CIS 419/519: Applied Machine Learning	Fall 2024

Homework 1

Handed Out: September 11 *Due: 7:59 pm October 2*

Name: Alan Wu

PennKey: alanlwu

PennID: 41855518

1 Declaration

- **Person(s) discussed with:** *Your answer*
- **Affiliation to the course: student, TA, prof etc.** *Your answer*
- **Which question(s) in coding / written HW did you discuss?** ***Your answer***
- **Briefly explain what was discussed.** *Your answer*

2 Multiple Choice & Written Questions

1. (a) Increase variance; bias stays the same
   (b) Decrease variance; Increase bias
   (c) Decrease variance; Increase bias
   (d) Variance increases; Bias stays the same
   (e) Variance stays the same; Bias stays the same
   (f) for the following values to decrease test loss
   n: Increase n
   λ: Increase λ
   d: Decrease d
   c: Decrease c
   α: Keep α the same

2. (a) To derive the gradient of the L1 regularization term for the:
   We need to simply take the gradient of the equation with respect to B_j.

$$L_{\ell_1} = \lambda \sum_{j=1}^{p} |B_j| \tag{1}$$

$$\frac{\partial L_{\ell_1}}{\partial B_j} = \lambda \operatorname{sign}(B_j) \tag{2}$$

We ignore the case where $B_j = 0$ and derive that equation from the loss function. Notice that the gradient of the L1 regularized loss function is only dependent on the sign of B_j and the magnitude of λ.

(b) We can analyze the effect of the L1 regularization term on the parameters of the model by looking at the gradient of both the L1 regularization term and the MSE loss function with respect to B_j.

We can derive the full gradient of the loss function as such:

$$L_{\ell_1} = \frac{1}{N}\sum_{i=1}^{N}(y_i - B^T x_i)^2 + \lambda \sum_{j=1}^{p} |B_j|$$

$$\frac{\partial L_{\ell_1}}{\partial B_j} = \frac{-2}{N}\sum_{i=1}^{N}(y_i - B^T x_i)x_{ij} + \lambda \text{ sign}(B_j)$$

Part 1: The MSE Loss Term
Our two cases for the MSE loss that we have to consider is when B_j is predictive of y_i and when it is weakly or not predictive of y_i. We know that the MSE loss is not dependent on the value of λ.

For predictive features, B_j will be larger and thus, the gradient of the MSE loss term will be larger in magnitude (positive or negative) than those of weakly predictive features. And in this way, the gradient of the MSE loss term for predictive features will change more than those that are weakly predictive because they have greater impact on minimizing the loss.

Thus, the magnitude of B_j is dependent on how strongly correlated/predictive the feature x_{ij} is in relation to y_i.

Part 2: The L1 Regularization term
As we derived earlier, the gradient of the L1 regularization term is only dependent on the sign of B_j and the magnitude of λ.

The equation is: $\frac{\partial L_{\ell_1}}{\partial B_j} = \lambda \text{ sign}(B_j)$

This observation gives us two cases where the feature B_j of the L1 loss regularization term will be scaled by the value of λ and the sign of B_j.
And when we scale λ, we will observe the following:
For small λ, the L1 regularization term will have a small impact on the overall gradient. With respect to B_j, this means that many B_j may be non-zero because the MSE loss term dominates the gradient. For both predictive and non-predictive features we may have non-zero values.
For large λ, the L1 regularization term may overtake the MSE loss term in the

gradient. For coeffcients B_j where B_j has small magnitude, the L1 regularization term will dominate and thus this parameter will be pushed down/up based on the sign of the parameter. Once the L1 regularization term dominates we will see that many B_j will shrink to 0, as once the B_j arrives at 0, it will stay there.

(c) The L2 regularization loss equation:

We know that the L2 regularization loss function is defined as:
$L_{\ell_2} = \frac{1}{N}\sum_{i=1}^{N}(y_i - B^T x_i)^2 + \lambda \sum_{j=1}^{p} B_j^2$
Therefore, the gradient of the L2 regularization term with respect to B_j is:
$\frac{\partial L_{\ell_2}}{\partial B_j} = 2\lambda B_j$

Just purely based on the gradient of the L2 regularization term, we can see that the L2 regularization term is dependent on the magnitude of both λ and B_j, as well as the sign of B_j. Compared to the gradient of the L1 regularization term, which solely depends on the sign of B_j and the magnitude of λ.

Because of this dependency, we can see that the L2 regularization term will not create a sparse matrix. In the scenario that we have values B_j that are very small (feature x_{ij} is weakly predictive of y_i), the gradient of the L2 regularization term will simply increase less as the value of B_j approaches 0. However, the L2 regularization term will not push the value of B_j to 0. Therefore, it does not create a sparse parameter matrix.

3. (a) We will do two derivations in this question. The first derivation will be the gradient of the loss function with respect to w_1^*. The second will be the gradient of the loss function with respect to w_0^*

.

We are given: $J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2$

First let us derive $\frac{\partial J(w)}{\partial w_1^*}$:

$$\begin{aligned}\frac{\partial J(w)}{\partial w_1^*} &= \frac{\partial}{\partial w_1^*}[\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2] \\ &= \frac{-2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)x_i\end{aligned}$$

Next, let us derive $\frac{\partial J(w)}{\partial w_0^*}$:

$$\begin{aligned}\frac{\partial J(w)}{\partial w_0^*} &= \frac{\partial}{\partial w_0^*}[\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)^2] \\ &= \frac{-2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1 x_i)\end{aligned}$$

(b) In order to show that $\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) = 0$, we need to manipulate the gradients found from the previous question to show that the equation is true.

We will find the optimal values of w_0 and w_1 by setting their gradient equations to zero, and algebraically determining a form for each term, then substituting it in to show that the above expression is true.

First, we will set the gradient of loss with respect to w_0 to 0:

$$\begin{aligned}
\frac{-2}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i) &= 0 \\
= \frac{-2}{n}[\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} w_0^* - w_1^* \sum_{i=1}^{n} x_i] & \\
= -2 * (\bar{y} - w_0^* - w_1^* \bar{x}) &= 0 \\
w_0^* &= \bar{y} - w_1^* \bar{x}
\end{aligned}$$

Next, we will set the gradient of loss with respect to w_1 to 0:

$$\begin{aligned}
\frac{2}{n}\sum_{i=1}^{n}(y_i - w_0 - w_1^* x_i)x_i &= 0 \\
= \frac{-2}{n}[\sum_{i=1}^{n}(y_i x_i - w0 x_i - w_1^* x_i^2)] &= 0 \\
= \frac{-2}{n}[\sum_{i=1}^{n}(y_i x_i - (\bar{y} - w_1^* \bar{x})x_i - w_1^* x_i^2)] &= 0 \\
= \frac{-2}{n}[\sum_{i=1}^{n}(y_i - \bar{y})x_i + w_1^* \sum_{i=1}^{n}(\bar{x} - x_i)x_i] &= 0 \\
= \frac{-2}{n}[\sum_{i=1}^{n}(y_i - \bar{y})x_i - w_1^* \sum_{i=1}^{n}(x_i - \bar{x})x_i] &= 0 \\
w_1^* &= \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}
\end{aligned}$$

A couple of notes in this derivation:

We substituted the equation for w_0^* into the equation to simplify our derivation. We also utilized a statistical algebraic manipulation twice, saying that $\sum_{i=1}^{n}(y_i - \bar{y})x_i = \sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})$. Same algebraic manipulation to say that $\sum_{i=1}^{n}(x_i - \bar{x})x_i = \sum_{i=1}^{n}(x_i - \bar{x})^2$

Now that we have those two derivations, we can substitute the optimal values of w_0^* and w_1^* into the original equation to show that it is true:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^* - w_1^* x_i)(x_i - \bar{x}) &= 0 \\
\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y} + w_1^*\bar{x} - w_1^* x_i)(x_i - \bar{x}) &= 0 \\
= \frac{1}{n}[\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) - w_1^* \sum_{i=1}^{n}(x_i - \bar{x})^2] & \\
= \frac{1}{n}[\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) - \frac{\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \sum_{i=1}^{n}(x_i - \bar{x})^2] & \\
= \frac{1}{n}[\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) - \sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x})] & \\
= \frac{1}{n}[0] = 0 &
\end{aligned}$$

(c)

4. (a) The value of the loss function at the beginning is going to be the loss function evaluated with the initial weights of $w = [0, 0, 0]^T$

$$\begin{aligned}
L(w) &= \frac{1}{N}\sum_{i=1} N(y_i - wx_i)^2 + \lambda||w||^2 \\
L(w_{\text{initial}}) &= \frac{1}{2}[(0 - [0,0]\cdot[1,-1])^2 + (1 - [0,0]\cdot[-1,-1])^2] + 1 \cdot ||[0,0]||^2 \\
&= \frac{1}{2}(1) = 0.5
\end{aligned}$$

Therefore, the value of the loss function initially is 0.5

(b) To find the final state of the trained weight vector after 2 steps and the corresponding value of the loss function, we need to compute the gradient at each step, and update the gradient and loss based on that value of the gradient. We will iteratively compute this:

Computing the gradient of the loss function:

$$\frac{\partial L(w)}{\partial w} = \frac{\partial}{\partial w}\frac{1}{N}\sum_{i=1} N(y_i - wx_i)^2 + \lambda||w||^2$$
$$= \frac{\partial}{\partial w}\frac{1}{N}\sum_{i=1} N(y_i - wx_i)^2 + \lambda\sum_{j=1}^{p} w_j^2$$
$$= \frac{-2}{N}\sum_{i=1}^{N}(y_i - wx_i)x_i + 2\lambda w$$

We also know that the gradient updates as the following: $w_{t+1} = w_t - \alpha\nabla L(w_t)$ where the weight vector of the next step is going to be the weight of the current step subtracted by the gradient of the loss function at the current step scaled by the learning rate

With this information, we can compute the weight vector after 2 steps by doing the following:

First we need to compute the value of the weight vector after step 1, of which we first need to compute the gradient at the first step:

$$\nabla L(w_{\text{initial}}) = \frac{-2}{2}\sum_{i=1}^{2}(y_i - [0,0]\cdot x_i)x_i + 0$$
$$= -1\cdot[(0-0)[1,-1] + (1-0)[-1,-1]]$$
$$= -1\cdot[-1,-1] = [1,1]$$

Now we update the weight vector:

$$w_{\text{step 1}} = w_{\text{initial}} - \alpha\nabla L(w_{\text{initial}})$$
$$= [0,0] - 1\cdot[1,1] = [-1,-1]$$
$$w_{\text{step 1}} = [-1,-1]$$

We need to compute the gradient at the second step with this new weights vector:

$$\nabla L(w_{\text{step 1}}) = \frac{-2}{2}\sum_{i=1}^{2}(y_i - [-1,-1]\cdot x_i)x_i + 2\cdot 1\cdot[-1,-1]$$
$$= -1\cdot[(0-[-1,-1]\cdot[1,-1])[1,-1] + (1-[-1,-1]\cdot[-1,-1])[-1,-1]] + [-2,-2]$$
$$= -1\cdot[[1,1]] + [-2,-2]$$
$$= [-1,-1] + [-2,-2] = [-3,-3] \tag{3}$$

Now we update the weight vector again using the gradient:

$$\begin{aligned} w_{\text{step 2}} &= w_{\text{step 1}} - \alpha \nabla L(w_{\text{step 1}}) \\ &= [-1, -1] - 1 \cdot [-3, -3] = [2, 2] \\ w_{\text{step 2}} &= [2, 2] \end{aligned}$$

We have to remember that all the operations are vectors, so we are subtracting/-multiplying vector.
Finally, we will compute the loss function after 2 epochs:

$$\begin{aligned} L(w_{\text{step 2}}) &= \frac{1}{2} \sum_{i=1}^{2} (y_i - [2, 2] \cdot x_i)^2 + 1 \cdot ||[2, 2]||^2 \\ &= \frac{1}{2}[(0 - [2, 2] \cdot [1, -1])^2 + (1 - [2, 2] \cdot [-1, -1])^2] + 1 \cdot (2^2 + 2^2) \\ &= \frac{1}{2}[(0 - 0)^2 + (1 - (-4))^2] + 8 \\ &= \frac{1}{2}(25) + 8 = 12.5 + 8 = 20.5 \end{aligned}$$

The final loss after 2 epochs of training is 20.5

3 Python Programming Questions