Assignment 1

Devansh Jain, 190100044

29 Aug 2021

Contents

1 Ordinary Least Squares (OLS) Regression in one variable

Notation:
$\mathbf{X}$ is $N \times 1$ vector,
$\mathbf{Y}$ is $N \times 1$ vector,
$\mathbf{1_N}$ is $N \times 1$ vector (all 1s),
x_i is scalar (i^{th} observed sample),
y_i is scalar (i^{th} observed output),
w is scalar,
b is scalar.

1.1 CS337: Theory

$$
\begin{aligned}
mse(w,b) &= \frac{1}{N}\sum_{i=1}^{N}((wx_i+b)-y_i)^2 \\
\frac{\partial mse(w,b)}{\partial w} &= \frac{1}{N}\sum_{i=1}^{N} 2\ ((wx_i+b)-y_i)\ x_i \\
&= \frac{2}{N}\sum_{i=1}^{N}(wx_i+b-y_i)\ x_i \\
&= \frac{2}{N}\ \mathrm{dot}(w\mathbf{X}+b\mathbf{1_N}-\mathbf{Y}),\mathbf{X}) \\
\frac{\partial mse(w,b)}{\partial b} &= \frac{1}{N}\sum_{i=1}^{N} 2\ ((wx_i+b)-y_i) \\
&= \frac{2}{N}\sum_{i=1}^{N}(wx_i+b-y_i) \\
&= \frac{2}{N}\ \mathrm{sum}(w\mathbf{X}+b\mathbf{1_N}-\mathbf{Y}) \\
\nabla mse(w,b) &= \begin{pmatrix} \frac{\partial mse(w,b)}{\partial w} \\ \frac{\partial mse(w,b)}{\partial b} \end{pmatrix} \\
&= \frac{2}{N}\begin{pmatrix} \mathrm{dot}(w\mathbf{X}+b\mathbf{1_N}-\mathbf{Y}),\mathbf{X}) \\ \mathrm{sum}(w\mathbf{X}+b\mathbf{1_N}-\mathbf{Y}) \end{pmatrix}
\end{aligned}
$$

1.2 CS335: Lab

(a)

Code for the function `split_data()` updated in notebook.

(b)

Code for the function `mse_single_var()` updated in notebook.

(c)

Code for the functions `singlevar_grad()` and `singlevar_closedform()` updated in notebook.

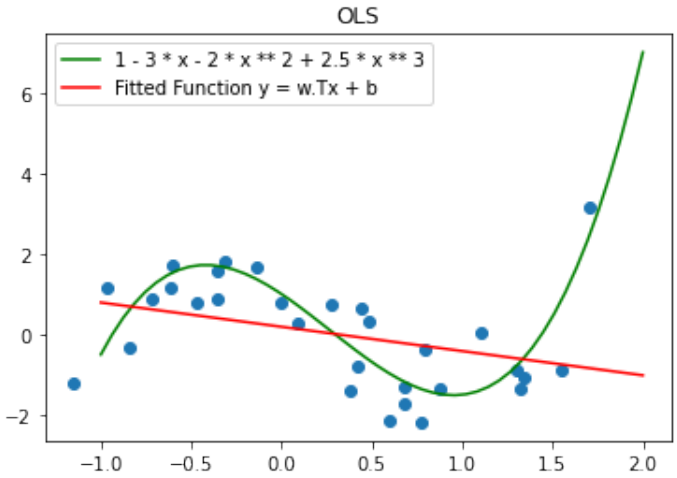

Figure 1: Single Variable Regression - Gradient Descent (`epochs`=1000, `lr`=1e-2)

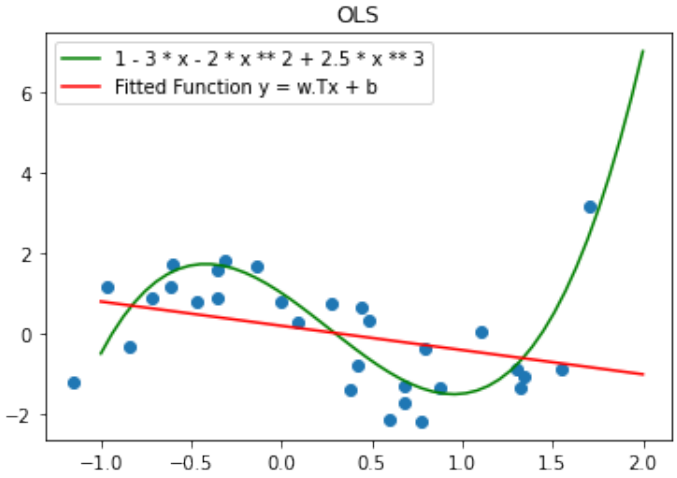

Figure 2: Single Variable Regression - Closed Form

(d)

As mentioned in description of the question (point 5), the error function (mse) is convex and has just one minimum.
By using the closed form, we get the point corresponding to minimum mse error. As we compute the parameters based on training data, the training loss is minimum at this point.
Therefore, training loss for solution of `singlevar_grad()` can never be strictly less than that of the solution obtained by `singlevar_closedform()`.

2 OLS and Ridge Regression

To avoid confusion in lab part where we should not regularize the bias term, I am considering the bias to be separate for now, i.e. apart from W the weights to be learnt, we also have to learn b the bias term.

Notation:
$\mathbf{X}$ is $N \times d$ matrix,
$\mathbf{Y}$ is $N \times 1$ vector,
$\hat{\mathbf{Y}}$ is $N \times 1$ vector,
$\mathbf{1_N}$ is $N \times 1$ vector (all 1s),
$\mathbf{W}$ is $d \times 1$ vector,
$\mathbf{x_i}$ is $d \times 1$ vector (i^{th} observed sample),
y_i is scalar (i^{th} observed output),
w_i is scalar (weight for i^{th} feature),
b is scalar.

2.1 CS337: Theory

(a)

$\hat{\mathbf{Y}} = \mathbf{XW} + b\mathbf{1_N}$, where $\mathbf{W}$ and b are learnt from the given N samples (which may be noisy).

(b)

Assuming that partial derivative is $1 \times d$ row vector (numerator layout).

$$
\begin{aligned}
mse(\mathbf{W}, b) &= \frac{1}{N}\sum_{i=1}^{N}((\mathbf{W}^T\mathbf{x_i} + b) - y_i)^2 \\
\frac{\partial mse(\mathbf{W}, b)}{\partial \mathbf{W}} &= \frac{1}{N}\sum_{i=1}^{N} 2\ ((\mathbf{W}^T\mathbf{x_i} + b) - y_i)\ \mathbf{x_i}^T \\
&= \frac{2}{N}\sum_{i=1}^{N}(\mathbf{W}^T\mathbf{x_i} + b - y_i)\ \mathbf{x_i}^T \\
&= \frac{2}{N}\ \text{dot}(\text{dot}(\mathbf{X}, \mathbf{W}) + b\mathbf{1_N} - \mathbf{Y}, \mathbf{X}) \\
\frac{\partial mse(\mathbf{W}, b)}{\partial b} &= \frac{1}{N}\sum_{i=1}^{N} 2\ ((\mathbf{W}^T\mathbf{x_i} + b) - y_i) \\
&= \frac{2}{N}\sum_{i=1}^{N}(\mathbf{W}^T\mathbf{x_i} + b - y_i) \\
&= \frac{2}{N}\ \text{sum}(\text{dot}(\mathbf{X}, \mathbf{W}) + b\mathbf{1_N} - \mathbf{Y})
\end{aligned}
$$

(c)

Assuming that partial derivative is $1 \times d$ row vector (denominator layout).

$$
\begin{aligned}
mse(\mathbf{W}, b) &= \lambda ||\mathbf{W}||_2^2 + \frac{1}{N}\sum_{i=1}^{N}((\mathbf{W}^T\mathbf{x_i} + b) - y_i)^2 \\
\frac{\partial mse(\mathbf{W}, b)}{\partial \mathbf{W}} &= 2\lambda\mathbf{W}^T + \frac{1}{N}\sum_{i=1}^{N} 2\ ((\mathbf{W}^T\mathbf{x_i} + b) - y_i)\ \mathbf{x_i}^T \\
&= 2\lambda\mathbf{W}^T + \frac{2}{N}\sum_{i=1}^{N}(\mathbf{W}^T\mathbf{x_i} + b - y_i)\ \mathbf{x_i}^T \\
&= 2\lambda\mathbf{W}^T + \frac{2}{N}\ \mathrm{dot}(\mathrm{dot}(\mathbf{X}, \mathbf{W}) + b\mathbf{1_N} - \mathbf{Y}, \mathbf{X}) \\
\frac{\partial mse(\mathbf{W}, b)}{\partial b} &= \frac{1}{N}\sum_{i=1}^{N} 2\ ((\mathbf{W}^T\mathbf{x_i} + b) - y_i) \\
&= \frac{2}{N}\sum_{i=1}^{N}(\mathbf{W}^T\mathbf{x_i} + b - y_i) \\
&= \frac{2}{N}\ \mathrm{sum}(\mathrm{dot}(\mathbf{X}, \mathbf{W}) + b\mathbf{1_N} - \mathbf{Y})
\end{aligned}
$$

(d)

Closed form of OLS is $\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.
This closed form is not applicable when $\mathbf{X}^T\mathbf{X}$ is not invertible (is singular).
This happens when $\mathbf{X}$ is not full column rank, i.e. the columns (features) of $\mathbf{X}$ are linearly dependent.
We don't have a unique solution and therefore can't have a closed form.
We actually have infinite solutions (of $\mathbf{W}$).
(We can't have no solutions of OLS as it is a minimization problem)
Gradient descent therefore would converge towards these solutions.
(Depending on the starting point, it would descent to one of these solutions)

2.2 CS335: Lab

(a)

Code for the functions `mse_multi_var()` and `mse_regularized()` updated in notebook.

(b)

Code for the functions `multivar_grad()` and `multivar_closedform()` updated in notebook.

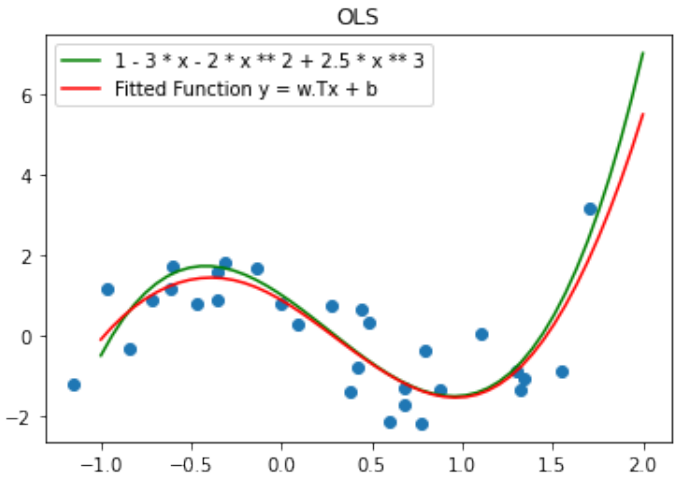

Figure 3: Multi Variable Regression - Gradient Descent (`epochs`=1000, `lr`=1e-1)

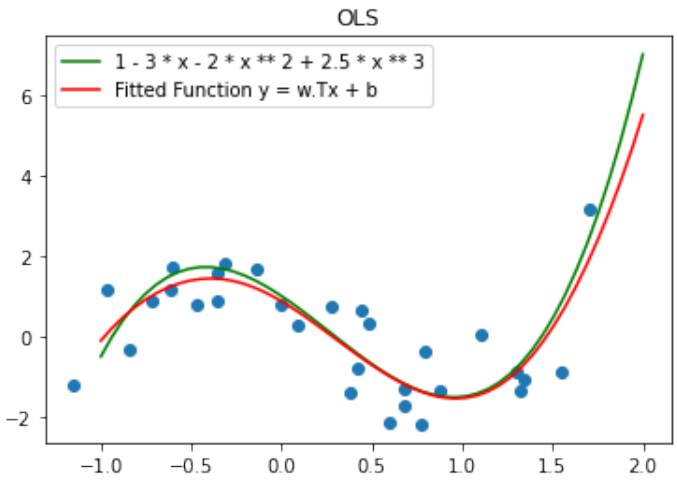

Figure 4: Multi Variable Regression - Closed Form

(c)

Code for the functions `multivar_reg_grad()` and `multivar_reg_closedform()` updated in notebook.

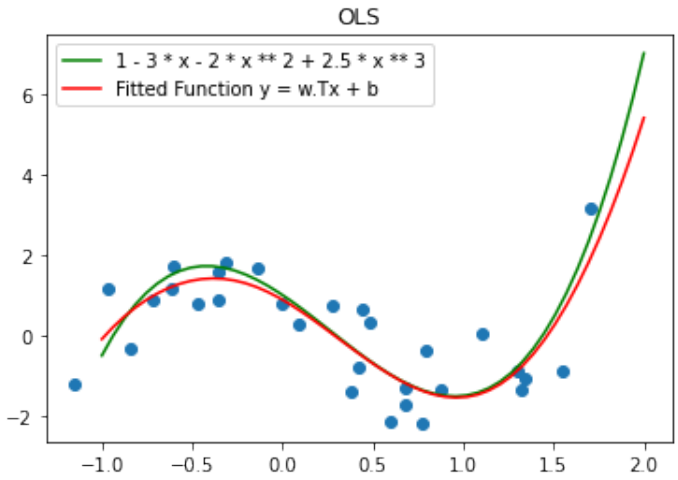

Figure 5: Regularization - Gradient Descent (`epochs`=1000, `lr`=1e-1, `lamda`=0.001)

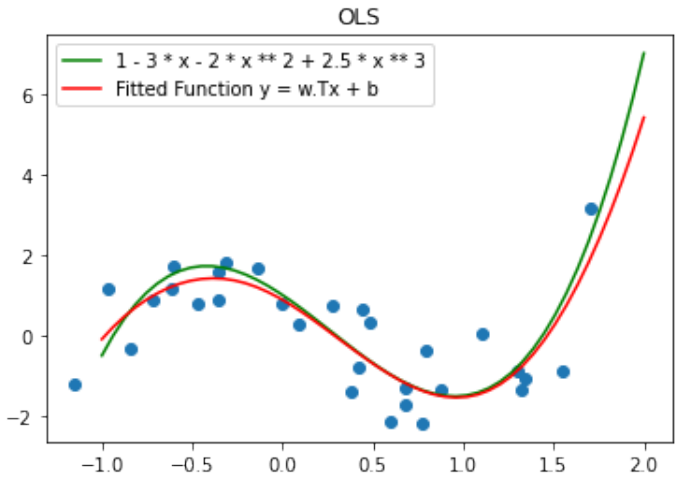

Figure 6: Regularization - Closed Form (`lamda`=0.001)

3 Bayesian Linear Regression

3.1 CS337: Theory

(a)

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

(b)

$$p(y|x; w) = \frac{1}{\sqrt{2\pi}} \exp\left(- \frac{(y - wx)^2}{2} \right)$$

(c)

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

(d)

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

(e)

$$p(w|\mathcal{D}) \propto \exp\left(- \frac{(w - \mu_0)^2 + \sum_{i=1}^{N} (y_i - wx_i)^2}{2} \right)$$

(f)

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

(g)

$$\begin{aligned} p(w|\mathcal{D}) &= \mathcal{N}(\mu_N, \sigma_N^2) \propto \exp\left(- \frac{(w - \mu_N)^2}{2\sigma_N^2} \right) \\ p(w|\mathcal{D}) &\propto \exp\left(- \frac{w^2(\frac{1}{\sigma_N^2}) - 2w(\frac{\mu_N}{\sigma_N^2}) + (\frac{\mu_N^2}{\sigma_N^2})}{2} \right) \end{aligned} \tag{2}$$

(h)

Comparing (1) and (2),

$$\sigma_N^2 = \frac{1}{1+\sum_i x_i^2}$$
$$\mu_N = (\mu_0 + \sum_i x_i y_i)\sigma_N^2$$
$$= \frac{\mu_0 + \sum_i x_i y_i}{1+\sum_i x_i^2} \qquad (= w_{BE}^*)$$

(i)

As $N \to \infty$, $\sum_i x_i^2 \gg 1$ and $\sum_i x_i y_i \gg \mu_0$.
$\sigma_N^2 \to \frac{1}{\sum_i x_i^2} \to 0$ (x_is are iid, so $\sum_i x_i^2$ can never be convergent)
$\mu_N \to \frac{\sum_i x_i y_i}{\sum_i x_i^2} = \frac{\sum_i x_i(w_{\text{true}} x_i + \epsilon_i)}{\sum_i x_i^2} \to w_{\text{true}}$ ($\epsilon_i \sim \mathcal{N}(0,1)$ and $E[y_i] = w_{\text{true}} x_i$)

(j)

As we sample more and more data, we get more confident about the estimated value of w, and thus the variance (σ_N^2) tends to zero, i.e. our estimate for w is converging.
Also, we converge towards the true value as data starts to dominate over the prior information, thus the mean (μ_N) tends towards the w_{true}.

3.2 MLE Estimate

(a)

$$w^* = \arg\max_w \; p(\mathcal{D}; w)$$
$$= \arg\max_w \; (2\pi)^{-N/2} \exp\left(- \frac{\sum_{i=1}^N (y_i - w x_i)^2}{2} \right)$$
$$= \arg\max_w \; \exp\left(- \frac{\sum_{i=1}^N (y_i - w x_i)^2}{2} \right)$$
$$= \arg\max_w \; - \frac{\sum_{i=1}^N (y_i - w x_i)^2}{2} \qquad \text{(ln is monotonically increasing function)}$$
$$= \arg\min_w \; \sum_{i=1}^N (y_i - w x_i)^2 \qquad \text{(-2x is monotonically decreasing function)}$$

$$\frac{d}{dw}\left(\sum_{i=1}^N (y_i - w x_i)^2 \right)\Big|_{w=w^*} = 0 \implies 2\left(\sum_{i=1}^N (y_i - w^* x_i) x_i \right) = 0$$
$$w^* = \frac{\sum_i x_i y_i}{\sum_i x_i^2} \; (= w_{MLE}^*)$$

(b)

$\lim_{N\to\infty} w^*_{BE} = w^*_{MLE} = \dfrac{\sum_i x_i y_i}{\sum_i x_i^2}$.
For infinite data, the prior becomes insignificant.
Maximum Likelihood Estimate (MLE) and Bayesian Estimate (BE) are equivalent and both converge towards the true value w_{true}.

3.3 CS335: Lab

(a)

Code for the function `bayesion_lr()` updated in notebook.

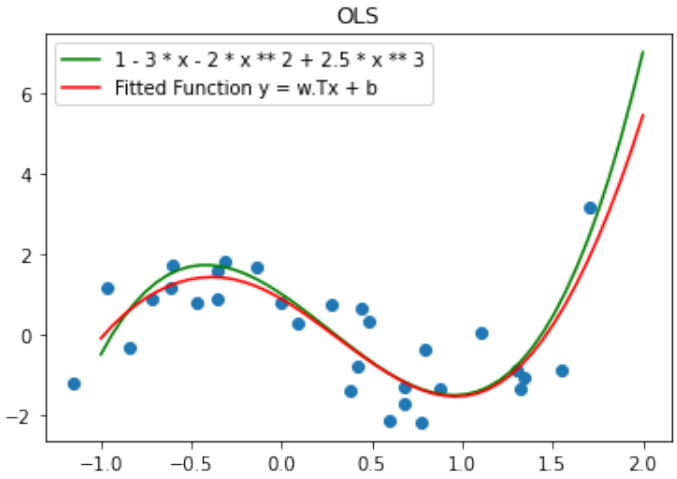

Figure 7: Bayesian Linear Regression - Closed Form (`sigma`=0.1)

(b)

In Closed form of OLS, we might have to invert a non-invertible matrix. X^TX would be non-invertible when features aren't linearly independent.
Here, we take inverse of $X^TX + \lambda I$ (reminds of ridge regression). Even if we have features which aren't linearly independent, $X^TX + \lambda I$ would be invertible due to presence of I which is a full rank matrix.

4 Conclusion

Comparing the seven methods for the assignment problem:

1. **Single Variable OLS Regression Gradient Descent**
   Gradient Descent can be slow;
   Need to set hyper-parameters - epochs and learning rate;
   Not a nice fit (Validation error approx 0.14)

2. **Single Variable OLS Regression Closed Form**
   Closed form computation is fast;
   No hyper-parameters;
   Not a nice fit (Validation error approx 0.14)

3. **Multi Variable OLS Regression Gradient Descent**
   Gradient Descent can be slow;
   Need to set hyper-parameters - epochs and learning rate;
   Nice fit (Validation error approx 0.03);
   Can give solution when the features aren't linearly independent

4. **Multi Variable OLS Regression Closed Form**
   Closed form computation is fast;
   No hyper-parameters;
   Nice fit (Validation error approx 0.03);
   Can't give solution when the features aren't linearly independent

5. **Multi Variable Regularized (Ridge) Regression Gradient Descent**
   Gradient Descent can be slow;
   Need to set hyper-parameters - lambda, epochs and learning rate;
   Nice fit (Validation error approx 0.03);
   Can give solution when the features aren't linearly independent

6. **Multi Variable Regularized (Ridge) Regression Closed Form**
   Closed form computation is fast;
   Need to set hyper-parameter - lambda;
   Nice fit (Validation error approx 0.03);
   Can give solution when the features aren't linearly independent

7. **Multi Variable Bayesian Regression Closed Form**
   Closed form computation is fast;
   Need to set hyper-parameter - prior and sigma;
   Nice fit (Validation error approx 0.03);
   Can give solution when the features aren't linearly independent

Only method which doesn't require hyper-parameters and gives a good fit is Method 4 - Multi Variable OLS Regression Closed Form. However, it fails to give a solution when the features aren't linearly independent.

Now that we have to set some hyper-parameter, we can decide the method based on number of features.
In all closed forms, we compute inverse of a $d \times d$ matrix. The complexity of this computation is $\mathcal{O}(d^3)$. Thus, this method might be infeasible for large number of features.

For large number of features, if the features are known to be linearly independent then we can use Method 3 - Multi Variable OLS Regression Gradient Descent, else we should prefer Method 5 - Multi Variable Regularized (Ridge) Regression Gradient Descent (this would remove redundant features).

For small number of features (like in this assignment), closed form solutions would be faster and more accurate. If we have some prior for the parameter then we can use Method 7 - Multi Variable Bayesian Regression Closed Form or we can try different lambdas for Method 6 - Multi Variable Regularized (Ridge) Regression Closed Form.

For this assignment, I prefer Method 6 - Multi Variable Regularized (Ridge) Regression Closed Form.