CIS 419/519: Homework 2

Zhuheng Jiang

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: { Yupeng Li, Chang Liu, Yihang Xu, Jiatong Sun, Mai'an Zhang}

1 Gradient Descent

a. Using a constant value for learning rate implies the weight is updated with a constant step size at each iteration while moving toward a minimum of a loss function. This may cause some issues: the convergence will be slow if learning rate is too small, and the model may overshoot or not be convergent if the learning rate is too large.

b. Using a learning rate a function of steps k implies the weight is updated with a changing step size at each iteration, which may adjust the step size to fit the condition. Larger learning rates at the beginning can effectively increase the speed and efficiency when it is far from the minimum and smaller learning rate when approaching the minimum will avoid the overshooting the minimum, failure to converge or even divergence.

2 Linear Regression [CIS 519 ONLY]

According to the OLS closed form solution, the parameters θ can be calculated with:

$$\theta = (X^T X)^{-1} X^T y$$

So the θ's transpose is:

$$\begin{aligned} \theta^T &= [(X^T X)^{-1} X^T y]^T \\ &= \mathrm{y}^T X((X^T X)^{-1})^T \end{aligned}$$

Since $h(x_i) = \theta^T x_i$, $f(x)$ can be expressed as :

$$\begin{aligned} f(x) &= \theta^T x \\ &= \mathrm{y}^T X((X^T X)^{-1})^T x \\ &= \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix} X((X^T X)^{-1})^T x \end{aligned}$$

According to the question,

$$f(x) = \textstyle\sum_{i=1}^{n} l_i(x; X) y,$$

So we can rewrite the $f(x)$ as:

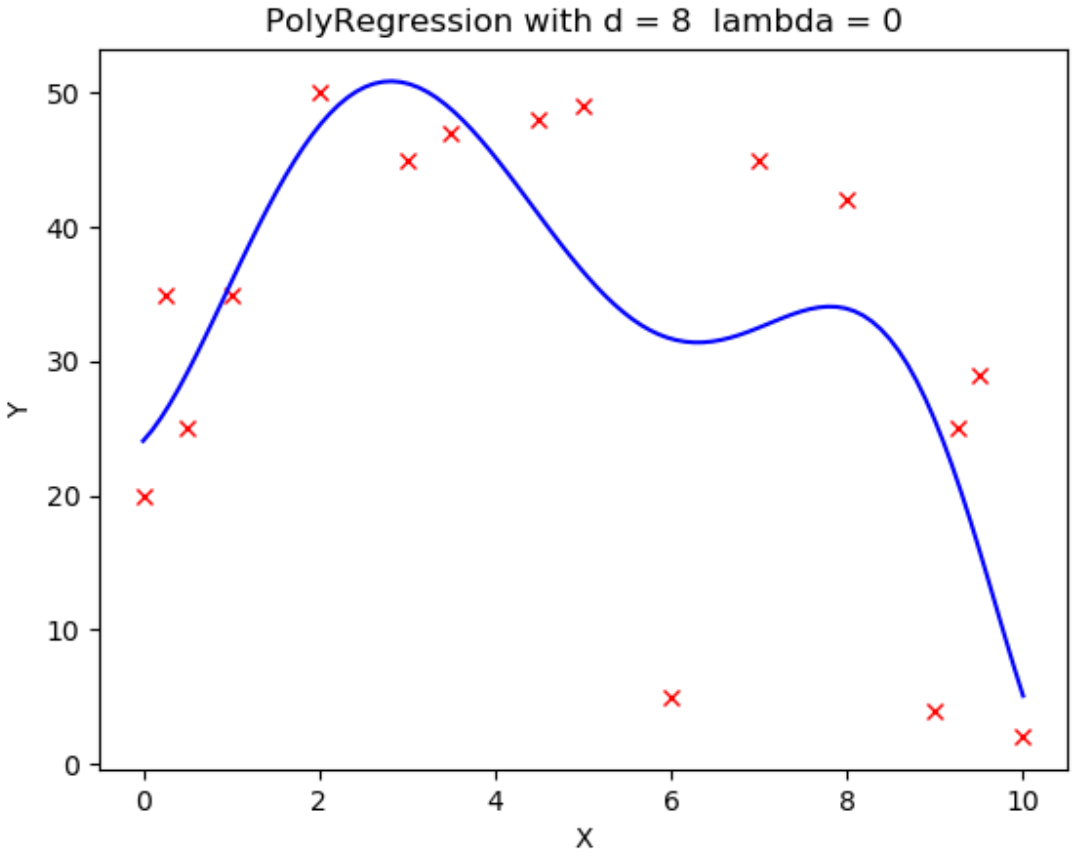
$$f(x) = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix} \begin{bmatrix} l_1 \\ l_2 \\ \dots \\ l_n \end{bmatrix}$$

The łcan be denoted as:

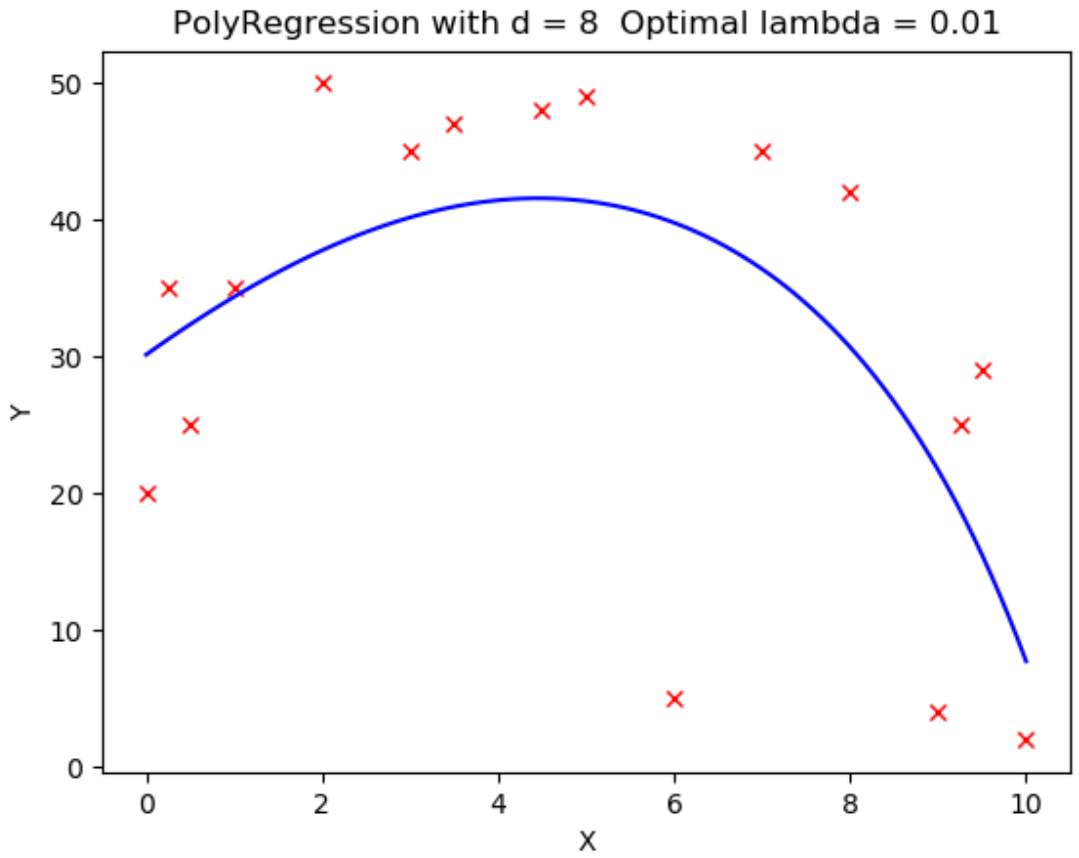
$$l(x; X) = X((X^T X)^{-1})^T x$$

3 PART II: PROGRAMMING EXERCISES

The following two plots are drawn by Polynomial regression program. The first plot is the curve without Lambda tuning under degree of 8, learning rate of 0.25 and epsilon of 1E-4:



And after applying the automatic lambda tuning , the optimal lambda was found before predicting the function. And the following plot shows the curve after tuning Lambda under degree of 8, learning rate of 0.25 and epsilon of 1E-4. The optimal lambda after tuning is 0.01.



Conclusions:

From the two plots we can find that the lambda can effectively reduce the model's overfitting. The curve becomes closer to the normal data points and the influence from the outliers is ameliorated.

And the lambda, or the regularization's essence is to get a better balance between the two goals of regression: 1. Try to modify the model to fit data better, 2. Keep the parameters as small as possible within a reasonable range to avoid overfitting. Applying the regularization can attenuate the weight the data can exert on the model which can eventually help improve the performance. But in my test experiences, applying this part cannot effectively reduce the computational cost of the program or reduce the steps before convergence.