EECS 545 Machine Learning: Homework #1

Due on January 25, 2022 at 12pm

Professor Honglak Lee Section A

Yuang Huang
yahuang@umich.edu

Problem 1

Linear regression on a polynomial

Solution

Part a(i): Find the coefficiets

• **Batch gradient descent(BGD)**

Learning Rate: 0.05

Epoch number: 200

Degree: 1

Coefficents: $\omega_0 = 1.947$, $\omega_1 = -2.824$ in the function $h(\mathbf{x}, \mathbf{w}) = \omega_0\phi_0(\mathbf{x}) + \omega_1\phi_1(\mathbf{x})$

Initial value of the coefficients: Generated by taking random values from $\mathcal{N}\ (0, 1)$, we choose $\omega_0 = -1.376$, $\omega_1 = -1.468$

• **Stochastic gradient descent(SGD)**

Learning Rate: 0.05

Epoch number: 200

Degree: 1

Coefficents: $\omega_0 = 1.921$, $\omega_1 = -2.853$ in the function $h(\mathbf{x}, \mathbf{w}) = \omega_0\phi_0(\mathbf{x}) + \omega_1\phi_1(\mathbf{x})$

Initial value of the coefficients: Generated by taking random values from $\mathcal{N}\ (0, 1)$, we choose the same as the batch gradient descent where $\omega_0 = -1.376$, $\omega_1 = -1.468$

Part a(ii): Over-fitting

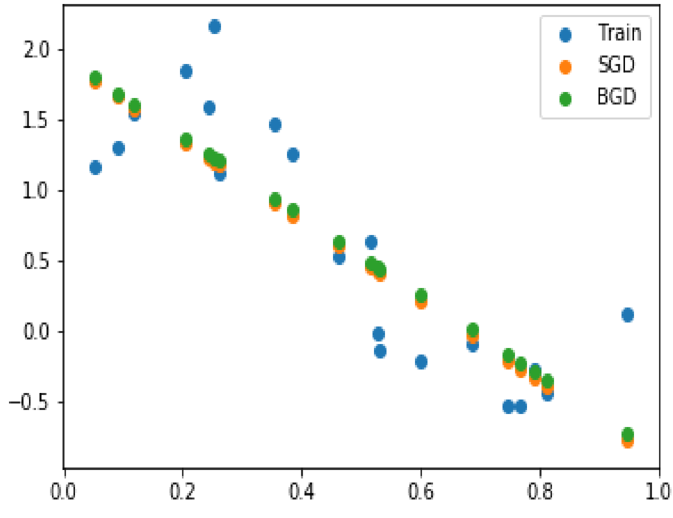

Figure 1: Fitting linear regression

Problem 1 continued on next page...

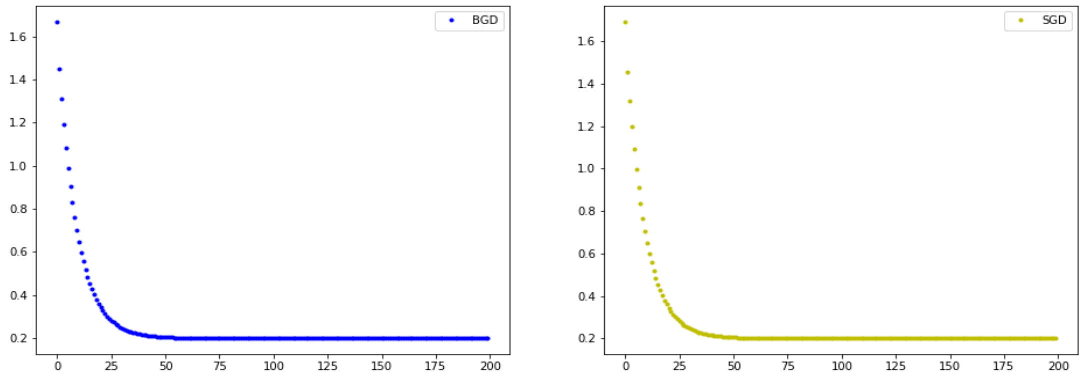

Figure 2: E_{MS} of BGD and SGD

Fig. 1 illustrates training data(blue), the prediction points using the BGD(green) and the SGD(orange), respectively. It is known from the figure that the fit of both methods is good and close. Fig. 2 illustrates the mean squared error (E_{MS}) curves of the BGD and the SGD. The convergence speed of the two methods is very close (so I draw the curves separately), and they both converge at around $epoch = 50$, and converge to $E_{MS} = 0.2$. In theory, the SGD will converge faster. In problem1, it may be hard to distinguish the convergence speed of the two methods because the training set is too small.

Part b(i):

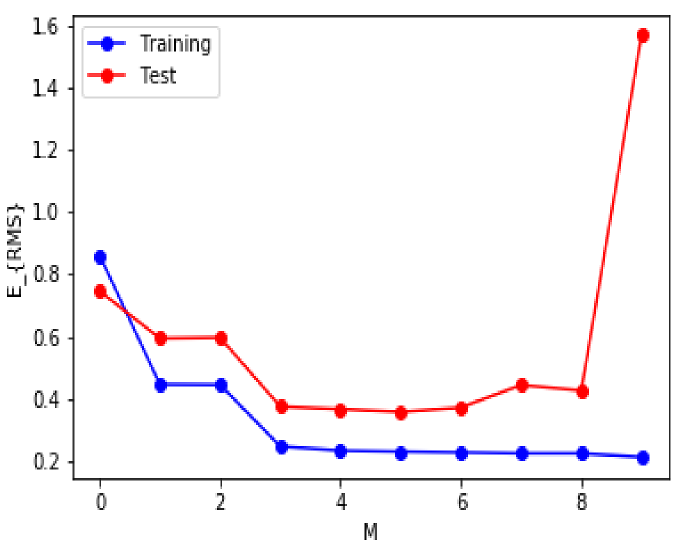

Figure 3: The trend of E_{RMS} changing with degree

Part b(ii):

Fig. 3 illustrates the trend of E_{RMS} changing with degree. It is easy to know from the figure that 0, 1, 2, 3 degree polynomials under-fitting the date and 9 degree polynomial over-fitting the data. I think 5 degree best fits the date because the Root-Mean-Square Error (E_{RMS}) of 5 degree polynomial function is relatively smaller and it needs relatively less calculations.

Part c(i):

The closed form solution of the ridge regression is:

$$W_{ML} = (\mathbf{\Phi^T \Phi} + \lambda \mathbf{I})^{-1} \mathbf{\Phi^T y} \tag{1}$$

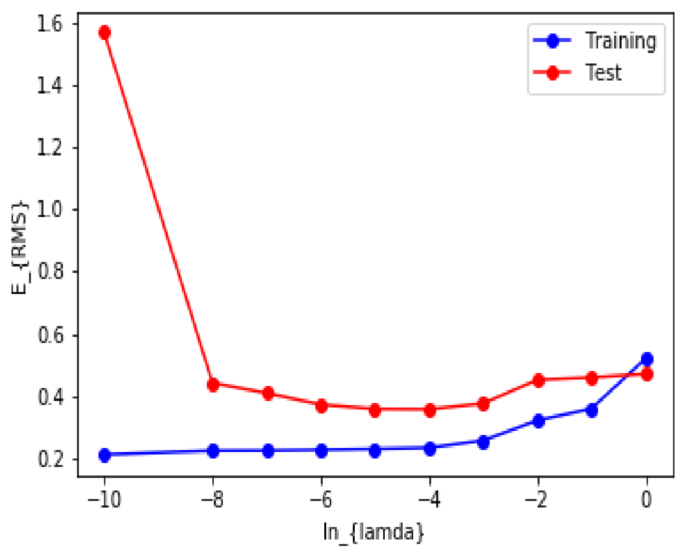

Figure 4: The trend of E_{RMS} changing with regulization factor λ using closed form solution

Part c(ii):

As shown in Fig. 5, the closed form solution reaches the lowest test E_{RMS} at $\lambda = 10^{-4}$, so $\lambda = 10^{-4}$ seemed to work the best.

Problem 2

Locally weighted linear regression

Solution
Part 2(a):

$$\begin{aligned} E_D(\mathbf{w}) &= (\mathbf{Xw}-\mathbf{y})^T\mathbf{R}(\mathbf{Xw}-\mathbf{y}) \\ &= \sum_{i=1}^{N}\sum_{j=0}^{M-1} R_{ij}(\mathbf{x}^{(\mathbf{i})}w_j - y^{(i)})^2 \\ &= \sum_{i=1}^{N} R_{ij}(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2 \end{aligned} \tag{2}$$

where R_{ij} is the element of matrix R and

$$R_{ij} = \begin{cases} \frac{1}{2}r^{(i)}, & i = j \\ 0, & i \neq j \end{cases} \tag{3}$$

Part 2(b):
Expand what we got in part2(a), we will get

$$\begin{aligned} E_D(\mathbf{w}) &= (\mathbf{Xw}-\mathbf{y})^T\mathbf{R}(\mathbf{Xw}-\mathbf{y}) \\ &= \mathbf{w^TX^TRXw} - \mathbf{y^TRXw} - \mathbf{w^TX^TRy} + \mathbf{y^TRy} \\ &= \mathbf{w^TX^TRXw} - 2\mathbf{w^TX^TRy} + \mathbf{y^TRy}, \end{aligned} \tag{4}$$

so the gradient of $E_D(\mathbf{w})$ is shown by:

$$\begin{aligned} \nabla_{\mathbf{w}} E_D(\mathbf{w}) &= \mathbf{X^TRXw} - \mathbf{X^TRy} \\ &= 0 \end{aligned} \tag{5}$$

and the closed form solution is descrided by:

$$\Rightarrow \mathbf{w} = (\mathbf{X^TRX})^{-1}\mathbf{X^TRy} \tag{6}$$

Part 2(c):

$$P(\mathbf{Y}|\mathbf{x}^{(i)};\mathbf{w}) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma^{(i)}} exp\left(-\frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{2(\sigma^{(i)})^2}\right) \tag{7}$$

$$\ln P = -\sum_{i=1}^{N} \ln\sigma^{(i)} - \frac{N}{2}\ln 2\pi - \sum_{i=1}^{N}\left(-\frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{2(\sigma^{(i)})^2}\right) \tag{8}$$

Taking gradient of function $\ln P$:

$$\nabla_{\mathbf{w}} \ln P = -\nabla_{\mathbf{w}} \sum_{i=1}^{N} \frac{(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)})^2}{2(\sigma^{(i)})^2} \tag{9}$$

Problem 2 continued on next page...

If

$$r^{(i)} = \frac{1}{2(\sigma^{(i)})^2} \tag{10}$$

in eq(2), the result of eq(9) will equals to the result of eq(2).

Part d(i):

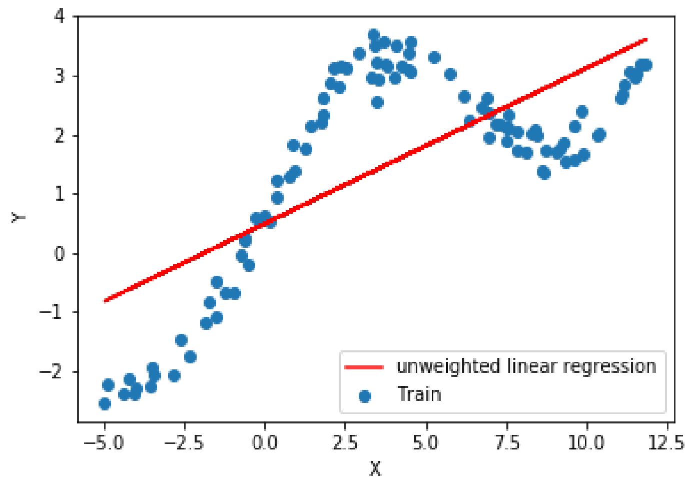

Figure 5: unweighted linear regression

Part d(ii):

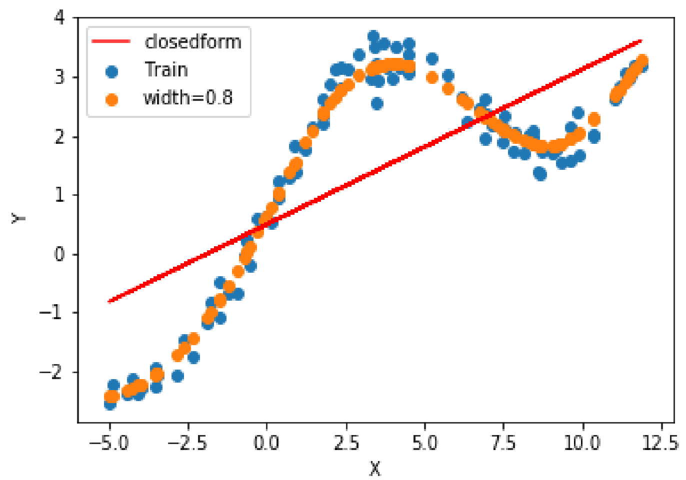

Figure 6: Locally weighted linear regression with a bandwidth parameter $\tau = 0.8$

Part d(iii):

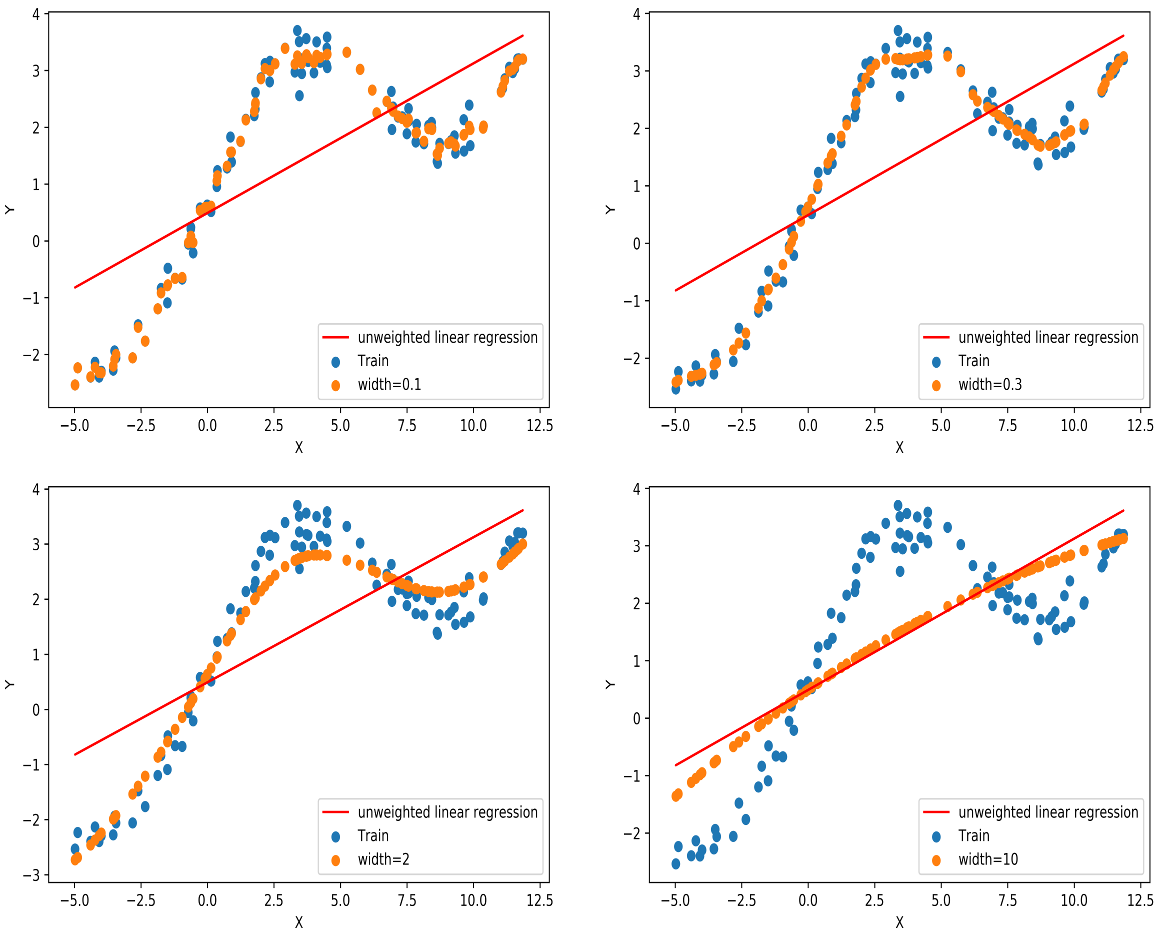

Figure 7: unweighted linear regression with different bandwidth parameters

In eq(11), we note that the weights depend on the particular point x at which we are trying to evaluate x. The bandwidth parameter τ controls how quickly the weight of a training example falls off with distance of its $x^{(i)}$. If $|x - x^{(i)}|$ is large, the weight is small, if $|x - x^{(i)}|$ is small, the weight is close to 1. Hence, if τ is small, its effect will equal to the large $|x - x^{(i)}|$ which let $\mathbf{w}$ more 'emphasis' on reducing the error at this point. If τ is large, the weigh is close to 1, which is close to unweighted linear regression. (This explanation is partly inspired by stanford cs229-note1.)

$$r^{(i)} = \exp\left(-\frac{(x - x^{(i)})^2}{2\tau^2}\right) \tag{11}$$

Problem 3

Derivation and Proof

Solution
Part a:
1D case of linear function:

$$h(x) = \omega_1 x + \omega_0 \tag{12}$$

The sum squared error:

$$L = \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - h(x^{(i)}))^2 \tag{13}$$

From eq(12) and eq(13), we can get the partial derivation of L:

$$\begin{aligned}
\frac{\partial L}{\partial \omega_0} &= \frac{\partial \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \omega_1 x^{(i)} - \omega_0)^2}{\partial \omega_0} \\
&= -\sum_{i=1}^{N}(y^{(i)} - \omega_1 x^{(i)} - \omega_0) \\
&= N\omega_0 + \omega_1 \sum_{i=1}^{N} x^{(i)} - \sum_{i=1}^{N} y^{(i)} \\
&= N(\omega_0 + \omega_1 \bar{X} - \bar{Y}) \\
\frac{\partial L}{\partial \omega_1} &= \frac{\partial \frac{1}{2}\sum_{i=1}^{N}(y^{(i)} - \omega_1 x^{(i)} - \omega_0)^2}{\partial \omega_1} \\
&= \sum_{i=1}^{N} x^{(i)}(y^{(i)} - \omega_1 x^{(i)} - \omega_0) \\
&= \sum_{i=1}^{N}(x^{(i)}y^{(i)} - \omega_1 {x^{(i)}}^2 - \omega_0 x^{(i)}) \\
&= \sum_{i=1}^{N}(x^{(i)}y^{(i)} - \omega_1 {x^{(i)}}^2) - N\omega_0 \bar{X}
\end{aligned} \tag{14}$$

where $\bar{X}$ is the mean of $\left\{x^{(1)}, x^{(2)}, \cdots, x^{(N)}\right\}$ and $\bar{Y}$ is the mean of $\left\{y^{(1)}, y^{(2)}, \cdots, y^{(N)}\right\}$. Let the partial derivation of L be equal to zero, respectively, the solution for ω_0 and ω_1 for this 1D case of linear regression is derived as follow:

$$\begin{aligned}
&\frac{\partial L}{\partial \omega_0} = N(\omega_0 + \omega_1 \bar{X} - \bar{Y}) = 0 \\
&\Rightarrow \omega_0 = \bar{Y} - \omega_1 \bar{X}
\end{aligned} \tag{15}$$

Problem 3 continued on next page...

$$
\begin{aligned}
\frac{\partial L}{\partial \omega_1} &= \sum_{i=1}^{N}(x^{(i)}y^{(i)} - \omega_1 x^{(i)^2}) - N\omega_0 \bar{X} \\
&= \sum_{i=1}^{N}(x^{(i)}y^{(i)} - \omega_1 x^{(i)^2}) - N(\bar{Y} - \omega_1 \bar{X})\bar{X} \\
&= \sum_{i=1}^{N}(x^{(i)}y^{(i)} - \omega_1 x^{(i)^2}) - N\bar{Y}\bar{X} - N\omega_1 \bar{X}^2 = 0 \\
\Rightarrow \omega_1 &= \frac{\sum_{i=1}^{N} x^{(i)}y^{(i)} - N\bar{Y}\bar{X}}{\sum_{i=1}^{N} x^{(i)2} - N\bar{X}^2} \\
&= \frac{\frac{1}{N}\sum_{i=1}^{N} x^{(i)}y^{(i)} - \bar{Y}\bar{X}}{\frac{1}{N}\sum_{i=1}^{N} x^{(i)2} - \bar{X}^2}
\end{aligned} \tag{16}
$$

Part b(i):

$$
\begin{aligned}
\mathbf{A} &= \mathbf{U\Lambda U^T} \\
&= \sum_{i=1}^{d} \lambda_i u_i u_i^T
\end{aligned} \tag{17}
$$

$$
\begin{aligned}
\mathbf{z^T A z} &= \mathbf{z^T U \Lambda U^T z} \\
&= \sum_{i=1}^{d} \lambda_i z^T u_i u_i^T z \\
&= \sum_{i=1}^{d} \lambda_i \|u_i^T z\|_2^2, (z \neq 0)
\end{aligned} \tag{18}
$$

It is obviously that $\sum_{i=1}^{d} \lambda_i \|u_i^T z\|_2^2 > 0$ iff $\lambda_i > 0$ for each i. So, $\mathbf{A}$ is PD iff $\lambda_i > 0$ for each i.

Part b(ii):
The matrix $\mathbf{\Phi^T \Phi}$ is real and symmetric, so it can be expressed by $\mathbf{\Phi^T \Phi} = \mathbf{U\Lambda U^T}$, so

$$
\begin{aligned}
\mathbf{z^T \Phi^T \Phi z} &= \mathbf{z^T U \Lambda U^T z} \\
&= \sum_{i=1}^{d} \lambda_i^2 z^T u_i u_i^T z \\
&= \sum_{i=1}^{d} \lambda_i^2 \|u_i^T z\|_2^2 \geq 0, (z \neq 0)
\end{aligned} \tag{19}
$$

$$\begin{aligned}\mathbf{z^T}(\mathbf{\Phi^T\Phi}+\beta\mathbf{I})\mathbf{z} &= \mathbf{z^T}\mathbf{U}(\mathbf{\Lambda}+\beta)\mathbf{U^T}\mathbf{z} \\ &= \sum_{i=1}^{d}(\lambda_i^2+\beta)z^T u_i u_i^T z \\ &= \sum_{i=1}^{d}(\lambda_i^2+\beta)\|u_i^T z\|_2^2 > 0, (z \neq 0),\end{aligned} \tag{20}$$

because $(\lambda_i^2 + \beta)$ always larger than zero. Hence, for any $\beta > 0$, ridge regression makes the matrix $\mathbf{\Phi^T\Phi} + \beta\mathbf{I}$ PD.