposition, initially using **all principal components**. Before performing PCA you usually need to **mean-center** the

data, see here why, which means you have to calculate the mean of each variable (column) and subtract it from the respective column. However, many libraries perform this step implicitly, so consult the documentation of the library you are going to use (e.g. PCA of sklearn).

b) [2pts.] Plot the **CDF of the explained variance** as a function of the number of principal components.

c) [1pt.] **Choose a number of principal components** to use by arguing why your choice is reasonable as a trade-off between the number of components used and classification performance. Afterwards, **train a kNN classifier** (choose a k that gave you the best results in 1-c).) and report **train and test accuracy**.

d) [2pts.] For this part we will perform the following experiment: **First**, randomly **sample a part of your dataset** (using a fixed k and all features), of size ranging from 3,000 to 21,000 (the whole dataset) in increments of 3,000. **Fit a kNN classifier and plot the running time.**
Now, use a fixed k and all samples of your dataset, but **fit a kNN classifier using a varying number of Principal Components**, ranging from 50 to 750 in increments of 100. P**lot the running time on the same plot as above**.
Describe the plot. What seems to affect -as a trend- the fitting time more? Number of samples used for trying, or the dimensions of the data?

e) [1pt.] **Bonus point**: Plot the images of the 10 first Principal Components.