CIS 520, Machine Learning, Fall 2018: Assignment 2
Due: Sunday, September 23rd, 11:59pm (via turnin)

Wentao He

September 23, 2018

Collaborators: Ruo Jia

1 Cross Validation

For the code part, please see `make_xval_partition.m` in **turnin**.

2 K if for Kernel Width

For the code part, please see `k_nearest_neighbours.m`, `kernel_regression.m`, `knn_xval_error.m` and `kernreg_xval_error.m`.

(i) For both the original data and the noisy, compute both the N-fold error on the training set, for $N = \{3, 5, 9, 15\}$, and the test error for K-NN, and Kernel Regression with $K = 1$ and $\sigma = 1$ respectively.

(a) Figure on original data with KNN.

Please see Figure 1.

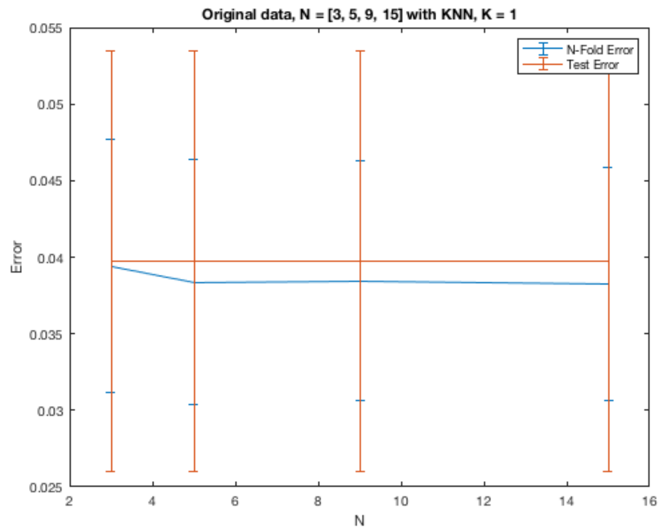

Figure 1: Figure on original data with KNN.

(b) Figure on noisy data with KNN.

Please see Figure 2.

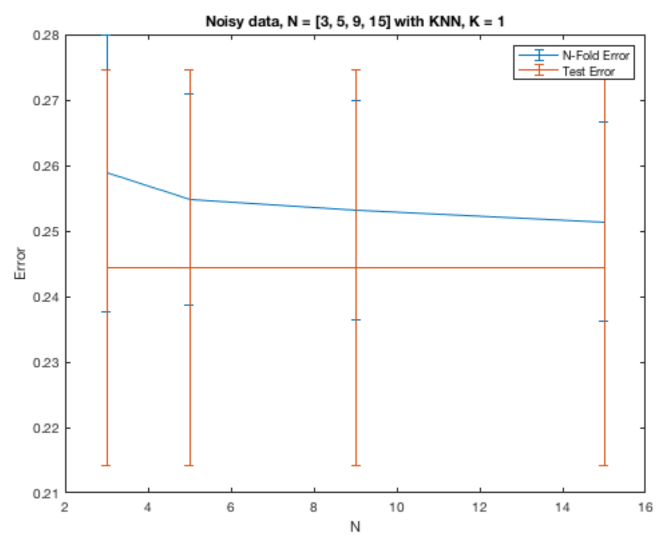

Figure 2: Figure on noisy data with KNN.

(c) Figure on original data with Kernel Regression.

Please see Figure 3.

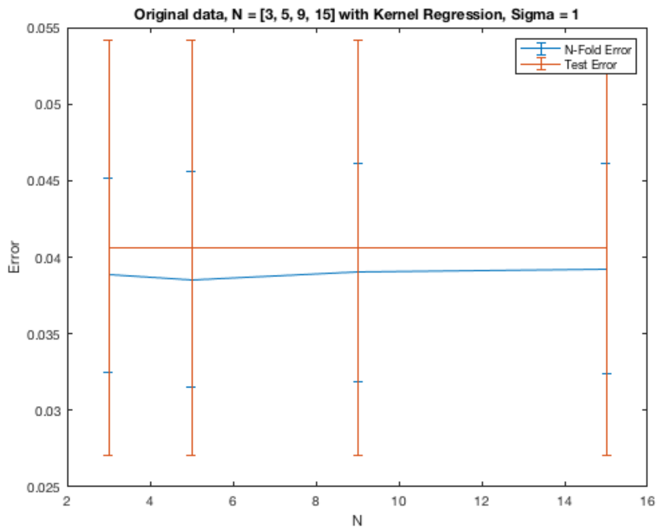

Figure 3: Figure on original data with Kernel Regression.

(d) Figure on noisy data with Kernel Regression.

Please see Figure 4.

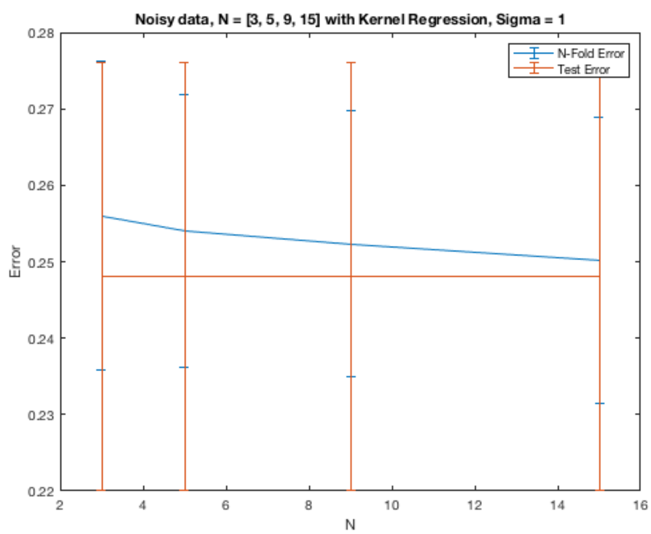

Figure 4: Figure on noisy data with Kernel Regression.

Based on these charts, the trend is that test error always stays the same while N-Fold error converges as the number of iterations increases.

(ii) For both the original data and the noisy, compute both the 10-fold cross validation error on the training set and the test error for K-NN with $K \in \{1, 3, 4, 6, 9, 14, 22, 35\}$ and for Kernel Regression with $\sigma \in \{1, 3, 5, 7, 9, 11\}$.

(a) Figure on original data with KNN.

Please see Figure 5.

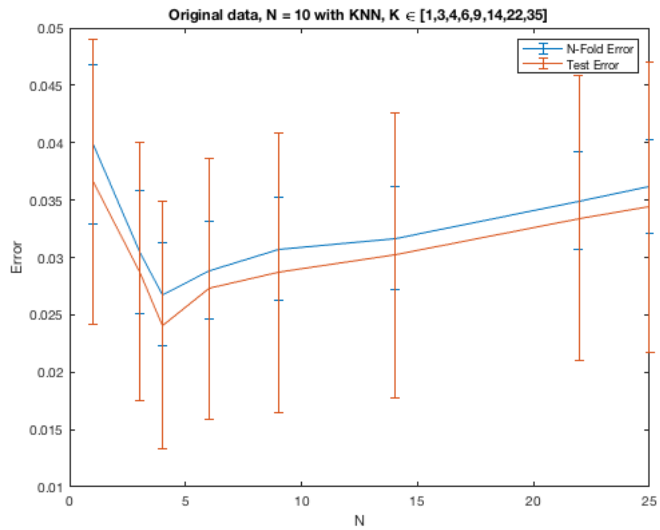

Figure 5: Figure on original data with KNN.

(b) Figure on noisy data with KNN.

Please see Figure 6.

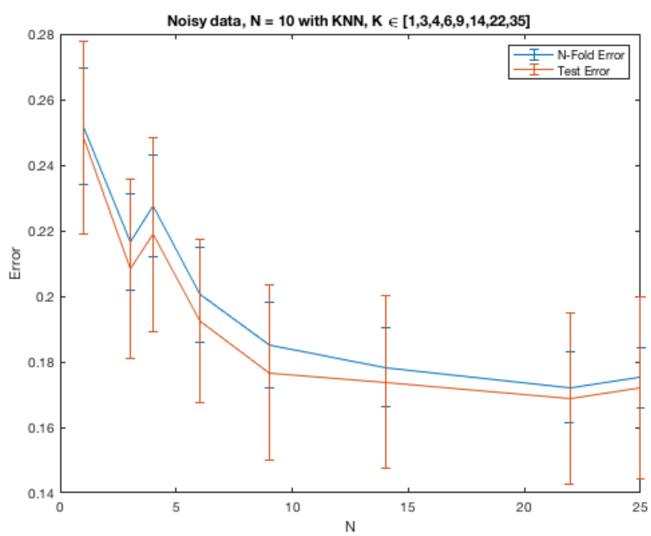

Figure 6: Figure on noisy data with KNN.

(c) Figure on original data with Kernel Regression.

Please see Figure 7.

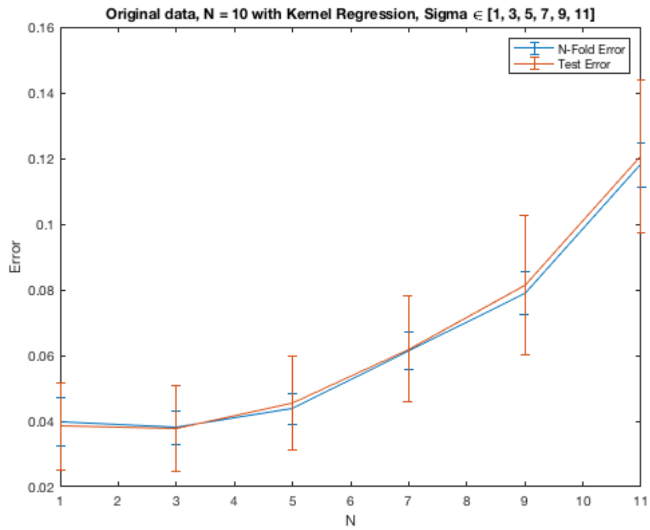

Figure 7: Figure on original data with Kernel Regression.

(d) Figure on noisy data with Kernel Regression.

Please see Figure 8.

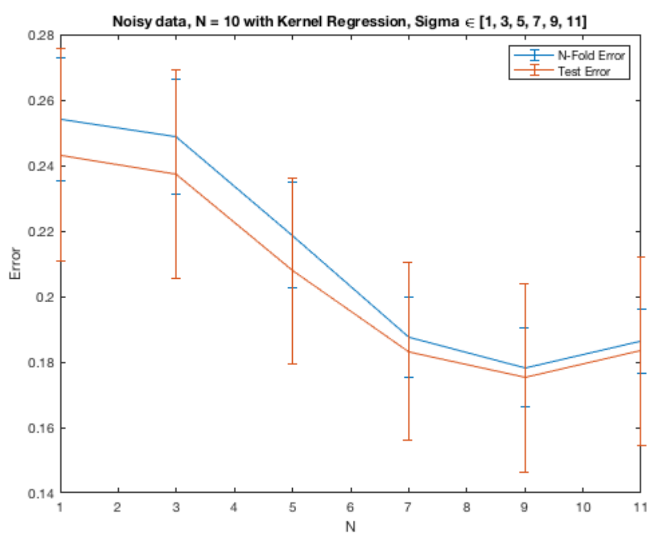

Figure 8: Figure on noisy data with Kernel Regression.

Based on these charts, the best K is 4 for original data, 3 for noisy data. The best σ is 3 for original data, 9 for noisy data.

3 Logistic Regression

For the code part, please see `gradient_ascent_fixed.m`, `gradient_ascent_decay.m`, `logistic_regression.m` and `logistic_xval_error.m`.

(a) Implement gradient ascent with step size that decays over time.

  i. Figure on original data.

  Please see Figure 9.

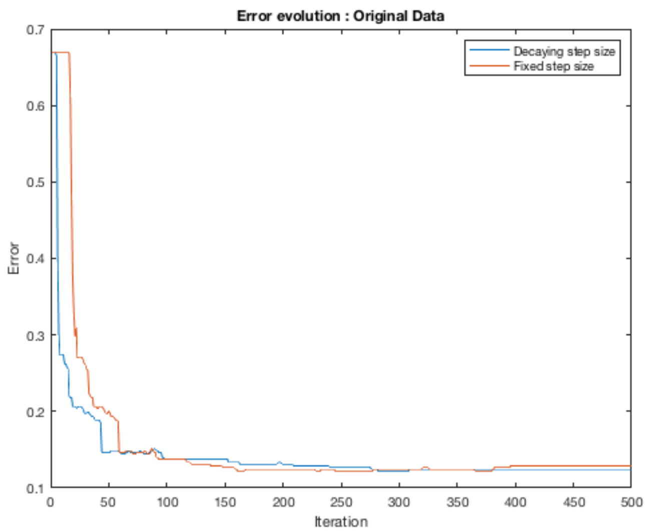

Figure 9: Figure on original data.

  ii. Figure on noisy data.

  Please see Figure 10.

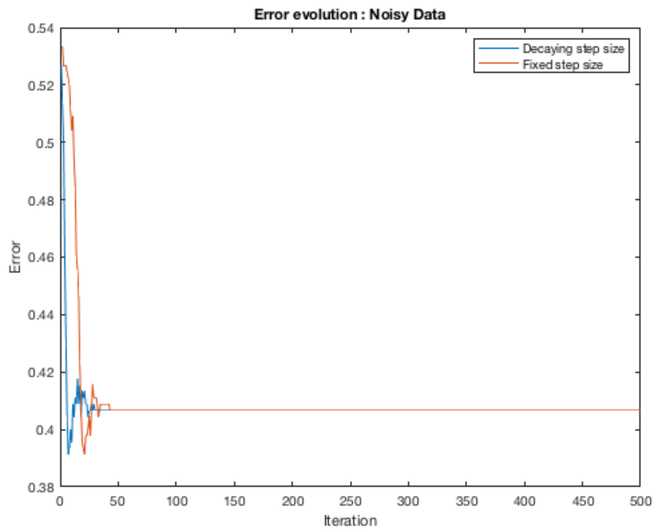

Figure 10: Figure on noisy data.

There is an improvement because with both step size being 0.0001, the error gradient ascent with step size that decays over time will continue to decrease if we run more steps. Gradient ascent with a step size that decays over time will converge to the optimal value faster than gradient ascent with a step size that is small enough.

(b) Add an extra feature to your data that is always set to 1, and perform gradient ascent on this new data.

i. Figure on original data.

Please see Figure 11.

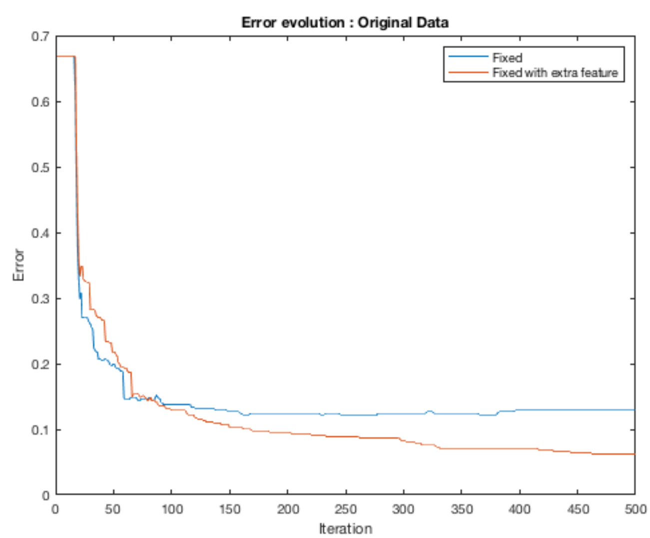

Figure 11: Figure on original data.

ii. Figure on noisy data.

Please see Figure 12.

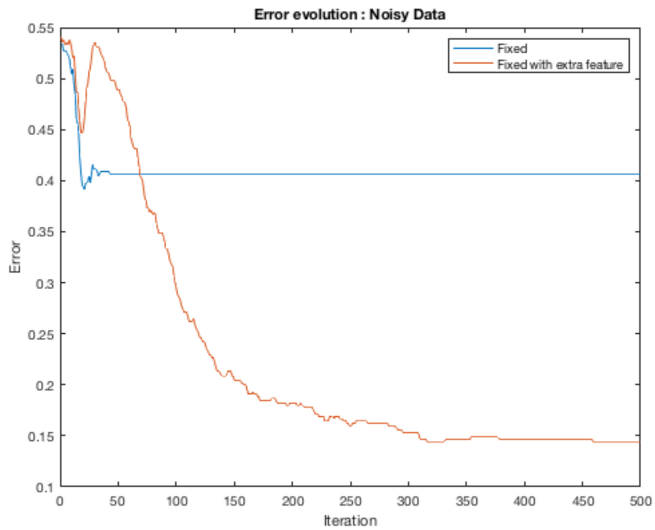

Figure 12: Figure on noisy data.

There is an improvement because with this extra feature added, gradient ascent now can converge to a lower error value compared to when it was run on the old data. By adding a constant to the scores, it allows the gradient ascent to shift in the direction that will fit the optimal value better.

(c) Implement `logistic_regression.m` and `logistic_xval_error.m`. For both the original data and the noisy, compute both the N-fold error on the training set, for $N = \{3, 5, 9, 15\}$, and the test error.

i. Figure on original data.

Please see Figure 13.

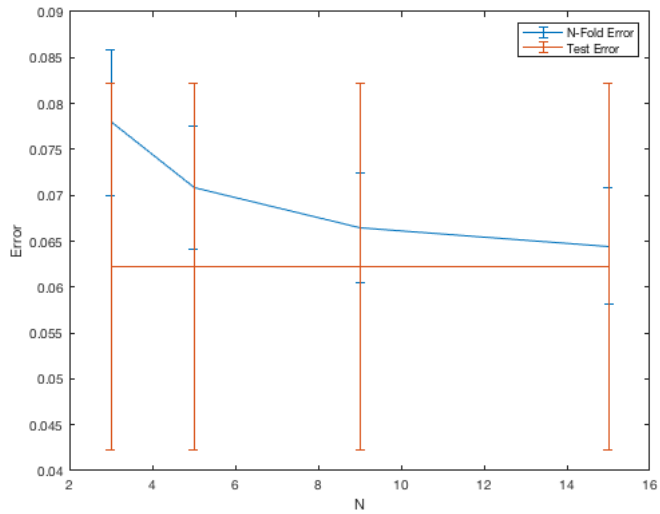

Figure 13: Figure on original data.

ii. Figure on noisy data.

Please see Figure 14.

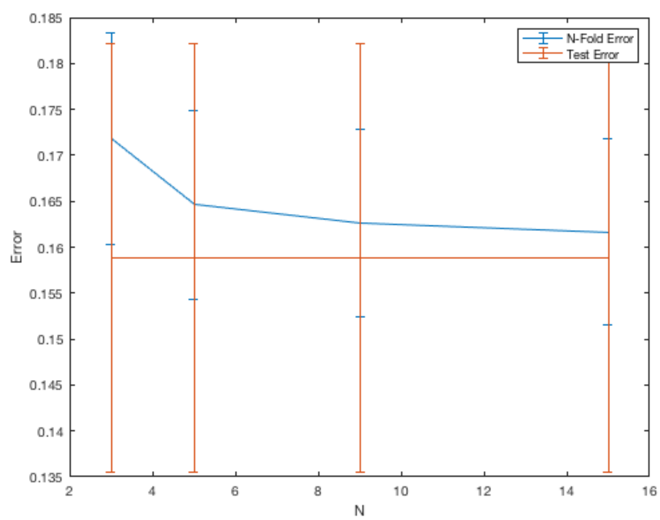

Figure 14: Figure on noisy data.