Ryan Slattery
CS541 - Artificial Intelligence
Homework #1: Random Projection for Search Engines
November 23, 2022

Goal:

This project is designed to compare different dimensions of the k-nearest neighbor classifier — specifically the values of k that could be used — and how the differing values of k affect the precision of classifying test points and the overall runtime of the program.

Methods:

The data from the MNIST dataset provided online was scraped and converted to fit into a Python list. The training set consisted of roughly 60,000 images and labels, and the testing set consisted of 10,000 images and labels. For this project, only 15,000 instances were used for the training data and 2,500 were used for the testing data.

Each feature vector was normalized using L2 normalization, with the equation provided below.

$$L_2 = \sqrt{x^T x}$$

Formula 1: L2 Normalization for a feature vector x

After normalizing the training and testing data, a distance matrix was generated, which measured the distance between the testing and training instances. Using this matrix, the training points with the minimum calculated distance would be considered the "nearest neighbors" to the given testing point. The number of nearest neighbors observed would be decided by a value k.

Once the k nearest neighbors were found for each testing instance, the indices of those neighbors would be used in another method which would compare the labels of those neighbors to that of the given test points. This would be used to calculate the precision value. If the label of a neighbor matched the label of the testing instance observed, the precision would increase by a value of $\frac{1}{k}$. This means if all k-neighbors matched the testing label, the precision at that given point would be 1, and if none matched the precision would be 0. The formula for calculating precision is provided below.

$$\text{precision@}k := \frac{\sum_{j=1}^{k} \mathbf{1}_{\{y_{i_j}=y\}}}{k}.$$

Formula 2: Precision

Precision was calculated for all testing instances in this project for different values of k. The summary of the results is provided in the next section.

Summary of Findings:

As one can see from the graph, the value of precision decreases as the number of neighbors (k) increases. The maximum precision achieved was roughly 0.94 when k = 1. This is because as k increases, the number of neighboring points observed near our test point increases. Points with different labels than our test point start to become more present with a greater number of neighboring points, and because of that the value of precision starts to decrease.

The rate of decrease becomes significantly larger as the difference in k grows as well. We see that when k is 1, 2, or 5, the difference in precision is rather small (only 0.01 or 0.02) while the difference in precision is much larger when k is 500 or 1000 (which has roughly a 0.12 difference).

Runtime was another issue with increasing the value of k, as the program would not run as fast with larger values of k. When k was 1, the program was executed in roughly 3 seconds, contrary to when k was 1000, where the program was executed in almost 40 minutes.

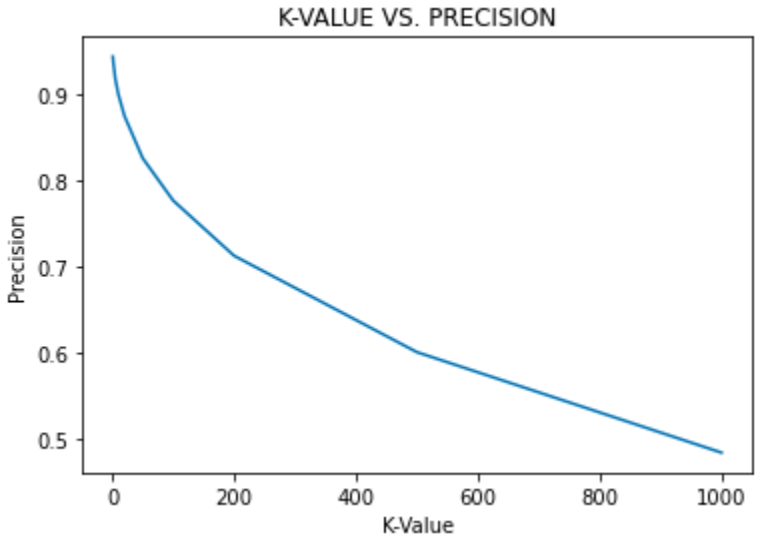

Figure 1: K-Value vs. Precision

```
[[1.07392717 1.1397808  1.22676781 ... 0.94561459 1.21339435 0.99995028]
 [1.1517042  1.14213475 1.31496549 ... 1.27392082 1.05339123 1.25102192]
 [1.1409795  1.18986695 1.36194237 ... 1.31037135 1.02489886 1.25231353]
 ...
 [1.01518341 0.99217924 1.1422349  ... 0.73609735 1.01121955 0.80144209]
 [1.20988819 1.27965944 0.83770956 ... 1.28908488 1.2896021  1.30563174]
 [1.1748962  1.2041558  0.99596357 ... 1.13069903 1.24284373 1.13266257]]
```

Figure 2: A redacted version of the distance matrix, showing the distances between each training and

testing point. Points were normalized using L2 normalization before computing the distance matrix.

```
Runtime for k = 1 is: 3.3646621704101562
Average Precision for k = 1 is: 0.9436

Runtime for k = 2 is: 5.183499097824097
Average Precision for k = 2 is: 0.936

Runtime for k = 5 is: 10.273832321166992
Average Precision for k = 5 is: 0.917999999999999

Runtime for k = 10 is: 18.188706874847412
Average Precision for k = 10 is: 0.9000800000000039

Runtime for k = 20 is: 33.9644410610199
Average Precision for k = 20 is: 0.8749000000000032

Runtime for k = 50 is: 83.64851498603821
Average Precision for k = 50 is: 0.8256960000000061

Runtime for k = 100 is: 171.78658890724182
Average Precision for k = 100 is: 0.7765240000000032

Runtime for k = 200 is: 336.93781089782715
Average Precision for k = 200 is: 0.7123819999999986

Runtime for k = 500 is: 841.1128120422363
Average Precision for k = 500 is: 0.6004760000000008

Runtime for k = 1000 is: 2429.4825043678284
Average Precision for k = 1000 is: 0.4837936000000005
```

Figure 3: Runtimes and Precision values for each value of k. Due to a period of time of my computer being idle and going to sleep, the runtime for k = 1000 is a little longer than expected.