CS 335/337 - Assignment 1

Aakriti
190050002

Due 29th Aug 2021

Contents

1 OLS Regression in one variable

1.1 CS337: Theory

We have to find the expression for gradient vector $\nabla mse(w, b)$

As given in the question, $\nabla mse(w, b)$ is defined as:

$$\nabla mse(w,b) = \begin{pmatrix} \frac{\partial mse(w,b)}{\partial w} \\ \frac{\partial mse(w,b)}{\partial b} \end{pmatrix} \tag{1}$$

Here, mse is defined as:

$$\begin{aligned} mse(w,b) &= \frac{1}{N}\sum_{i=1}^{N}\left((wx_i + b) - y_i\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(w^2x_i^2 + b^2 + 2wbx_i - wx_iy_i - by_i + y_i^2) \end{aligned}$$

We calculate the partial derivatives of mse with respect to w and b as:

$$\begin{aligned} \frac{\partial mse(w,b)}{\partial w} &= \frac{1}{N}\sum_{i=1}^{N} 2w{x_i}^2 + 2x_ib - 2x_iy_i \\ &= \frac{2}{N}\sum_{i=1}^{N} x_i(wx_i + b - y_i) \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{\partial mse(w,b)}{\partial b} &= \frac{1}{N}\sum_{i=1}^{N} 2b + 2wx_i - 2y_i \\ &= \frac{2}{N}\sum_{i=1}^{N}(wx_i + b - y_i) \end{aligned} \tag{3}$$

Putting (1), (2) and (3) together, we get:

$$\nabla mse(w,b) = \begin{pmatrix} \frac{2}{N}\sum_{i=1}^{N} x_i(wx_i + b - y_i) \\ \frac{2}{N}\sum_{i=1}^{N}(wx_i + b - y_i) \end{pmatrix}$$

Vectorizing this, we get:

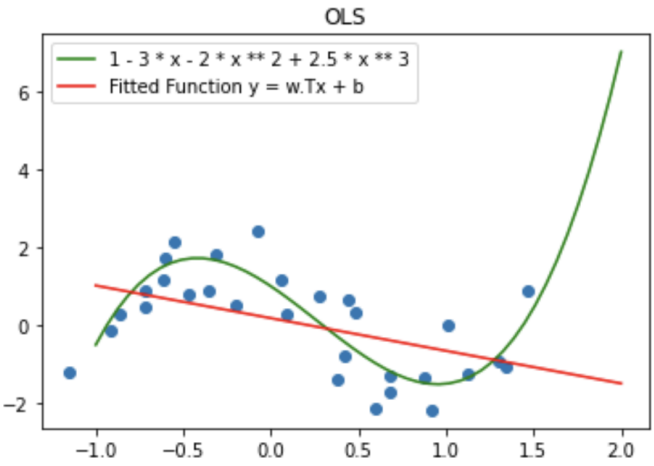
$$\nabla mse(w,b) = \begin{pmatrix} \frac{2}{N}X^T(W^TX + b - Y) \\ \frac{2}{N}\sum_{i=1}^{N}(wx_i + b - y_i) \end{pmatrix}$$

1.2 CS335: Lab

(c)

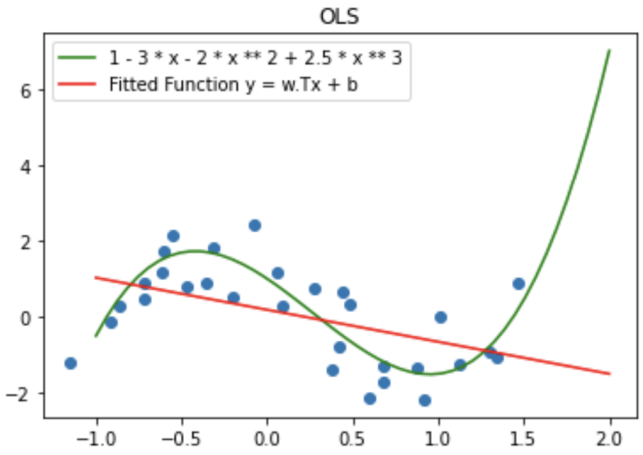
1) `singlevar_grad()` with `lr = 0.01` and `max_iter=500`

```
Validation loss is 3.8720507734475755
Training Loss loss is 1.1926998680458005
```



2) `singlevar_closedform()`

```
Validation loss is 3.8834706563093673
Training Loss loss is 1.1926891877346768
```



(d)

No, it is not possible to obtain a solution using `singlevar_grad()` such that its training loss is strictly less than that of the solution obtained by `singlevar_closedformd()`. When loss is defined as *mse*, loss value is the distance between $\hat{Y} = \Phi W$ and Y. The closed form solution minimizes the distance between $\hat{Y}$ and Y, so a loss value less than that cannot be obtained by any method.

However, if Φ is not full rank, in some cases, closed form solution, may not exist so gradient descent would be helpful. Here, since we have single variate Φ, closed form is optimal.

2 OLS and Ridge Regression

2.1 CS337: Theory

Let N be the number of samples each having d features. Given the feature matrix $X(N \times d$ dimensional matrix) the outputs Y (vector of size N) and W the weights to be learnt.

(a)

Predicted Outputs:

$$\hat{Y} = X \cdot W$$

If we include a bias term b, it would be d-dimensional vector such that predicted output:

$$\hat{Y} = X \cdot W + b$$

(b)

Let, X_i is d-dimensional vector forming one sample which are stacked up to form X as :

$$X = \begin{bmatrix} -X_1^T- \\ -X_2^T- \\ \dots \\ \dots \\ -X_N^T- \end{bmatrix}$$

For the minimum squared error loss function

$$\begin{aligned} \text{mse} &= \frac{1}{N}\sum_{i=1}^{N}(\hat{y_i} - y_i)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(X_i^T W - y_i)^2 \\ \frac{\partial mse(w,b)}{\partial W} &= \frac{2}{N}\big(\sum_{i=1}^{N} X_i(X_i^T W - y_i)\big) \end{aligned}$$

Vectorizing this expression, we get:

$$\boxed{\frac{\partial mse(w,b)}{\partial W} = \frac{2}{N}X^T \cdot (X \cdot W - Y)}$$

Here, $(X \cdot W - Y)$ is a N - dimensional vector as:

$$(X \cdot W - Y) = \begin{bmatrix} X_1^T W - y_1 \\ X_2^T W - y_2 \\ \dots \\ \dots \\ X_N^T W - y_N \end{bmatrix}$$

(c)

For the Ridge regression, we have an additional term $\lambda\|W\|^2$.

$$\begin{aligned} mse &= \frac{1}{N}\sum_{i=1}^{N}(\hat{y_i} - y_i)^2 + \lambda\|W\|^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(X_i^T W - y_i)^2 + \lambda\|W\|^2 \\ \frac{\partial mse(w,b)}{\partial W} &= \frac{2}{N}\big(\sum_{i=1}^{N} X_i(X_i^T W - y_i)\big) + 2\lambda W \end{aligned}$$

Vectorizing this expression, we get:

$$\boxed{\frac{\partial mse(w,b)}{\partial W} = \frac{2}{N}X^T \cdot (X \cdot W - Y) + 2\lambda W}$$

Here again, $(X \cdot W - Y)$ is a N - dimensional vector as:

$$(X \cdot W - Y) = \begin{bmatrix} X_1^T W - y_1 \\ X_2^T W - y_2 \\ \dots \\ \dots \\ X_N^T W - y_N \end{bmatrix}$$

(d)

There exists no solution for the closed form for OLS in the case when the columns of X are **not full rank** or in other words when $X^T X$ is not an invertible matrix.

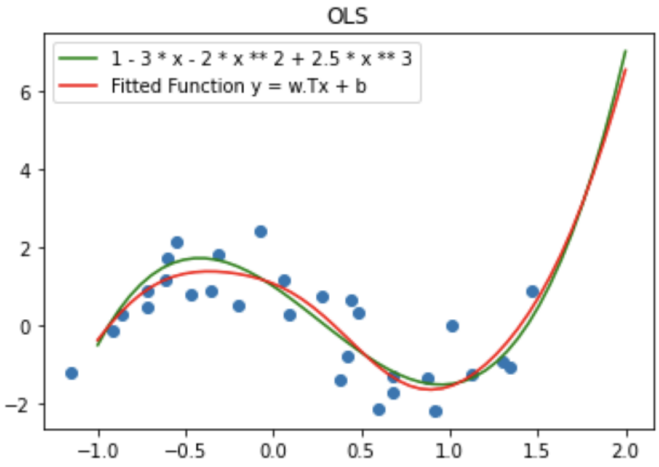
Yes, gradient descent does converge to a solution. The loss function may have multiple local minima apart from a global minima but even in that case, by adjusting the learning rate and number of epochs we can converge to an appropriate minima.

2.2 CS335: Lab

(b)

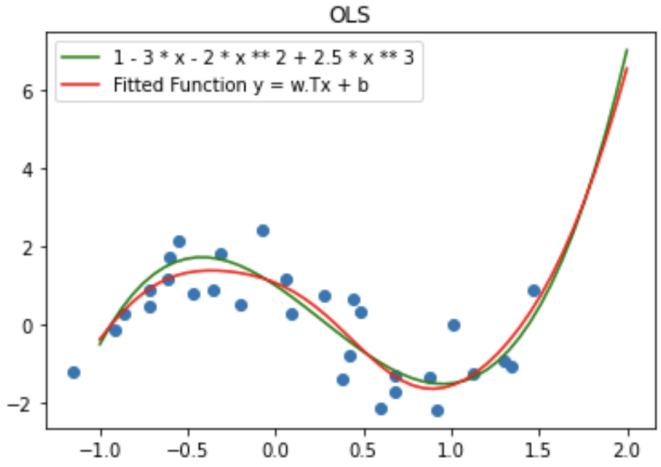
1) `multivar_grad()` with `lr = 0.1` and `max_iter=1200`

```
Validation loss if 0.8696916021563439
Training Loss loss if 0.45433024703559416
```



2) `multivar_closedform()`

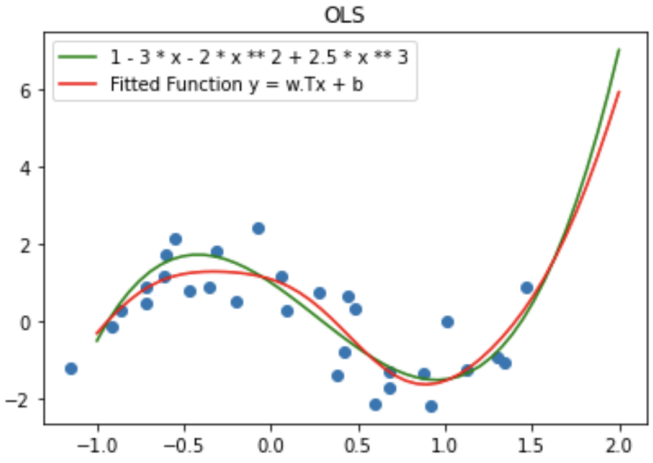
```
Validation loss if 0.8690010458418722
Training Loss loss if 0.4543297001460214
```



(c)

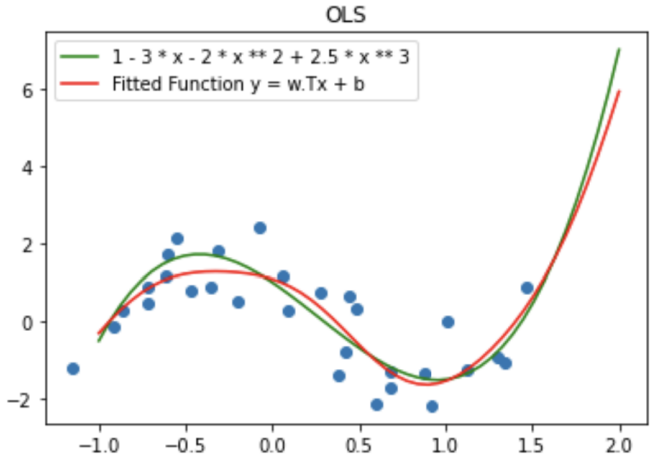
1) `multivar_reg_grad()` with `lr = 0.1`, `lamda = 0.005` and `max_iter=1000`

```
Validation loss if 0.89311128933525
Training Loss loss if 0.45956704941681725
```



2) `multivar_reg_closedform()` with `lamda = 0.005`

```
Validation loss if 0.8986517057890268
Training Loss loss if 0.4598367886351609
```



3 Bayesian Linear Regression

3.1 CS337: Theory

We consider dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \mathcal{R}$ and $y_i \in \mathcal{R}$. We need to learn parameter $w \in \mathcal{R}$ and bias $b = 0$.

(a)

We assume that w has a prior given by Gaussian distribution $\mathcal{N}(\mu_0, 1)$. So, $p(w) \sim \mathcal{N}(\mu_0, 1)$:

$$p(w) = \frac{1}{\sqrt{2\pi}} \exp\big(\frac{-(w-\mu_0)^2}{2}\big)$$

(c)

We are given that the data $\mathcal{D}$ is IID which means the following

$$\begin{aligned} p(y|x;w) &= p(\{y_1, \ldots, y_N\}|\{x_1, \ldots, x_N\}; w) \\ &= \prod_{i=1}^{N} p(y_i|x_i; w) \end{aligned}$$

Now, we are given $p(y|x;w) \sim \mathcal{N}(wx, 1)$. Hence, the above equation becomes the following

$$\begin{aligned} p(D|w) &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\big(\frac{-(y_i - wx_i)^2}{2}\big) \\ &= \frac{1}{(2\pi)^{N/2}} \exp\big(\frac{-\sum_{i=1}^{N}(y_i - wx_i)^2}{2}\big) \end{aligned}$$

(d)

We know that $p(D) = \int p(D|w)dw$ We know $p(D|w)$ and $p(w)$, so use the Bayes theorem we get $p(w|D)$ as:

$$\begin{aligned} p(w|D) &= \frac{p(D|w)p(w)}{p(D)} \\ &= \frac{\frac{1}{(2\pi)^{N/2}} \exp\big(\frac{-\sum_{i=1}^{N}(y_i - wx_i)^2}{2}\big) \times \frac{1}{\sqrt{2\pi}} \exp\big(\frac{-(w-\mu_0)^2}{2}\big)}{\int p(D|w)dw} \\ &= \frac{\frac{1}{(2\pi)^{(N+1)/2}} \exp\big(\frac{-\sum_{i=1}^{N}(y_i - wx_i)^2 - (w-\mu_0)^2}{2}\big)}{\int \left(\frac{1}{(2\pi)^{N/2}} \exp\big(\frac{-\sum_{i=1}^{N}(y_i - wx_i)^2}{2}\big)\right) dw} \end{aligned}$$

(h)

Taking only numerator of $p(w|D)$, we get:

$$p(w|D) \propto \exp\left(\frac{-1}{2}\left[\sum_i (y_i - wx_i)^2 + (w-\mu_0)^2\right]\right)$$

$$p(w|D) \propto \exp\left(\frac{-1}{2}\left[w^2(\sum_i x_i^2 + 1) - 2w(\sum_i y_i x_i + \mu_0) + \sum_i y_i^2 + \mu_0^2\right]\right) \tag{1}$$

We know that posterior has form $p(w|D) = N(\mu_N, \sigma_N^2)$

$$p(w|D) \propto \exp\left(-\frac{1}{2\sigma_N^2}(w-\mu_N)^2\right) = \exp\left(-\frac{1}{2\sigma_N^2}(w^2 - 2w\mu_N + \mu_N^2\right) \tag{2}$$

Using the (1) and (2), we compare coefficients of w and w^2 to get these relations:

$$\sum_i x_i^2 + 1 = \frac{1}{\sigma_N^2}$$

$$\sum_i y_i x_i + \mu_0 = \frac{\mu_N}{\sigma_N^2}$$

Hence, we get μ_N and σ_N^2 as follows:

$$\boxed{\mu_N = \frac{\sum_i y_i x_i + \mu_0}{\sum_i x_i^2 + 1}}$$

$$\boxed{\sigma_N^2 = \frac{1}{\sum_i x_i^2 + 1}}$$

These can be vectorized as -

$$\mu_N = (1 + X^T X)^{-1}(\mu_0 + X^T Y)$$
$$\sigma_N^2 = (1 + X^T X)^{-1}$$

(i)

As $N \to \infty$, the estimates change to the following:

$$\boxed{\mu_N = \frac{\sum_i y_i x_i}{\sum_i x_i^2}} \quad \text{or} \quad \boxed{\mu_N = (X^T X)^{-1} X^T Y}$$

$$\boxed{\sigma_N^2 = \frac{1}{\sum_i x_i^2}} \quad \text{or} \quad \boxed{\sigma_N^2 = (X^T X)^{-1}}$$

Here, limit of σ_N^2 can even be taken as $\sigma_N^2 \to 0$ as when $\sum_i x_i^2$ value becomes too large, probability factor will diminish and we can get close to exact estimates.

(j)

When $N \to \infty$ i.e. the size of our dataset becomes huge.
Posterior $\propto$ Prior $\times$ Data
It is safe to ignore the values given by prior as the observed data will dominate on the posterior value.

3.2 MLE Estimate

(a)

As already calculated:

$$p(D|w) = \frac{1}{(2\pi)^{N/2}} \exp\left(\frac{-\sum_{i=1}^{N}(y_i - wx_i)^2}{2}\right)$$

MLE estimate maximizes the data likelihood as:

$$w^* = argmax_w p(D|w)$$

We can maximize the *log* of data likelihood too as it is an increasing function:

$$\begin{aligned}
w^* &= argmax_w \log p(D|w) \\
&= argmax_w \log \frac{1}{(2\pi)^{N/2}} \exp\left(\frac{-\sum_{i=1}^{N}(y_i - wx_i)^2}{2}\right) \\
&= argmax_w -\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{N}(y_i - wx_i)^2 \\
&= argmax_w -\frac{1}{2}\sum_{i=1}^{N}(y_i - wx_i)^2 \\
&= argmax_w -\sum_{i=1}^{N}(y_i - wx_i)^2 \\
&= argmin_w \sum_{i=1}^{N}(y_i - wx_i)^2
\end{aligned}$$

Hence, we can see that w^* from MLE Estimate is same as Least Square Solution:

$$w^* = argmin_w \sum_{i=1}^{N}(y_i - wx_i)^2$$
$$w^* = (X^T X)^{-1} X^T Y$$

(b)

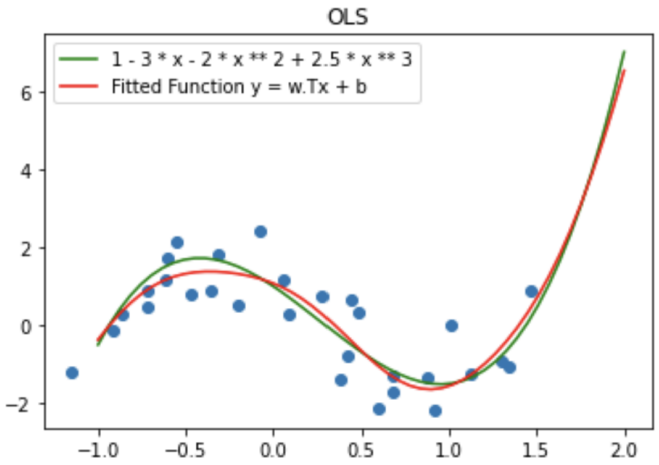
When the data $\to \infty$, then both the MLE and Bayesian estimates converge to same solutions. This is because the data would dominate heavily from the prior and as Bayesian posterior $\propto$ prior $\times$ data, posterior would also converge to MLE Estimate solution which maximized the likelihood of data.

3.3 CS335: Lab

(a)

`bayesion_lr()` with $\Sigma_0 = 0.5$, $\sigma = 0.2$ and $\mu_0 = 0$

```
Validation loss if 0.923096436384142
Training Loss loss if 0.4543741122058695
```



(b)

When the X^TX is not invertible matrix, MLE estimate fails to give a solution. This problem does not come up when the Bayesian regression because it also involves taking in account the prior distribution so $\sum_N$ is estimated as follows:

$$\sum_N = (\sum_0^{-1} + \frac{1}{\sigma^2} X^T X)^{-1}$$

Therefore, even if the inverse of X^TX does not exist, we have the covariance matrix $\sum_0^{-1}$ which is always symmetric positive semi definite and thus, invertible. This will give us an overall solution for the bayesian regression.

Bayesian closed form will always exist as the expected value of the pdf will always exist.

4 Conclusion

Single Variable Linear Regression has a high training and test error and does not converge to the solution as the distribution here $f(x)$ is a polynomial function. So, we only compare Multi Variable Linear Regression (with and without regularization) and Bayesian Linear Regression and their gradient descent and closed form versions.

4.1 Comparison in terms of running time

Closed form VS Gradient descent

For smaller datasets, if closed form solution exists, the running time is generally less than gradient descent. Here, since the number of data points are only 50, and closed form solution exists, they run faster than gradient descent.

However, since closed form involves taking inverse of matrix $X^T X$ or something similar, if the number of data points increases, this will be a heavy computation and the memory taken will also be quite large. So, we use gradient descent in cases where dataset is bigger.

Ordinary(Multivariate) VS Ridge VS Bayesian

For multivariate data, running time of ordinary and ridge regression will mostly be comparable. Ridge regression might become a little more time consuming but will fit the data better, if it is noisy. Here the noise is not much so multivariate is preferred is running time is a concern. Bayesian estimate for a small dataset like ours is also quite fast, and performs the best so, Bayesian is even better than multivariate.

4.2 Comparison in terms of convergence

Closed form VS Gradient descent

If closed form solution exists, both the methods always converge to the optimal solution.

Ordinary(Multivariate) VS Ridge VS Bayesian

When the split for training data is reasonable i.e. (> 0.6), all 3 methods can converge similarly with comparable bias and variance as shown in the plots above. Here, since the function $f(x)$ is polynomial, between multivariate and ridge regression, multivariate performs better.

However, when train split reduces to < 0.3, bayesian regression outperforms the others. In problems where we have limited data or have some prior knowledge that we want to use in our model, the Bayesian Linear Regression approach can both incorporate prior information and show our uncertainty.

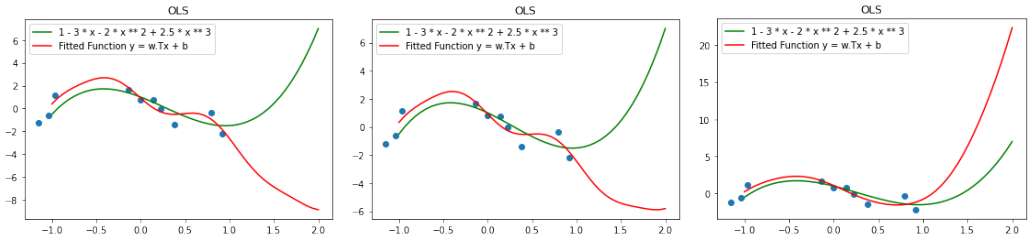


Figure 1: (a)Ordinary Multivariate, (b)Ridge, (c) Bayesian

Final thoughts

Since, the overall dataset is small, we cannot comment with certainty as to which approach works better in all cases. All the approaches converge to solutions more or less in a similar fashion. Since, function $f(x)$ to be fitted is polynomial, multivariate (without regularization) would be preferred. if we do have prior knowledge available, bayesian approach is the best for a small dataset.