ISYE 7406 - Homework 1

Matthew Bernstein

January 17, 2023

Introduction

The purpose of this assignment is to compare two different methods of supervised learning: Linear Regressions and K Nearest Neighboor classifiers. To do so, we used the *zipcode* dataset. This dataset is used for handwriting analysis by representing a 16x16 pixel image of a digit (1-9) as 256 individual greyscale values. We were particularly focused on the digits 2 and 7.

The problem being evaluated is the performance of the 2 different model types, as well as understanding how to properly tune a KNN classifier and the importance and effect of cross validation on model selection.

Exploratory Data Analysis

The dataset, after filtering for only the digits 2 and 7, contains 1376 observations. Each observation contains the intended digit, which is the response value, as well as 256 columns describing the greyscale value of each pixel in the 16x16 image. Figure 1 shows an example image representation of one of the observations. Of the 1376 observations, 731, or 53% of them are for the digit "2".

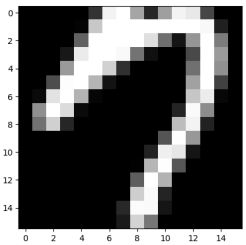


Figure 1: Example image reconstructed from data. Y = 7

Each greyscale value is a number between -1, meaning black, and 1, meaning white. Figure 2 shows a correlation heatmap between each of the pixels as well as the response digit. It

does appear that any single pixel has a strong correlation, white for positive and black for negative, with the response variable. Each does appear to be positively correlated with the pixels next to it as well as above and below it. There seems to be some highly negative correlation between the first 2 rows of pixels and pixels towards the last few rows.

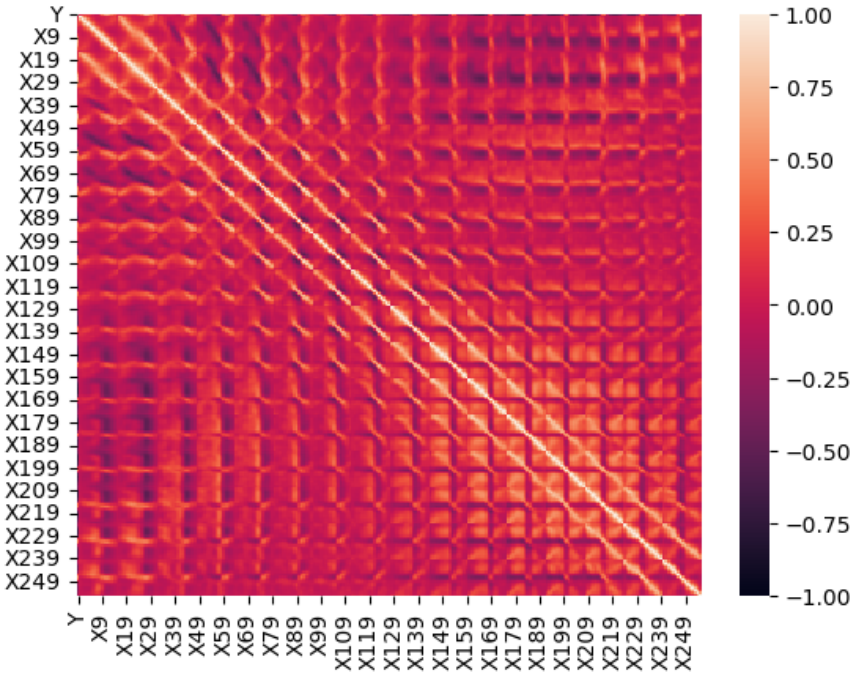


Figure 2: Correlation heatmap between all data columns

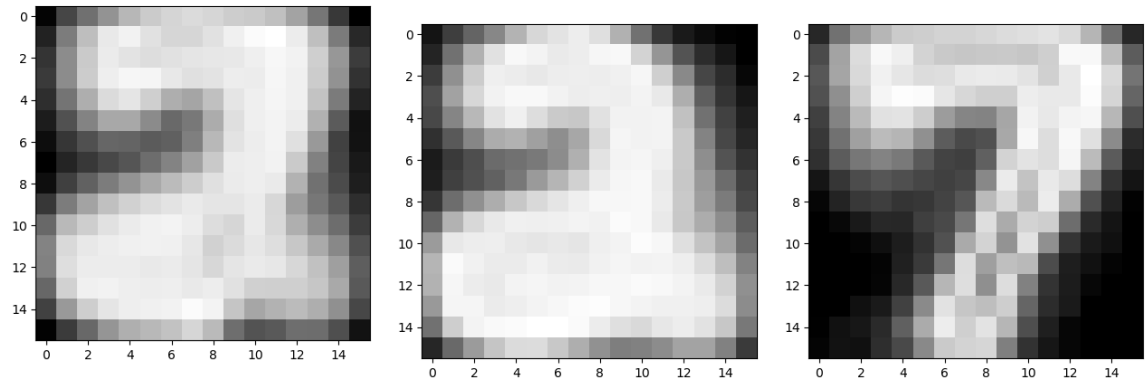
When looking at the average response for each image type in Figure 3 shows that a classifier should be fairly accurate given the different images. The average 7 closely resembles a 7 and while the average 2 is less clear, there are distinct regions that separate it from the average 7. Notably, the bottom left and right corners appear light on a 2, but dark in a 7.

Methodology

Two different models were used in this analysis. First a linear regression model was constructed based on the data. The resulting value was determined to be a "2" or a "7" depending on if the output of the model was above or below 4.5, the center point between 2 and 7. The 8 KNN classifiers were trained using $k = 1, 3, 5, 7, 9, 11, 13, 15$.

Training error was reported for each of the 9 models. They were then tested against the test data set, which had also been filtered for only "2" and "7". The test error for each model was then calculated.

Finally, a Monte Carlo cross validation was performed for each of the 9 models. In this, the



(a) Average of all observations

(b) Average "2"

(c) Average "7"

Figure 3: Average images for each response type

training and test data sets were combined and then re-split in the same proportions. Each of the 9 models were trained on the new training set and test error was recorded for the new test set. This procedure was performed 100 times and then average test error for each model was calculated.

Results

Model	Training Error (%)	Test Error (%)
Linear Regression	0.073	1.739
KNN, k = 1	0.000	1.739
KNN, k = 3	1.017	1.449
KNN, k = 5	1.236	1.449
KNN, k = 7	1.454	1.739
KNN, k = 9	1.599	1.739
KNN, k = 11	1.599	1.739
KNN, k = 13	1.744	2.029
KNN, k = 15	1.744	2.029

Table 1: Training and test errors for all models

Table 1 shows the training and test error results from the 9 tested models. Each model shows training error lower than test error, which is expected, however, they are quite close meaning each of the models is fairly robust. Going by test error, the $k = 3$ and $k = 5$ models performed the best with the linear regression and $k = 1, 7, 9, 11$ performing slighly worse.

Higher orders of k can cause problems on the borders of the data as the farther out data

could swing the internal "vote" towards a different classification. This is demonstrated by the data since the lower values of k out perform larger values.

Table 2 shows the Monte Carlo Cross Validation results. On a whole, error and variance are low for all methods. Linear Regression performs the best over all the different models as it has the lowest average error and the second lowest error variance. Suprisingly, KNN $k = 1$ performs the best of all KNN models. Error and variance geenrally rise as k increases with a large jump in average error at $k = 5$ and a large jump in error variance at $k = 7$.

Model	CV Test Error (%)	CV Test Variance
Linear Regression	1.176	2.6×10^{-5}
KNN, k = 1	1.258	2.5×10^{-5}
KNN, k = 3	1.310	2.8×10^{-5}
KNN, k = 5	1.528	4.0×10^{-5}
KNN, k = 7	1.673	4.5×10^{-5}
KNN, k = 9	1.742	4.8×10^{-5}
KNN, k = 11	1.855	5.2×10^{-5}
KNN, k = 13	1.947	5.1×10^{-5}
KNN, k = 15	1.986	5.6×10^{-5}

Table 2: Training and test errors for all models

Based on the results from the test set and cross validation, a KNN classifier with $k = 3$, would be the recommended model moving forward.

Conclusion

In conclusion, both Linear Regression models and KNN classifiers can perform quite well for performing classification tasks. Even though Linear Regression tends to be less accurate for discrete data, it peformed exceedingly well for this dataset. It would be interesting to see if it maintains its peformance accross the whole data set, and not just a filtered subsection.

For the KNN classifiers, it was shown that higher orders of k can have a lot more variance accross cross validation trials and can be less accurate. Therefore, it is recommended to keep the k parameter low to ensure that only the most relevant neighbors are considered when classifying new data.