Assignment 2 Solutions

1 Linear Regression and Model Selection

Please refer to this Jupyter notebook for the completed code.

1.1 Cross-Validation

Increasing the degree of polynomial decreases the training loss average due to more expressive models. Increasing the polynomial degree reduces the Validation and Test losses to a minimum at the degree 3. Beyond degree 3, they increase. Models with a polynomial degree of less than three underfit the data. As the polynomial degree increases, the model gets more prone to overfitting the training data.

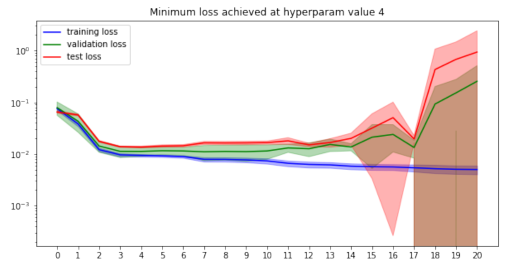


Figure 1: Mean and STD of Training, Validation, and Test loss for different polynomial degree.

1.2 Regularization

Regularization penalizes the model weights from having large values by adding the squared magnitudes of weights to the loss function. Therefore, we achieve the best loss with a higher polynomial degree due to more generalization; in fact, coefficients of the higher order terms tend to be very small. The standard deviations of the three losses are also much smaller compared to the model without regularization. However, the regularization method requires the choice of an additional hyperparameter to train the model.

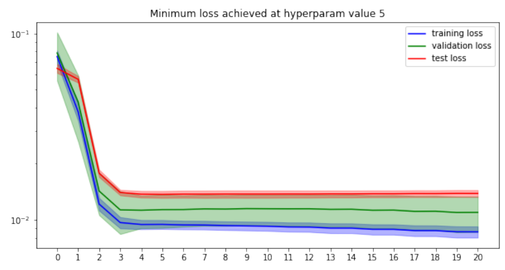


Figure 2: Means and standard deviations of Training, Validation, and Test losses with $\lambda = 0.05$

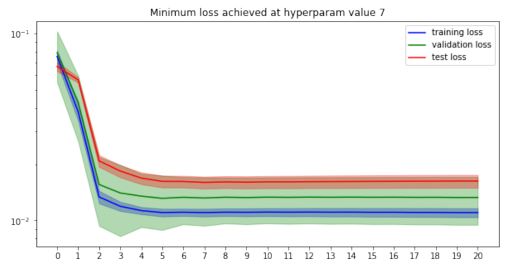


Figure 3: Means and standard deviations of Training, Validation, and Test losses with $\lambda = 0.5$

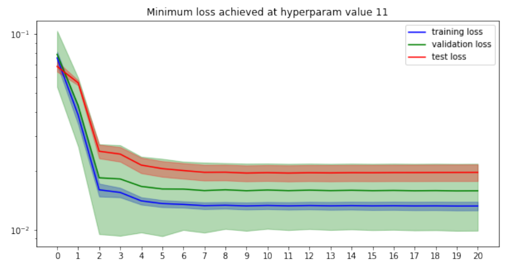


Figure 4: Means and standard deviations of Training, Validation, and Test losses with $\lambda = 1$

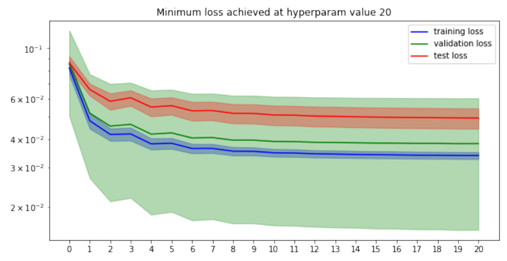


Figure 5: Means and standard deviations of Training, Validation, and Test losses with $\lambda = 5$

2 MAP Estimate and Bayesian Inference

a) Calculate w_{MAP}.

Solution:

Stacking the data points, we have $X = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$, $X^T = \begin{bmatrix} 1 & 3 \end{bmatrix}$, $Y = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{6} \end{bmatrix}$.

$$\begin{aligned}
\hat{w}_{MAP} &= (X^\top X + \lambda I)^{-1} X^\top Y \\
&= (\begin{bmatrix} 1 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix} + 1 \cdot I)^{-1} \cdot \begin{bmatrix} 1 & 3 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{3} \\ \frac{1}{6} \end{bmatrix} \\
&= \frac{1}{11} \cdot (1 \cdot \frac{1}{3} + 3 \cdot \frac{1}{6}) \\
&= \frac{1}{11} \cdot \frac{5}{6} \\
&= \frac{5}{66}
\end{aligned}$$

b) Calculate $p(w \mid \mathcal{D})$, where $\mathcal{D} = \{(1, \frac{1}{3}), (3, \frac{1}{6})\}$. What is the mean $\mathbb{E}[w|\mathcal{D}]$ and standard deviation σ_w of the posterior distribution $p(w \mid \mathcal{D})$?

Solution:
We know

$$P(w|D) = \frac{P(w) \cdot P(\mathcal{D}|w)}{P(\mathcal{D})} \tag{1}$$

For component $P(w)$ in (1):

$$\begin{aligned}
P(w) &= \frac{1}{\sqrt{(\frac{2\pi\sigma^2}{\lambda})^n}} \cdot \exp(-\frac{\lambda}{2\sigma^2} \cdot \|w\|_2^2) \\
&= \frac{1}{\sqrt{(2\pi)^2}} \cdot \exp(-\frac{1}{2} \cdot \|w\|_2^2)
\end{aligned}$$

For component $P(\mathcal{D}|w)$ in (1):

$$\begin{aligned}
P(\mathcal{D}|w) &= \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp(\sum_{n=1}^{n} -\frac{(y_i - wx_i)^2}{2\sigma^2}) \\
&= \frac{1}{\sqrt{2\pi}} \cdot \exp(-\frac{(\frac{1}{3} - w \cdot 1)^2}{2} - \frac{(\frac{1}{6} - w \cdot 3)^2}{2})
\end{aligned}$$

Therefore,

$$\begin{aligned}
P(w) \cdot P(\mathcal{D}|w) &= \frac{1}{\sqrt{2\pi\sigma^2} \cdot \sqrt{(\frac{2\pi\sigma^2}{\lambda})^n}} \cdot \exp(-\frac{\lambda}{2\sigma^2} \cdot \|w\|_2^2 + \sum_{n=1}^{n} -\frac{(y_i - wx_i)^2}{2\sigma^2}) \\
&= \frac{1}{\sqrt{(2\pi)^3}} \cdot \exp(-\frac{1}{2} \cdot \|w\|_2^2 - \frac{(\frac{1}{3} - w \cdot 1)^2}{2} - \frac{(\frac{1}{6} - w \cdot 3)^2}{2})
\end{aligned}$$

$P(\mathcal{D})$ does not depend on w, so it is just part of a normalizing constant. Let $P(\mathcal{D}) = k$.

$$\begin{aligned}
P(w|D) &= \frac{P(w)\cdot P(\mathcal{D}|w)}{P(\mathcal{D})} \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\exp\{-\frac{1}{2}w^2-\frac{1}{2}\cdot[(\frac{1}{3}-w)^2+(\frac{1}{6}-3w)^2]\} \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\exp[-\frac{1}{2}w^2-\frac{1}{2}\cdot(\frac{1}{9}+w^2-\frac{2}{3}w+\frac{1}{36}+9w^2-w)] \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\exp[-\frac{1}{2}\cdot(11w^2-\frac{5}{3}+\frac{5}{36})] \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\exp[-\frac{11}{2}\cdot(w^2-\frac{5}{33}w+\frac{5}{396})] \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\exp\{-\frac{11}{2}\cdot[(w-\frac{5}{66})^2+\frac{5}{726}]\} \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\frac{1}{exp(\frac{11}{2}\cdot\frac{5}{726})}\cdot\exp[-\frac{11}{2}\cdot(w-\frac{5}{66})^2] \\
&= \frac{1}{k}\cdot\frac{1}{\sqrt{(2\pi)^3}}\cdot\frac{1}{exp(\frac{11}{2}\cdot\frac{5}{726})}\cdot\exp[-\frac{(w-\frac{5}{66})^2}{\frac{2}{11}}] \\
&\propto \exp[-\frac{(w-\frac{5}{66})^2}{\frac{2}{11}}]
\end{aligned}$$

Note that in all the steps above, we could have just dropped all the constants in every step and used "$\propto$" instead of "$=$".

Since $P(w|D) \propto \exp[-\frac{(w-\frac{5}{66})^2}{\frac{2}{11}}]$, $P(w|D)$ is a normal distribution. Its mean is $\frac{5}{66}$ (which agrees with w_{MAP}), and its standard deviation is $\frac{1}{\sqrt{11}}$. Therefore, $P(w|D) \sim N(\frac{5}{66}, \frac{1}{11})$. The pdf is given by

$$P(w|D) = \frac{1}{\sqrt{\frac{2}{11}\pi}}\exp[-\frac{(w-\frac{5}{66})^2}{\frac{2}{11}}].$$

The constant in front can now be read off based on the the mean and variance.

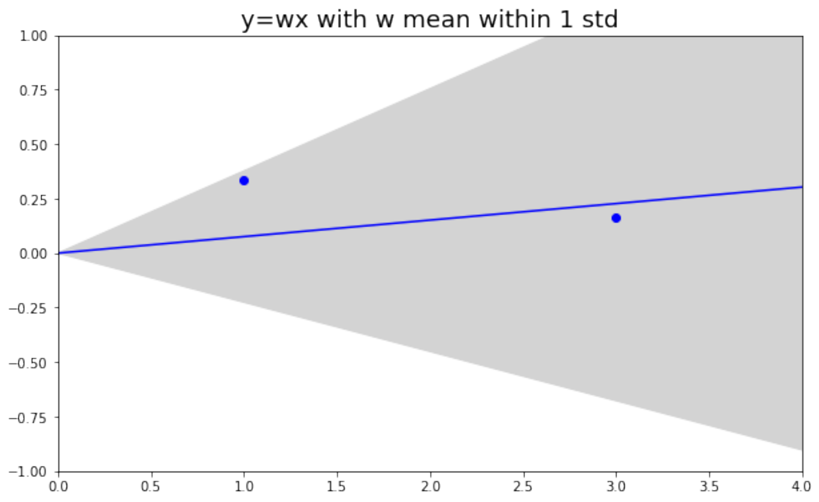
c) Plot the two data points, the mean regression line $y = \mathbb{E}[w|\mathcal{D}]x$, and the two lines that are one-standard-deviation above and below the mean regression line, i.e. $y = (\mathbb{E}[w|\mathcal{D}] \pm \sigma_w)x$.

Sample Code:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
```

```
w_mean = 5/66
std = 1/math.sqrt(11)
xval = np.arange(-10, 10, 0.5)
yval = w_mean * xval
yval_plus1std = (w_mean + std) * xval
yval_minus1std = (w_mean - std) * xval
points_x = np.array([1, 3])
points_y = np.array([1/3, 1/6])

plt.figure(1, figsize=(10,6))
plt.title('y=wx␣with␣w␣mean␣within␣1␣std', fontsize=18)
plt.xlim(0,4)
plt.ylim(-1,1)
plt.plot(xval, yval, color='blue')
plt.fill_between(xval,yval_minus1std,yval_plus1std,color='lightgray')
plt.scatter(points_x, points_y, color = 'blue')
```



d) From the plot, which data point is closer to the mean regression line? Explain why in 1-2 sentences.

Solution:
The data point $(3, \frac{1}{6})$ is closer to the mean regression line. This is because the prior $P(w)$ has a mean of zero, so it biases the mean regression line towards $w = 0$, which represents a line with zero slope.

3 Nonlinear Optimization

This Jupyter notebook provides the completed code for the following parts.

a) Plot the objective function for the specified range of the inputs in the notebook.

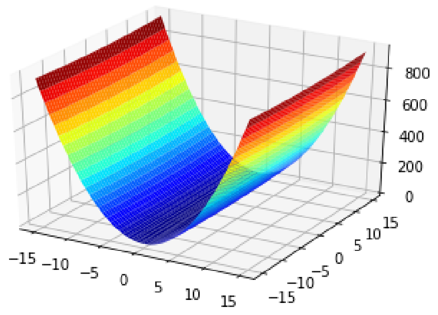


Figure 6: Objective function f

b) Implement a basic Gradient Descent algorithm for the objective function (1).

- Use step size of 0.2, total iterations of 200, and initial point of $(-10, 10)$ for the objective function. Provide a contour plot of the solutions.

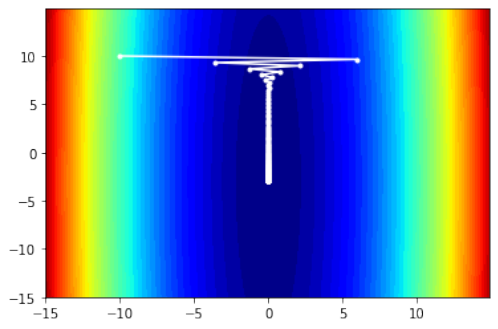


Figure 7: Basic Gradient Descent algorithm

- Change the step size to 0.3. Report your observation.

  Using the step size of 0.3, one can observe the values found by the solution are very large. Hence, the Gradient Descent algorithm can not find the optimal solution.

c) Implement the Adaptive Gradient (AdaGrad) algorithm for the objective function (1).

- Use step size of 0.2, total iterations of 200, and initial point of $(-10, 10)$ for the objective function. Provide a contour plot of the solutions.

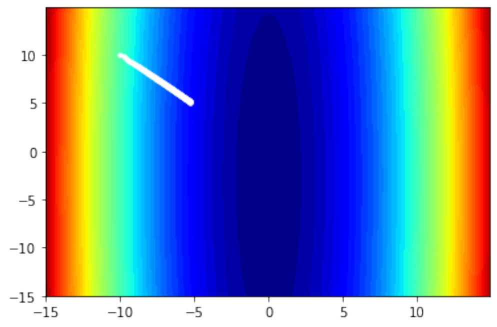


Figure 8: AdaGrad algorithm - step size of 0.2

- Change the step size to 0.3. Report and plot your observation.

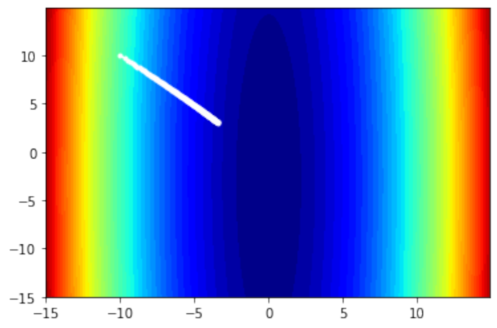


Figure 9: AdaGrad algorithm - step size of 0.3

Using the step size of 0.3, one can observe that the convergence is still slow. Using a greater value for the step size will fix this problem.

d) Implement the Adam algorithm for the objective function (1).

- Use step size of 0.2, total iterations of 200, and initial point of $(-10, 10)$ for the objective function. Use default values for other parameters. Provide a contour plot of the solutions.

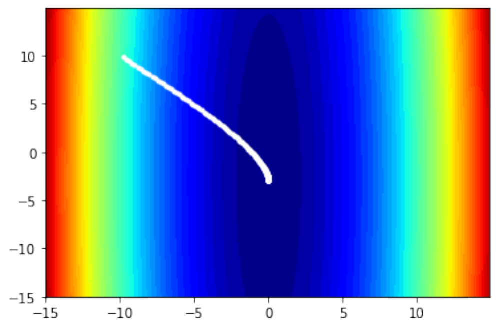


Figure 10: Adam algorithm

- Does the solution converge to the minimal objective value?

  Yes. The solution converges to the minimal objective value $(0, -3)$.