CIS 520, Machine Learning, Fall 2021
Homework 3
Due: Sunday, Oct 3rd, 11:59pm
Submit to Gradescope

Yixuan Meng, Zhouyang Fang

1 Regularization Penalties

1. $\hat{w} = x^T y/(x^T x) = \begin{pmatrix} 0.889 \\ -0.826 \\ 4.190 \end{pmatrix}$

2. $\hat{w} = (x^T x + \lambda I)^{-1} x^T y = \begin{pmatrix} 0.841 \\ -0.816 \\ 4.056 \end{pmatrix}$

3. $\hat{w} = \begin{pmatrix} 1.004 \\ -1.078 \\ 4.075 \end{pmatrix}$

4. The combinations, weights and errors are presented in Table. 1. The value of w is $\begin{pmatrix} 0.889 \\ -0.826 \\ 4.190 \end{pmatrix}$

Combination cases	weights	Errors
[0, 0, 0]	[0, 0, 0]	4266.6
[1 0 0]	[1.16, 0, 0]	4225.2
[0 1 0]	[0, -1.503, 0]	4160.1
[0 0 1]	[0, 0, 4.293]	3171.1
[1 1 0]	[0.903, -1.393, 0]	4136.0
[1 0 1]	[1.039, 0, 4.278]	3138.1
[0 1 1]	[0, -0.934, 4.192]	3131.1
[1 1 1]	[0.889, -0.826, 4.19]	3107.8

Table 1: 8 cases for 3 unknown w_i

5. The first estimate of $\hat{w}$ without regularization penalty is for Ordinary Least Square method. With L2 regularization, each w_i in $\hat{w}$ is shrunk a little, and the largest element shrunk most. With L1 regularization, weights are shrunk less because the value of lambda is smaller. With L0 regularization, the optimal weights is the same as no regularization, because it minimizes the error (error reduced by adjusting weights is smaller than the increase in the sum of squared error).

6. (a) The value of the ratio is 0.006130.

(b) i. When going from N to 2N samples, I expect the error to increase, Sum of squared errors for linear regression **do not** directly depend on the number of training samples.
   ii. If double the number of training samples, I expect $||\hat{w}_{MLE}||_2^2$ does not change. It does not directly depend on the number of training samples when N is large enough because it would stop the shrinkage.

(c) lambda = 4

2 Feature Selection

1. Streamwise regression.

   (a) $Err_0 = 93$, $Err_1 = 26.53333$, $Err_2 = 24.6$, $Err_3 = 0.6$

$$w_1 = [3.33333333], w_2 = \begin{bmatrix} 0.4 \\ 1.6 \end{bmatrix}, w_3 = \begin{bmatrix} 157.85714286 \\ -64.71428571 \\ -15.71428571 \end{bmatrix}$$

   All three features are selected.

   (b) Only x_2 feature is selected.

   (c) The order of adding features really matters in the streamwise regression.

2. Stepwise regression.

   (a) $Err_{old} = 93$

   (b) $Err_{x_1} = 26.533333333333333, Err_{x_2} = 24.438095238095237, Err_{x_3} = 61.13054954565124$.

   (c) Add x_2. The updated $Err_{old} = 24.438095238095237$.

   (d) Halt after adding x_2, the error doesn't shrink after that.

   (e) Just x_2.

3. Findings: Pros: doesn't rely on the order of the features. Cons: (1)increased calculation burden, especially with large amount of features. (2) Cannot give best fit as streamwise regression when order is perfect like 2.1(a).

3 Kernel Regression

1. Build the model. (auto-graded only)

2. Analysis of the model.

   Figures:

Figure 1: $\sigma = 0.01$

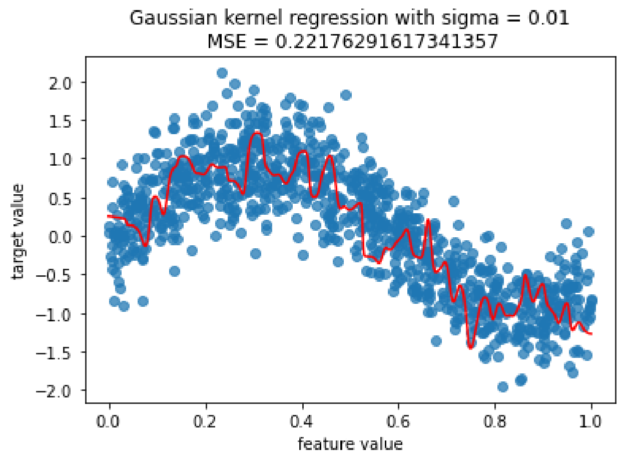


Figure 2: $\sigma = 0.05$

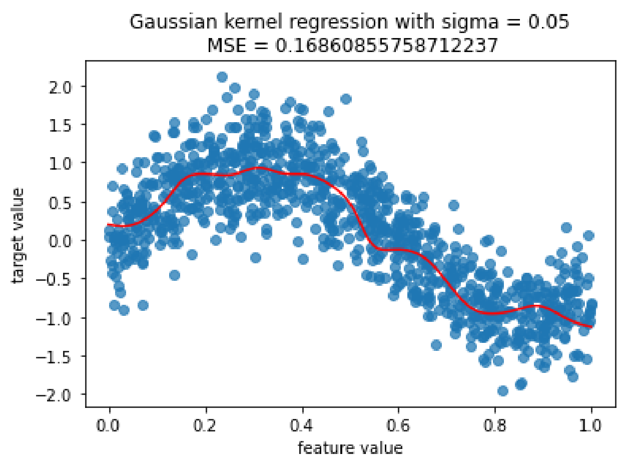


Figure 3: $\sigma = 0.1$

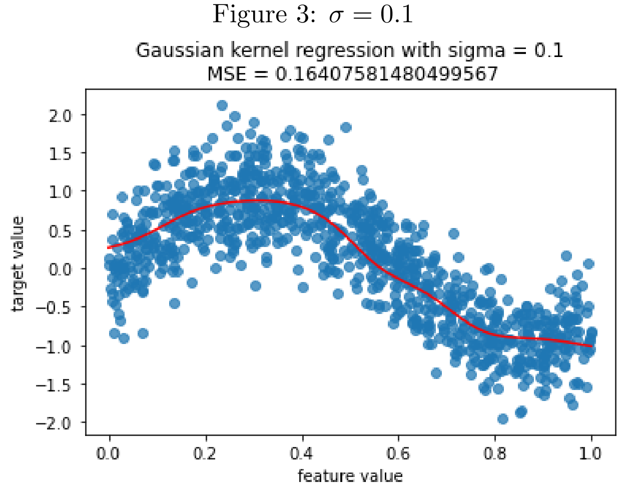


Figure 4: $\sigma = 0.15$

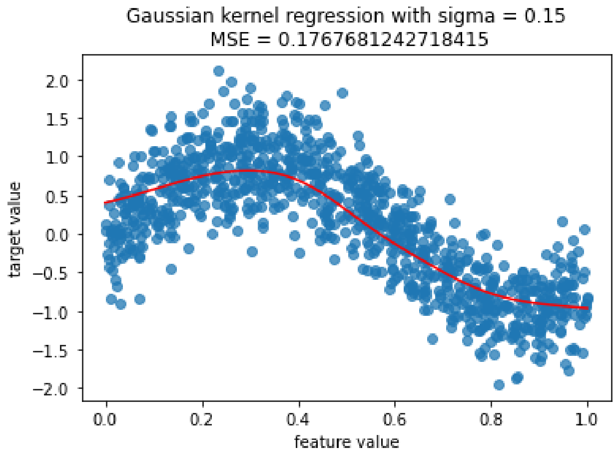


Figure 5: $\sigma = 0.2$

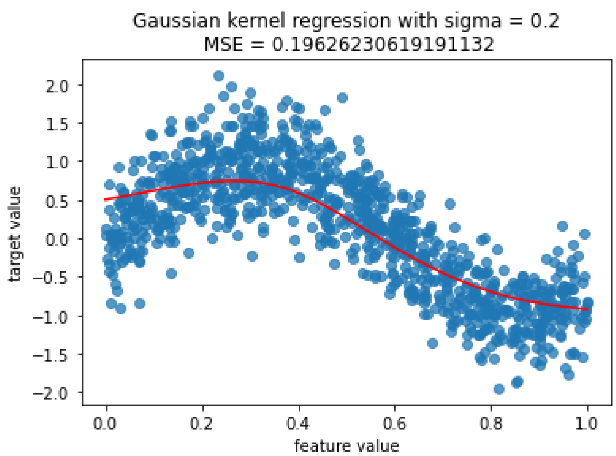


Figure 6: $\sigma = 0.5$

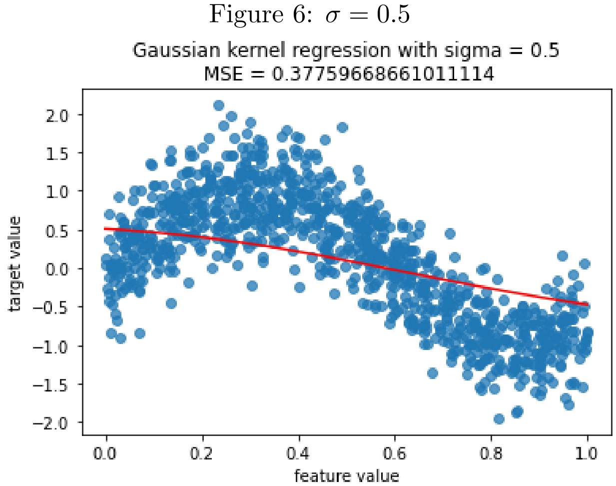
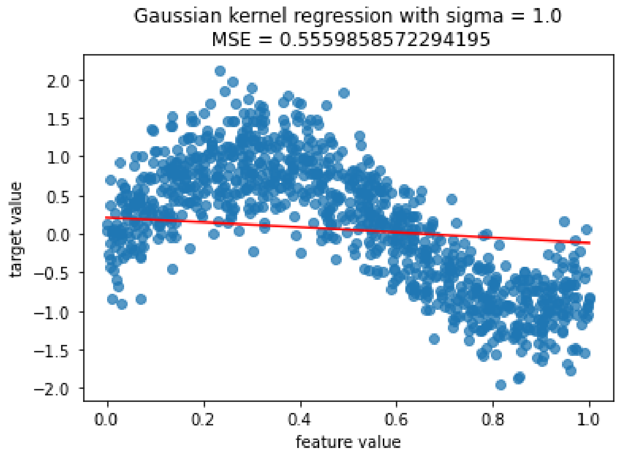


Figure 7: $\sigma = 1.0$



Sigma value with the smallest MSE is 0.1.
Sigma represents bandwidth value, if it is too small, the model overfits because it assigns high weights to data points in training dataset; if the value is too large, then the model underfits and cannot reflect true distribution of data.
Based on the comparison, as the value of sigma increases, the MSE decreases first, then increase. The optimal value of sigma is about 0.1.

4 Gradient Descent on Logistic Regression

1. Accuracy on the test set: 0.81

   Logistic regression coefficient(Scikit): [8.77256274e-05 3.21072680e-05 -8.33843505e-06 -6.45662790e-04 -1.03208486e-03 3.75014841e-03 9.91122852e-06 -7.39092935e-06 -3.97880851e-06 -1.54728944e-03 -7.71032299e-07]

2. 1) The log likelihood when $Y = 1$ is

$$\begin{aligned} LL(f(x;w)) &= log(f(x;w)) && (1)\\ &= log(h_w(x)) && (2)\\ &= log(\frac{1}{1+e^{-w^Tx}}) && (3)\\ &= -log(1+e^{-w^Tx}) && (4) \end{aligned}$$

When $Y = 0$

$$\begin{aligned} LL(f(x;w)) &= log(1-h_w(x)) && (5)\\ &= log(\frac{e^{-w^Tx}}{1+e^{-w^Tx}}) && (6)\\ &= log(e^{-w^Tx}) - log(1+e^{-w^Tx}) && (7)\\ &= -w^Tx - log(1+e^{-w^Tx}) && (8) \end{aligned}$$

$$LL(f(x;w)) = \begin{cases} -log(1+e^{-w^Tx}) & Y=1 \\ -w^Tx - log(1+e^{-w^Tx}) & Y=0 \end{cases} \quad (9)$$

2) Calculating the derivative:

$$\frac{\partial LL(f(x;w))}{\partial w} = \begin{cases} \frac{xe^{-w^T x}}{1+e^{-w^T x}} & Y = 1 \\ -\frac{x}{1+e^{-w^T x}} & Y = 0 \end{cases} \tag{10}$$

$$\frac{\partial LL(f(x;w))}{\partial w} = (h_w(x) - Y)^T X \tag{11}$$

3) Writing out the update formula:

$$w := w + \eta(h_w(x) - Y)^T X \tag{12}$$

3. Accuracy on the training set: 0.665 Accuracy on the test set: 0.6675

   Logistic regression coefficient (SGD): [-3.42517924e-02 5.18403885e-02 -1.10732366e-02 -3.15681367e-01 -9.61945981e-01 3.42749351e+00 1.53979273e-02 -1.56133962e-02 -6.72767542e-03 -1.80482548e+00 -8.99456290e-04]

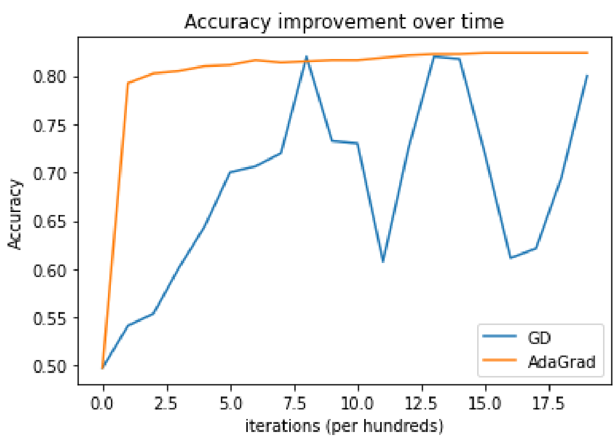
4. Accuracy on the training set: 0.82375

   Accuracy on the test set: 0.815

   Logistic regression coefficient (AdaGrad):[8.56662093e-05 7.32614055e-03 -4.06504586e-03 -1.84267619e-04 -1.05933061e-03 4.09573947e-03 7.04325844e-03 -7.02781789e-03 -6.96851681e-03 -2.22061428e-03 -2.22196570e-03]

5. Comparison of Scikit, GD and AdaGrad convergence

Figure 8: Accuracy vs. iteration for SGD and AdaGrad (2 points)



The AdaGrad curve is more smooth, and the GD curve is unstable. The AdaGrad has better accuracy on the test dataset. Because the AdaGrad adjust learning rate so that frequent updated features will be updated less, thus avoid saddle points better.