Using K-Nearest Neighbors on the MNIST Dataset

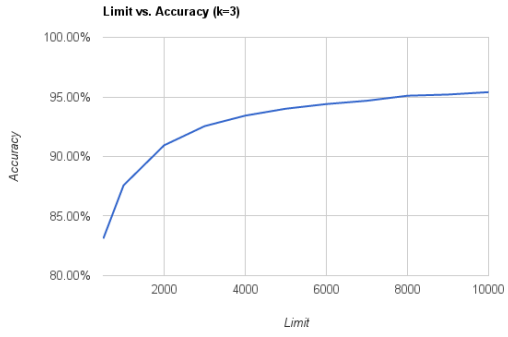
1. Introduction

In this assignment, I use a K-Nearest-Neighbors (KNN) model, implemented using scikit-learn and numpy, to classify the MNIST handwriting dataset. I then show how the role of the number of datapoints as well as K contribute to accuracy. Afterwards I analyze which numbers get confused with eachother most easily.

2. Analysis

After completing the model, I ran a series of trials comparing the number of training examples with the accuracy while holding k constant. Before running the test I hypothesized that as we gave the model more training examples we would have higher accuracy. By graphing the results it is clear that the accuracy is asymptotically approaching somewhere around 95-96 percent.

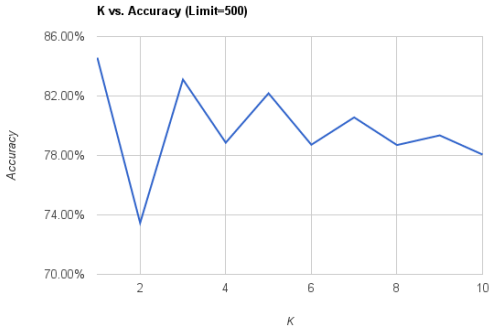
Figure 1. Graph of KNN Accuracy vs. Training example limit.



Next I plotted accuracy vs k. I decided to keep the limit set to 500 because at 500 the change in accuracy would be easier to see with change in k. The results were interesting; as k grew the accuracy sinusoidally approached somewhere around 79 percent.

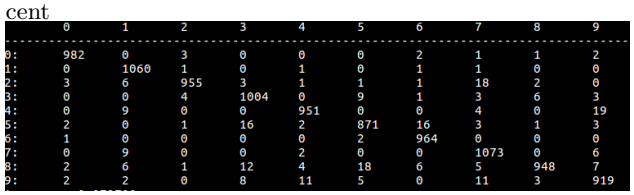
Another interesting metric to consider with the MNIST dataset is what numbers get confused with other numbers. It is easy to analyze using the confusion matrix printed at the end of the program. By analyzing this matrix, you can see that the numbers

Figure 2. Graph of KNN Accuracy vs. k.



5 and 8 get confused with other numbers most often. The number 5 mostly gets confused with the numbers 3 and 6 whereas 8 gets confused with 3 and 5. Another outlier is how the number four gets confused with 9, an occurance that happend 19 times in our test. All of these confusions make sense due to the numbers that are being confused looking somewhat similar. It would be weird if 1 got confused with 5 because they look nothing alike. The vectors representing 8 and 3 on the other hand must look very similar due to their curved nature.

Figure 3. Confusion Matrix with final accuracy of 97 percent



	0	1	2	3	4	5	6	7	8	9
0:	982	0	3	0	0	0	2	1	1	2
1:	0	1060	1	0	1	0	1	1	0	0
2:	3	6	955	3	1	1	1	18	2	0
3:	0	0	4	1004	0	9	1	3	6	3
4:	0	9	0	0	951	0	0	4	0	19
5:	2	0	1	16	2	871	16	3	1	3
6:	1	0	0	0	0	2	964	0	0	0
7:	0	9	0	0	2	0	0	1073	0	6
8:	2	6	1	12	4	18	6	5	948	7
9:	2	2	0	8	11	5	0	11	3	919

3. Conclusion

In this assignment I analyzed how accuracy was effected by comparing it to both the k value and the limit of training examples. I then analyzed what numbers were getting confused the most. The final accuracy was achieved was slightly above 97 percent which makes it acceptable in the research context but still less accurate than methods like deep learning.