Project 1: Kernel Prediction and Cross-Validation

Isaiah Chen

February 25, 2019

Question 1

(1) For each pair $(N,\ h)$, the functions $(U,\ f(U))$, $(U,\ \hat{f}_h(U))$, and the sample points $(x_k,\ y_k)$, where $k = 1,\ ...,\ N$, are plotted below in Figures 1 and 2.

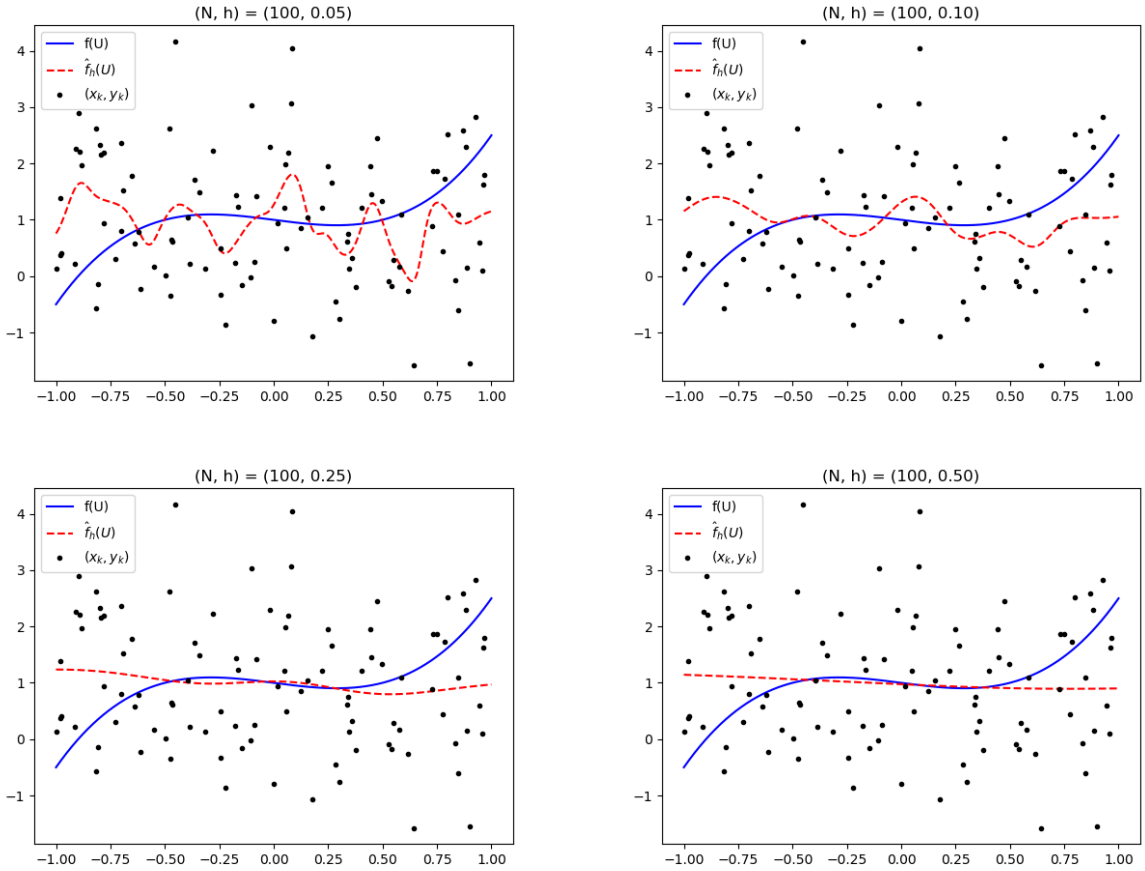

Figure 1: Kernel regression estimators for $N = 100$

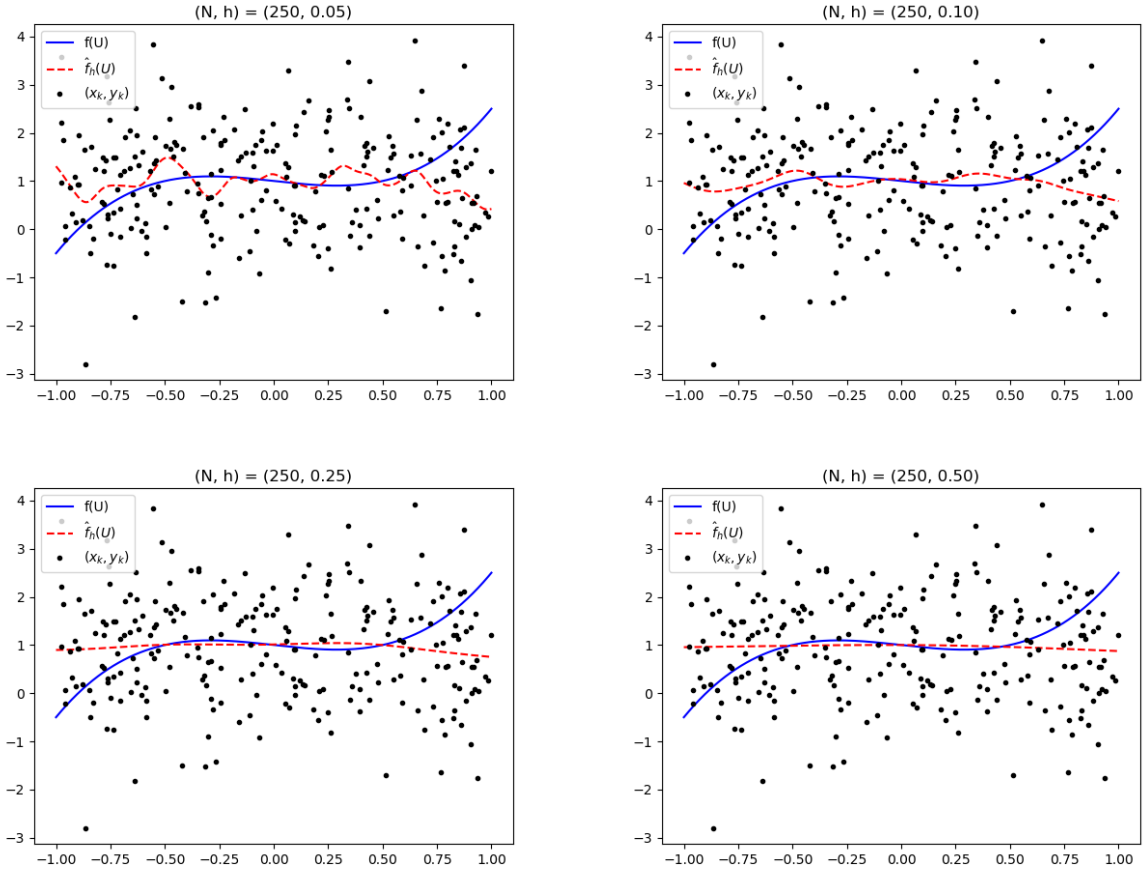

Figure 2: Kernel regression estimators for $N = 250$

As the kernel bandwidth increases, the variability of the calculated kernel regression estimator decreases. For each case, several error metrics were used to judge the accuracy of the kernel regression estimator. Values of e_1, e_2, and e_3 as are shown below in Table 1, alongside the error returned by 10-fold cross-validation. As the kernel bandwidth increases, e_1, e_2, and e_3 values tend to increase and the cross-validation error tends to decrease.

N	h	e_1	e_2	e_3	ε_{cv}
100	0.05	1.0397	0.0148	1.0572	1.4899
100	0.10	1.2083	0.0170	1.0693	1.4131
100	0.25	1.3111	0.0572	1.1092	1.4257
100	0.50	1.3304	0.1256	1.1724	1.3898
250	0.05	1.1658	0.0148	1.0572	1.3342
250	0.10	1.2150	0.0170	1.0693	1.2810
250	0.25	1.2431	0.0572	1.1092	1.2893
250	0.50	1.2519	0.1256	1.1724	1.2858

Table 1: Estimates of error for the kernel regression estimator

(2) $E((Y - f_h(X))^2 \geq 1$, because the given standard deviation of Y is 1. If the deviation of Y was larger, the variability of the data would increase and the corresponding kernel regression estimator would also increase, resulting in a larger expected error. A similar trend can be inferred about decreasing the standard deviation of Y and having a smaller error. Of the three error metrics evaluated in Table 1, e_3 estimates this quantity the best because it calculates the error between the testing data and the kernel regression estimator. The testing data is more representative of the entire data set and the training data is only a selected subset of the generated random variable Y.

Question 2

For values of h uniformly spaced over the interval [0.01, 1], the 10-fold cross-validation error $\varepsilon_{cv}(h)$ is calculated and shown below in Figure 3. The cross-validation error is minimized for an h_0 value of 0.9336 ($\varepsilon_{cv}(h_0) = 1.2179$). Once $\hat{f}_{h_0}$ has been retrained using the whole training set, the error evaluated using the test set is calculated to be 1.1374. However, it is important to note that the data appear very noisy and the optimal kernel bandwidth may fluctuate between different runs of the program.

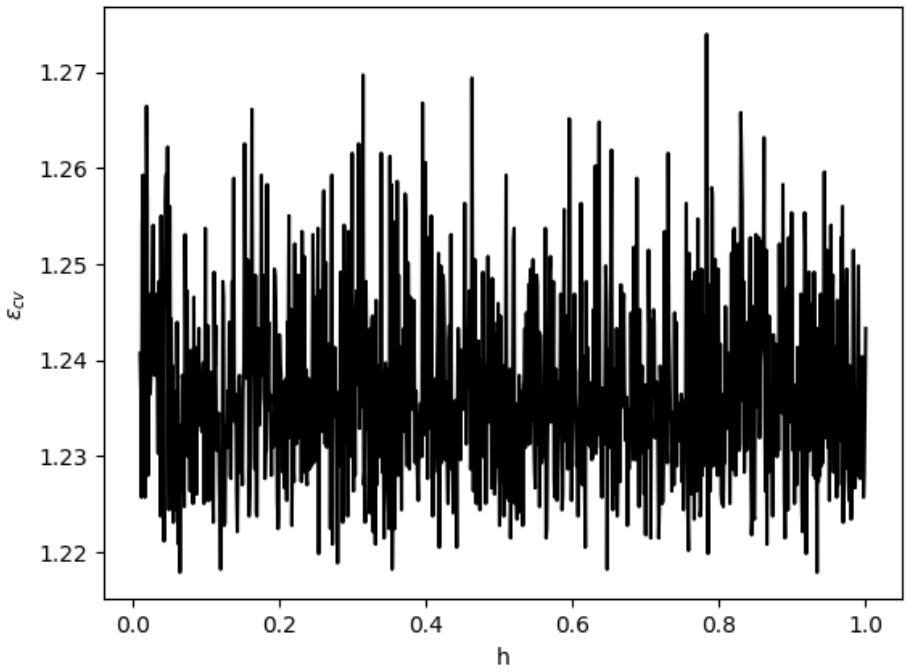

Figure 3: Cross-validation error as a function of h for the kernel regression estimator

Question 3

(1) For values of h uniformly spaced over the interval [0.01, 1], the 10-fold cross-validation error $\varepsilon_{cv}(h)$ is calculated for the kernel classification estimator and is shown below in Figure 4. The cross-validation error is found to be independent of the kernel bandwidth. The misclassification error of the classifier evaluated on both the training set and the testing set is 0 in both cases. However, these conclusions could be incorrect, due to possible programming errors that result in incorrect calculations of the kernel classification estimator.

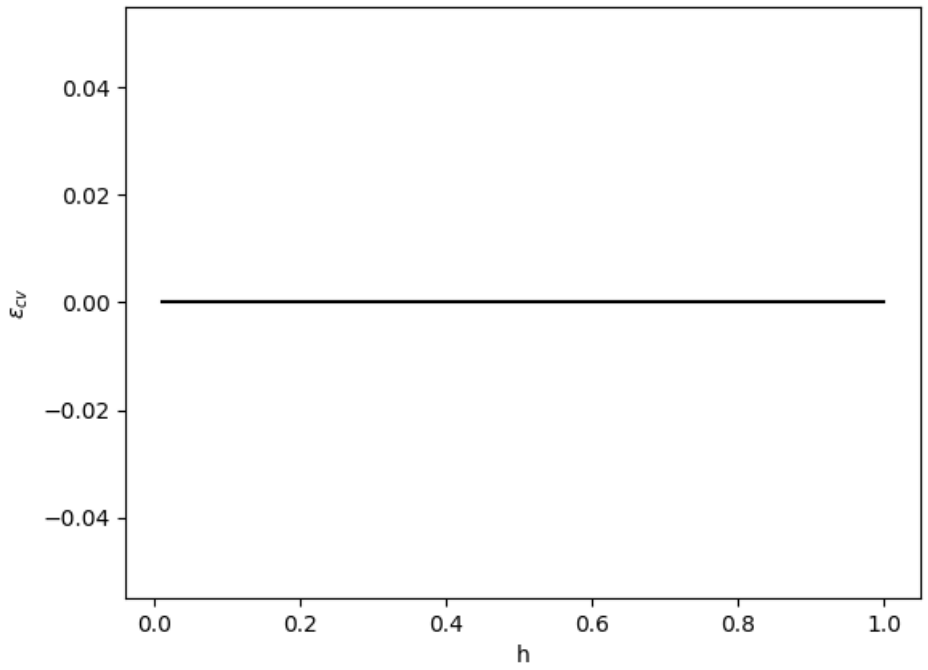

Figure 4: Modified cross-validation error as a function of h for the kernel classification estimator

(2) If we modify the cross-validation portion of the previous question and repeat the same procedure for finding an optimal value of h, the optimal value can be found at (0.9802, 10.6695), as shown below in Figure 5. As the kernel bandwidth increases, the error decreases and converges to minimum value. Similar to the previous question, the misclassification error of the classifier evaluated on both the training set and the testing set is 0 in both cases. Again, this could be due to possible programming errors.

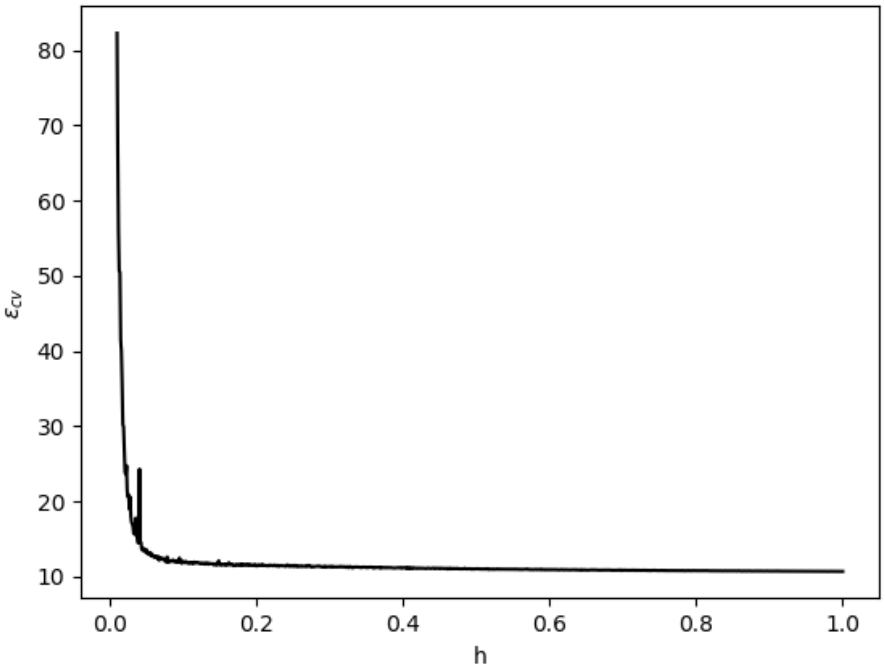

Figure 5: Cross-validation error as a function of h for the kernel classification estimator

*Note: The files submitted for this assignment also include the 3 programs that were used to generate figures and important values. The programs were written using Python 2.7 and use modules that are included in Anaconda.