CIS 520, Machine Learning, Fall 2018: Assignment 2

Simran Arora, Shubhankar Patankar

September 23, 2018

Problem 2

1. The following plots the K-NN N-fold error on the training set for, $N = \{3, 5, 9, 15\}$, and the test error for K-NN, with $K = 1$, and for Kernel Regression with $\sigma = 1$. For the noisy data, an increase in the number of folds, generally gives a decrease in the amount of error. This makes sense because with more folds, our test set is smaller and the amount of data being used in training the model is thus higher - this means that we can reduce overfitting while training the model because there are more data points to train on. For the original data, we observe that the $N - fold$ error shows a less dramatic decrease in error as the number of folds increases, which makes sense because the original data is used for training as well.

   For the noisy data, we see the trend that the N-fold error is larger than the test error. For the N-fold error, we take our original data set, randomly split it into folds, leave out a single fold one at a time, while training on the remaining folds. Then we use the left out fold to test our model and compute error, averaging as we do this for each fold. In the test error, we split the data such that 450 points are used for training and 150 randomly selected points are used for testing. It is possible given the random split, that we produce a slightly lower error. Note that the overall N-fold and test errors for noisy data are relatively close and within 0.02 of each other.

   For both K-NN and Kernel Regression, we note that there is still some error due to inherrant/irremovable data noise. The baseline error is much lower on the original data because there is lower noise in this dataset. For the original data, the error is between 0.035 and 0.04 while for the noisy data, error is near 0.25. There is more inherrant/irremovable error due to the higher noise level in the noisy set.

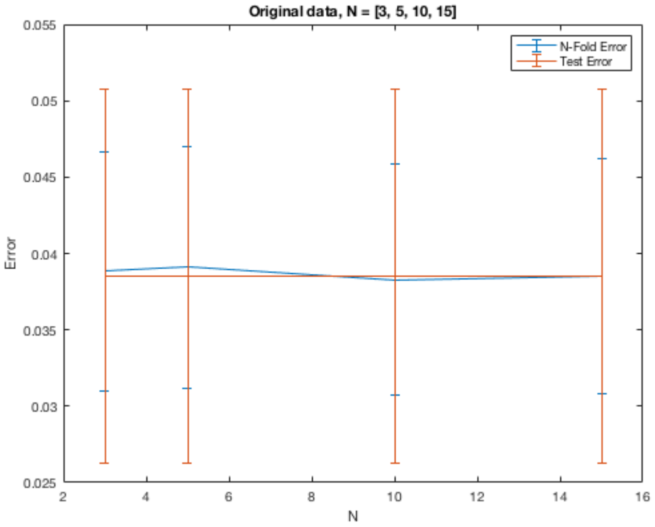

Figure 1: K-NN N-fold Error on Original Set

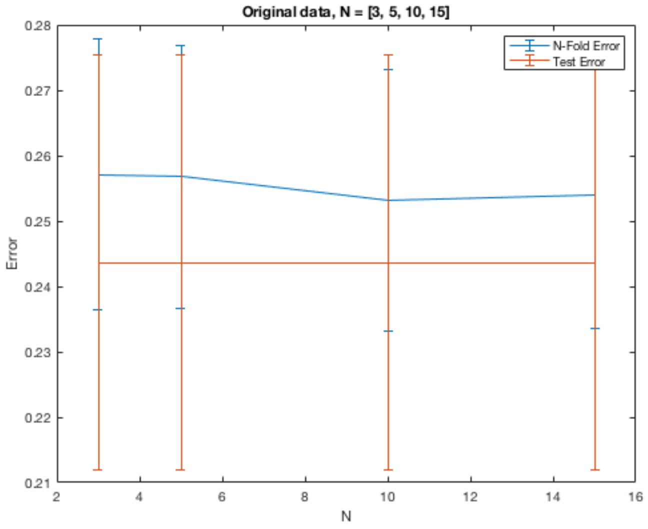

Figure 2: K-NN N-fold Error on Noisy Set

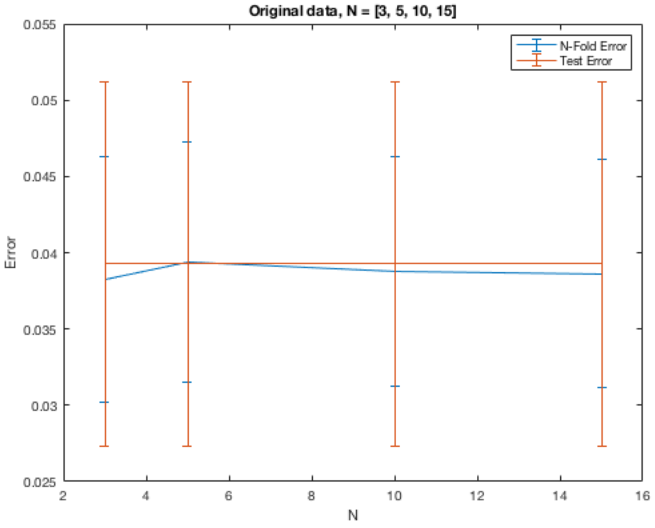

Figure 3: Kernel Regression N-fold Error on Original Set

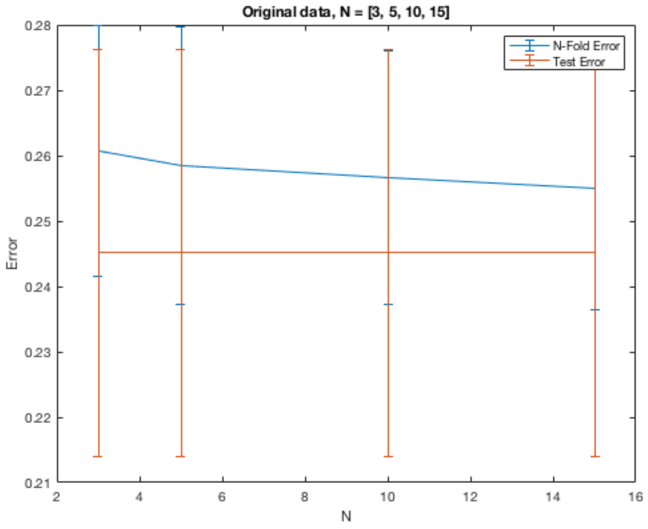

Figure 4: Kernel Regression N-fold Error on Noisy Set

2. The following figures show the 10-fold cross validation error on the training set and the test error for K-NN with $K \in \{1, 3, 4, 5, 9, 14, 22, 35\}$ and $\sigma \in \{1, 3, 5, 7, 9, 11\}$. On the original data, we can see that the best σ value is 3 and the best K value is 9. On the noisy data, we can see that the best σ value is 9 and the best K value is 35. These values minimize the testing error on the noisy test set. The best values are $\sigma = 9$ and $K = 35$ to minimize test set error. In the figures we notice that increasing the complexity (lower K and σ) yields increases in error because of overfitting. For low complexity (high K and σ) we also see a higher error because our model is underfitting. On the kernel regression plots, we can see both these regions where the complexity is too high or low. On the K-NN plots, we see the error decrease as K increases and complexity decreases, however, we don't see the region as yet, where the error starts increasing again (high$K$) and underfitting the data because the complexity is too low.

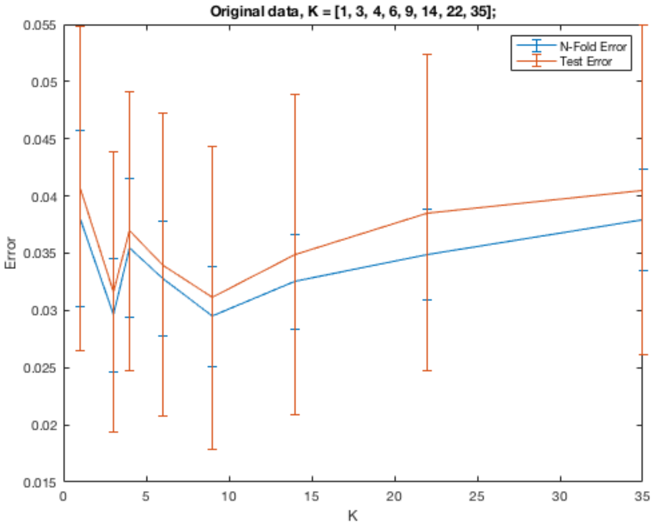

Figure 5: K-NN 10-fold Cross Validation Error on Original Set with Various K Values

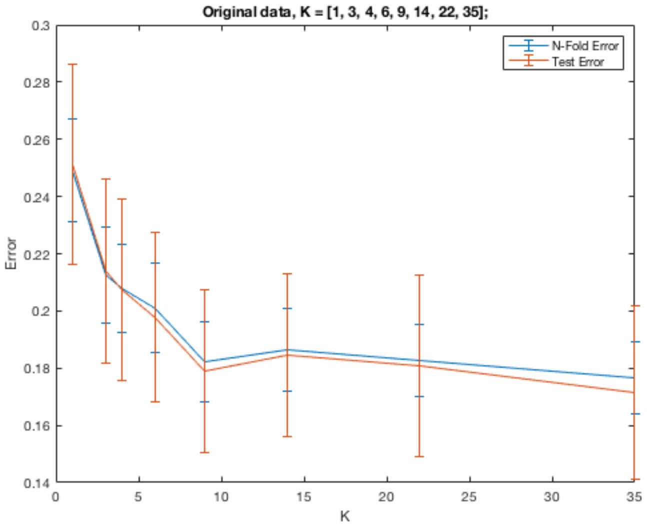

Figure 6: K-NN 10-fold Cross Validation Error on Noisy Set with Various K Values

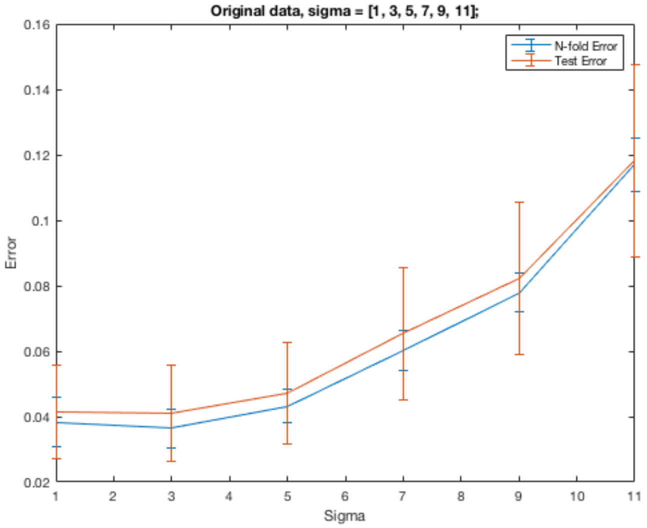

Figure 7: Kernel Regression 10-fold Cross Validation Error on Original Set with Various Sigma Values

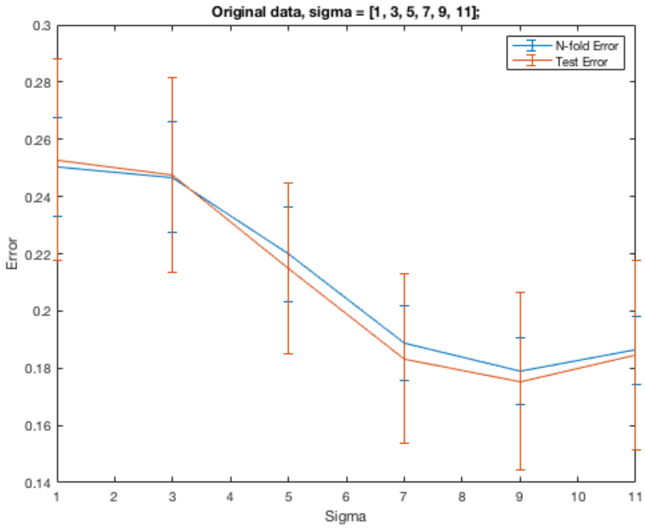

Figure 8: Kernel Regression 10-fold Cross Validation Error on Noisy Set with Various Sigma Values

Problem 3

1. The following figures show the error per iteration - the evolution of the zero-one loss - over the training data as the gradient ascent proceeds, for gradient ascent with decay. We noticed improvement over the gradient ascent with a constant step size because the number of iterations required to reduce the error is far less with decay, by almost half as many iterations. This is likely because the decay reaches a tradeoff between looking for the maximum, and converging on the maximum. In decay, we take large steps at first to look for the maximum, and then smaller steps as we converge. In the constant-step size version, even as we near the maximum, we still take big steps, which cause us to jump around closer and farther from our desired end point. This is very clear on both the original and noisy data. Note that the original data asymptotes at a lower error value, which makes sense because there is less irremovable/inherant noise in the dataset.

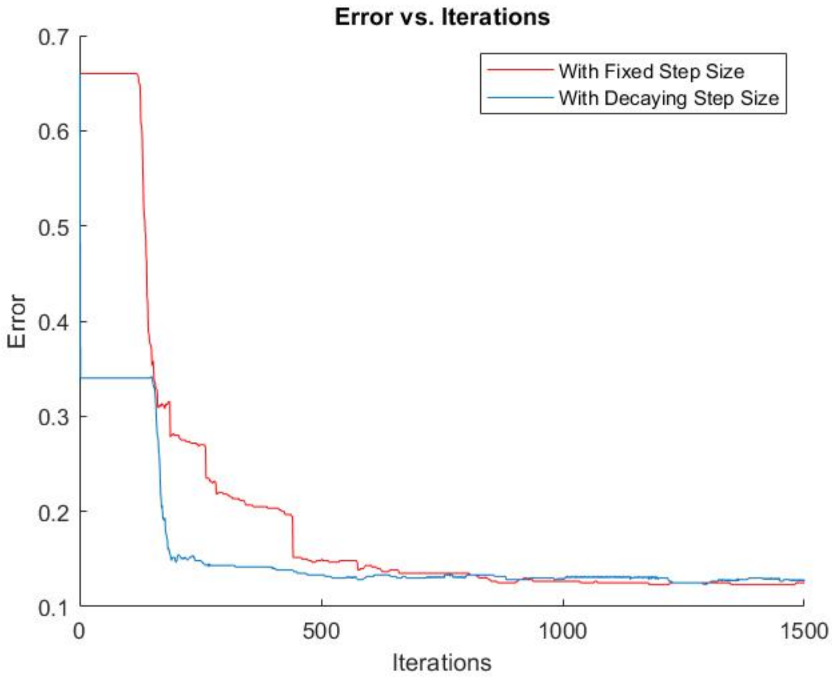

Figure 9: Error Per Iteration on Original Data

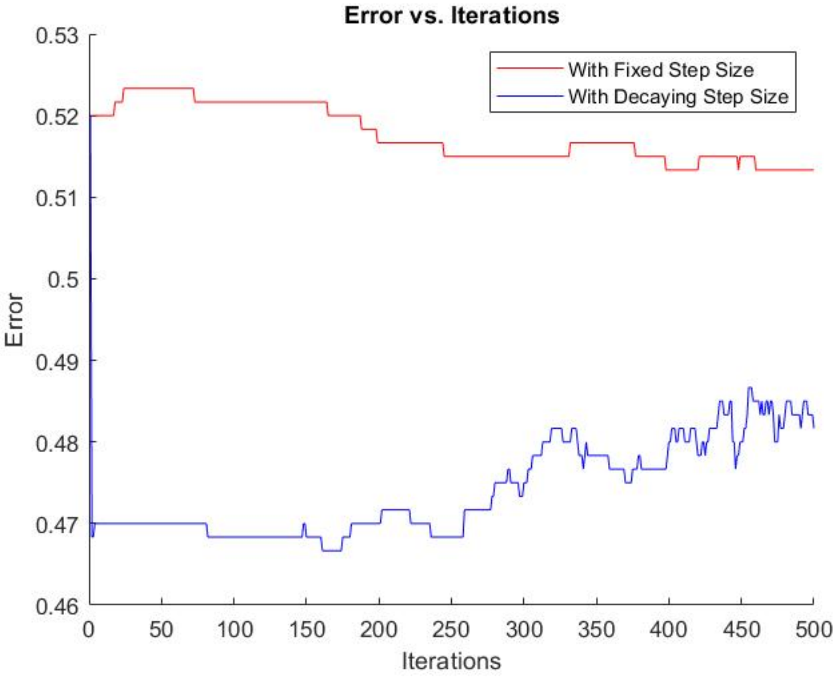

Figure 10: Error Per Iteration on Noisy Data

2. One extra feature was added to the data that is always set to 1. The following figures show the error per iteration - the evolution of the zero-one loss - over the training data as the gradient ascent proceeds, for gradient ascent with decay. Adding the new feature corresponds to adding a constant to the scores,

i.e. w, x + c. We notice that the number of iterations of gradient ascent required to reduce the error decreases without the extra feature, and the plot without the constant feature asymptotes at a much higher error than the plot with the extra feature.

To understand why this improvement likely occurred, we first note that error is a combination of the bias and variance of the model. As additional features are added in gradient descent, the complexity increases. With more complexity, there is less bias and the training error always decreases. We see that the bias with the constant feature is reduced, as the error aysmptotes at a lower value. When we observe that changing the label of a single feature, or shifting the model by a single constant changes the learned prediction model, we can get a measure of the learning algorithm's sensitivity and stability. We chose to analyze the effect of the extra feature using the fixed gradient descent. This helps explain why on the noisy data, we see lots of noise in the tail, because this gradient method jumps around a lot.

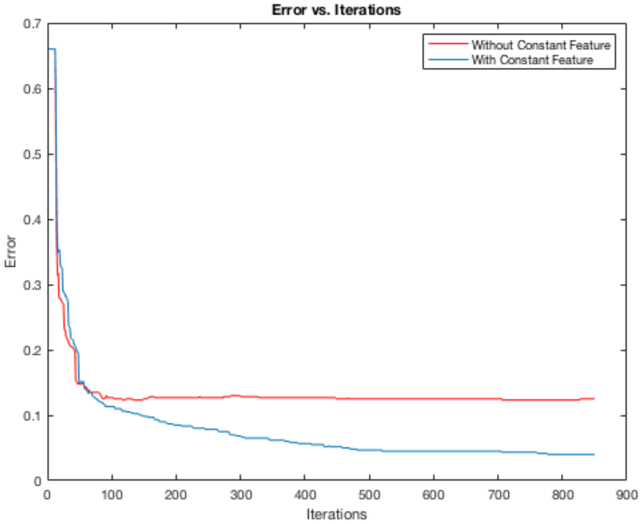

Figure 11: Error Per Iteration on Original Data with Extra Feature

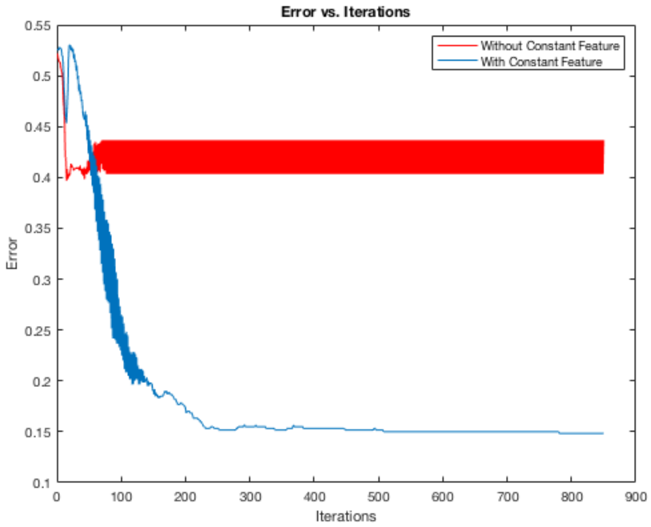

Figure 12: Error Per Iteration on Noisy Data with Extra Feature

3. The following figures show the result from logistic regression. We computed both the N-fold error on the training set, for $N = \{3, 5, 9, 15\}$ and the test error. We notice that the $N-fold$ error is generally constant as the number of folds increases. Note that the difference in error is small < 0.005 between the N-folds and test error at all points. It makes sense that the test error his higher than the N-folds error overall because N-folds uses an average error while leaving out folds one at a time, whereas the test error randomly select 150 points to test with and produces error from that. We do not see this for the original data though, which is possible due to the random nature of the selected folds. Thus, the N-folds adds the extra cross-validation. The errors for the original data asymptote at a lower error value (near $error = 0.13$) than the noisy datas (near $error = 0.425$). The inherant/irremovable noise in the noisy dataset contributes to this higher baseline error.

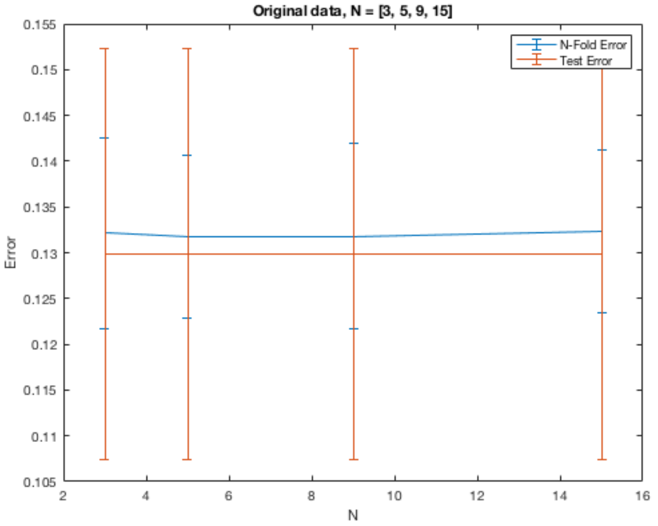

Figure 13: Logistic Regression Cross-Validation on Original Data

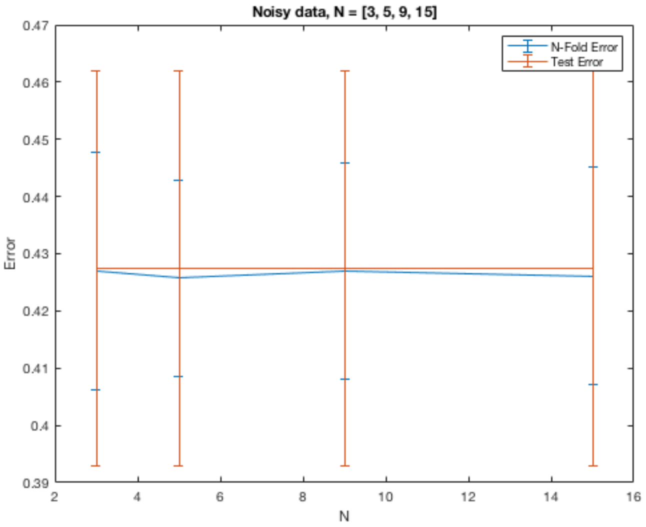

Figure 14: Logistic Regression Cross-Validation on Noisy Data