Homework 2

Kyle Hadley

In [1]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
import warnings
warnings.simplefilter('ignore')
```

Note: Worked with Joaquin Santecchia on this homework assignment.

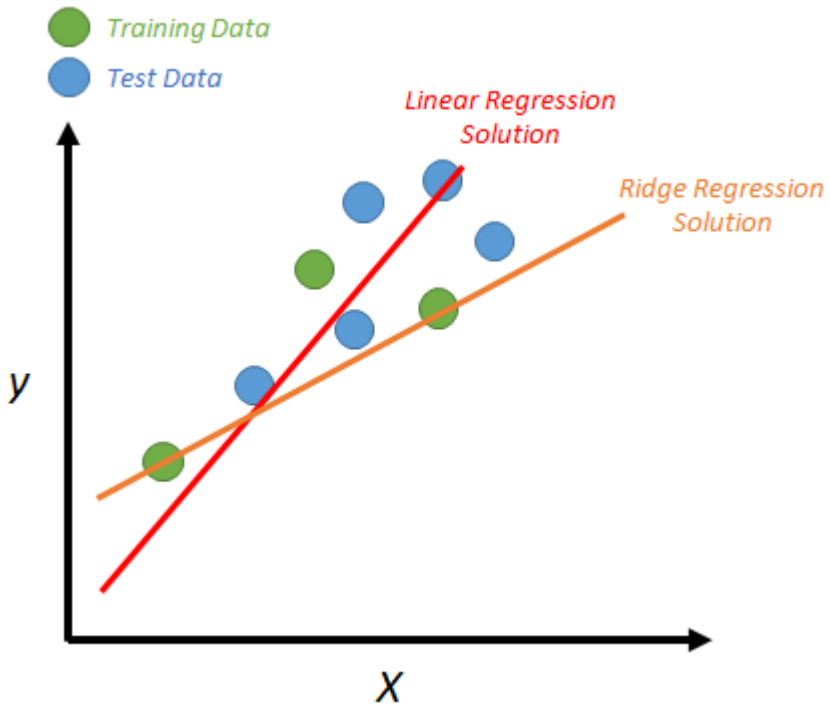
1. Ridge Regression

Given our equation for linear regression of $min_w \|Xw - y\|_2^2$ and our equation for ridge regression of $min_w \|Xw - y\|_2^2 + \frac{\eta}{2}\|w\|_2^2$.

(a)

For parts (a) and (b) I've generated a dataset (X, y) such that the training data is equivalent across both examples, but the test data (and thus the entire dataset of X) varies.

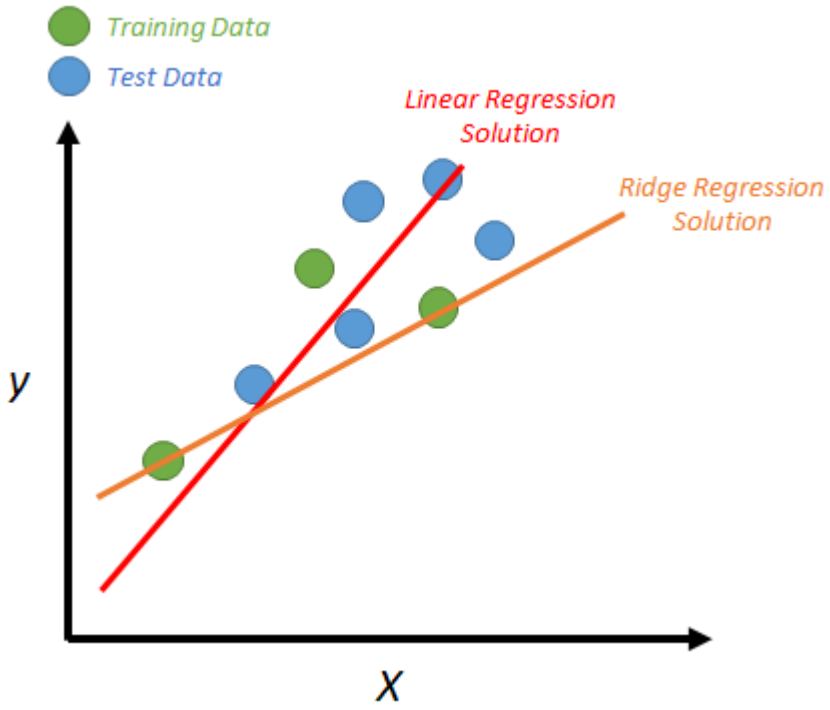
Given the dataset as shown below, we can see that in this instance the linear regression solution is preferred for our dataset X than the ridge regression model.



In this situation, while the linear regression solution has a high slope due to the nature of our training data, the overfitting to the training data matches (low variance) with the total dataset which includes the test data.

(b)

Given the dataset as shown below, we can see that in this instance the ridge regression solution is preferred for our dataset X than the linear regression model.



In this situation, the linear regression is too biased towards our initial training data and overfits with respect to this training data. The ridge regression solution provides a bias that attempts to resolve the issue of overfitting to the training data.

(c)

The regularization hyper-parameter η impacts the fit of the optimization model for a given problem. If η is significantly large, then the model will underfit resulting in high bias and low variance. In contrast, if η is small, then the model will overfit resulting in a low bias and high variance.

(d)

To solve the closed form solution, we can set the gradient of our objective function $F(w)$ equal to zero and then solve for w. We can write our $F(w)$ as,

$$F(w) = \|Xw - y\|_2^2 + \frac{\eta}{2}\|w\|_2^2 = \frac{1}{n}(Xw - y)^T(Xw - y) + \frac{\eta}{2n}w^T w$$

Taking the derivative of $F(w)$ and setting it equal to zero we get the relationship,

$$\frac{\partial F(w)}{\partial w} = 0 = \frac{2}{n} X^T (Xw - y) + \frac{\eta}{n} w$$

Now solving for w we can find our closed form solution.

$$0 = \frac{2}{n} X^T (Xw - y) + \frac{\eta}{n} w$$

$$0 = X^T (Xw - y) + \frac{\eta}{2} w$$

$$0 = X^T Xw - X^T y + \frac{\eta}{2} w$$

$$w \left(X^T X + \frac{\eta}{2} \right) = X^T y$$

$$w = \left(X^T X + \frac{\eta}{2} \right)^{-1} X^T y$$

(e)

(1) When the columns of X are more than the rows, i.e. $m > n$, we are unable to compute the closed-form solution of the vanilla linear regression because the term $X^T X$ will no longer be invertible. Thus, we will unable to solve for our weights (w). However, in contrast the ridge regression solution can be found.

(2) When the columns of X are highly correlated, we are unable to compute a closed-form solution of the vanilla linear regression. If we have two variables that are nearly identical (or identical in the extreme case) we are unable to generate a linear model that can account for both of these as inputs to our output y.

(f)

(i)

Given,

$$F(w, w_0) = \|Xw + w_0 - y\|_2^2 + \frac{\eta}{2} \|w\|_2^2 = \frac{1}{n} (Xw + w_0 - y)^T (Xw + w_0 - y) + \frac{\eta}{2n} w^T w$$

First, we can take the derivative of $F(W)$ with respect to w_0 to solve for w_0 and set equal to zero.

$$\frac{\partial F(w, w_0)}{\partial w_0} = 0 = \frac{2}{n} (Xw + w_0 - y)$$

$$0 = (Xw + w_0 - y)$$

$$w_0 = y - Xw$$

Taking the derivative of $F(w)$ and setting it equal to zero we get the relationship,

$$\frac{\partial F(w)}{\partial w} = 0 = \frac{2}{n} X^T (Xw + w_0 - y) + \frac{\eta}{n} w$$

Now solving for w with a substitution for w_0 we can find our closed form solution.

$$0 = \frac{2}{n} X^T (Xw + (y - Xw) - y) + \frac{\eta}{n} w$$

$$0 = \frac{2}{n} X^T I + \frac{\eta}{n} w$$

$$0 = X^T + \frac{\eta}{2} w$$

$$w = \frac{2}{\eta} X^T$$

(ii)

If we include a penalty for our offset parameter, then our model would be dependent on the origin chosen for our data. Adding a constant c to each of the target y_i would not simply result in a shift of the predictions by the same amount c - which is an effect we would want. I.e. if we were to modify the origin of our data, having a penalized offset parameter would mean that it wouldn't account for this shift in the origin correctly.

There are other benefits of having no penalty for the offset parameter such as the average value of y_i and the average value of $\hat{y}$ being equal. If we penalize the intercept, then this relationship would no longer hold true.

2. Bias and Variance

(a)

Given that $\tilde{\theta} = (X^T X)^{-1} X^T \vec{y}$ and $\vec{y} = X\theta + \vec{\epsilon}$, we can combine the relationships such that

$$\tilde{\theta} = (X^T X)^{-1} X^T \vec{y} = (X^T X)^{-1} X^T (X\theta + \vec{\epsilon})$$

$$\tilde{\theta} = (X^T X)^{-1} X^T X\theta + (X^T X)^{-1} X^T \vec{\epsilon}$$

$$\tilde{\theta} = \theta + (X^T X)^{-1} X^T \vec{\epsilon}$$

From this relationship, we can see that the model is unbiased as long as (i) X is fixed (i.e. non-stochastic) so that

$$E[\tilde{\theta}] = E[\theta] + E[(X^T X)^{-1} X^T \epsilon]$$

$$E[\tilde{\theta}] = \theta + (X^T X)^{-1} X^T E[\epsilon]$$

where $E[\epsilon] = 0$ by assumption or (ii) X is stochastic by independent of ϵ such that

$$E[\tilde{\theta}] = E[\theta] + E[(X^T X)^{-1} X^T \epsilon]$$

$$E[\tilde{\theta}] = \theta + (X^T X)^{-1} E[X^T \epsilon]$$

were $E[X^T \epsilon] = 0$.

(b)

Note: Did not attempt

3. Programming Problem: Linear Algebra

(a)

By observation, it seems that the most correlated features to the house price (MEDV) are LSTAT, RM and B.

Note: As discussed in part (b), we can see that actually the most correlated features are RM, LSTAT, and PTRATIO.

(b)

On reviewing the heatmap of seaborn, the most correlated features to the house price (MEDV) are RM, LSTAT, and PTRATIO - which do not directly align with what was observed initially in part (a). However, this misalignment is noted in part(a).

(c)

After running both the linear and ridge regression on the randomly train-test split training set, these are the obtained coefficients w.

Note: The reported coefficients for ridge regression used an $\eta = 15$.

Coefficients (w)	**Linear**	**Ridge**
CRIM	-0.099324	-0.104002
ZN	0.052251	0.052225
INDUS	0.004516	0.037553
CHAS	2.957261	2.689784
NOX	1.127938	-5.698645
RM	5.854198	6.144848
AGE	-0.014957	-0.008247
DIS	-0.920844	-0.989810
RAD	0.159519	0.163495
TAX	-0.008934	-0.007848
PTRATIO	-0.435674	-0.424732
B	0.014905	0.015801
LSTAT	-0.474751	-0.443902

(d)

After calculating the RMSE for both Linear and Ridge regression, against both training and test dataset, the resulting RMSE values are outputted in the table below.

	Linear	Ridge
Training Set	4.8206	4.8419
Test Set	5.2092	5.1474

We can see that the RMSE for the linear regression is lower than the RMSE for ridge regression when evaluating against the training set, but is higher for the test set. It could be likely that given these results, the linear regression model is overfitting the training data and thus has a higher error when we evaluate against a test set. While the ridge regression model is not as accurate for the given training set, we can see that it is more accurate when evaluated aginst a dataset not used for training.

In both cases, the RMSE is greater for the test set than the training set. This may be expected because models will inherently be more accurate within the training dataset, as the training dataset is what is used to calculate our weights within the model. Inherently, a model will natural "overfit" to the training data - the goal is strike a balance between over and underfitting.

(e)

After training both a linear and ridge regression model against only the top-3 features, we get the following RMSE values as shown in the table below.

	Linear	Ridge
Training Set	5.2734	5.2739
Test Set	5.4947	5.4880

Given the results, we can see that using only the top-3 features results in higher RMSE values in all aspects (linear v. ridge and training v. test). The difference between the RMSE value isn't insignificant (an increase in ~10% error), but the increase may be considered reasonable in order to reduce the complexity of the model. Being able to reduce the number of inputs for a model from 13 to 3 could significantly increase computational efficiency when the dataset expands or other limitations exist.

(f)

Note: Did not attempt

4. Programming Problem: Logistic Regression

(a)

Given $F(W) = \frac{1}{n}\sum_{i=1}^{n} -\log[Pr(y = y_i|x = X_i; W)] + \frac{\eta}{2}\|W\|_F^2$, we can solve for our gradient, i.e. $\frac{\partial F(W)}{\partial W}$.

We can first re-write our loss function as the following,

$$F(W) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_n * \log(e^z) - \log\left(\sum_{j=1}^{c} e^{z_j}\right)\right] + \frac{\eta}{2}\|W\|_F^2$$

$$F(W) = \frac{1}{n}\sum_{i=1}^{n}\left[-y_n * z + \log\left(\sum_{j=1}^{c} e^{z_j}\right)\right] + \frac{\eta}{2}\|W\|_F^2$$

Now we can solve for our derivative such that

$$\frac{\partial F(W)}{\partial W} = \frac{1}{n}\sum_{i=1}^{n}\left[-y_n * X_n + \frac{e^z}{\sum_{j=1}^{c} e^{z_j}} X_n\right] + \eta W$$

$$\frac{\partial F(W)}{\partial W} = \frac{1}{n}\sum_{i=1}^{n}\left[X_n(Pr(y = k|X; W) - y_n)\right] + \eta W$$

Our gradient descent rule will be,

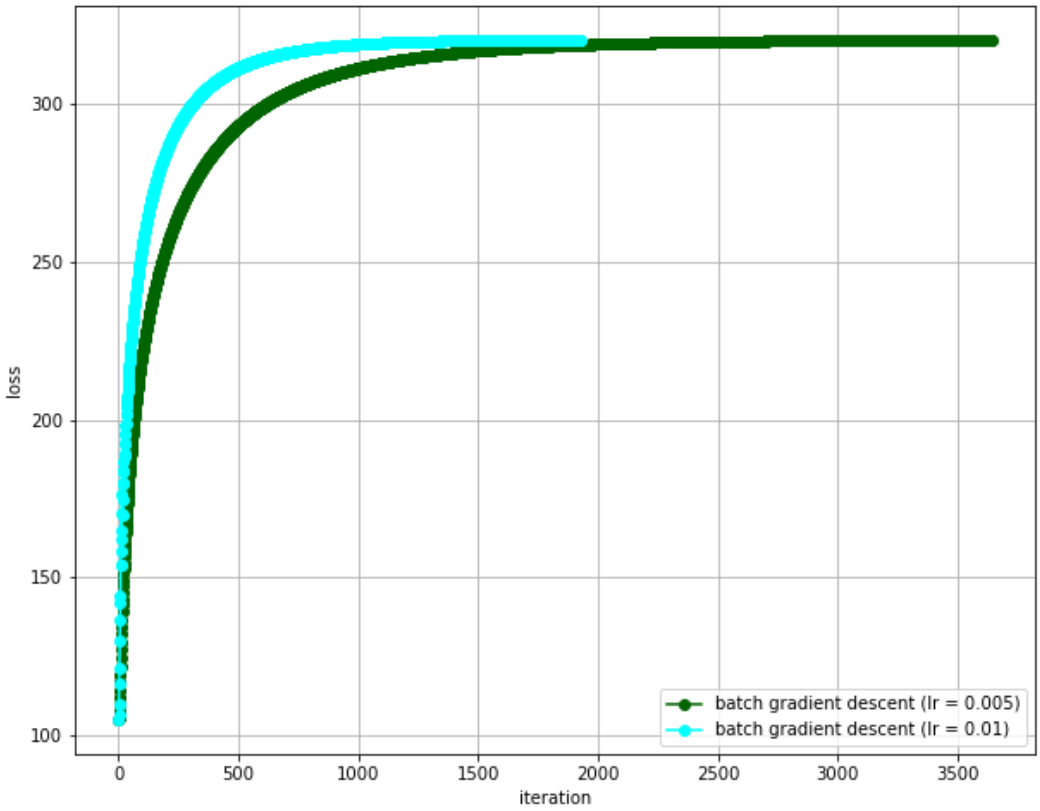
$$W \leftarrow W + \alpha\frac{\partial F(W)}{\partial W}$$

where α is our learning rate. With respect to the LMS update rule, this is nearly identical to the LMS update rule expect for two terms: (1) our $\frac{1}{n}$ and (2) our ηW term.

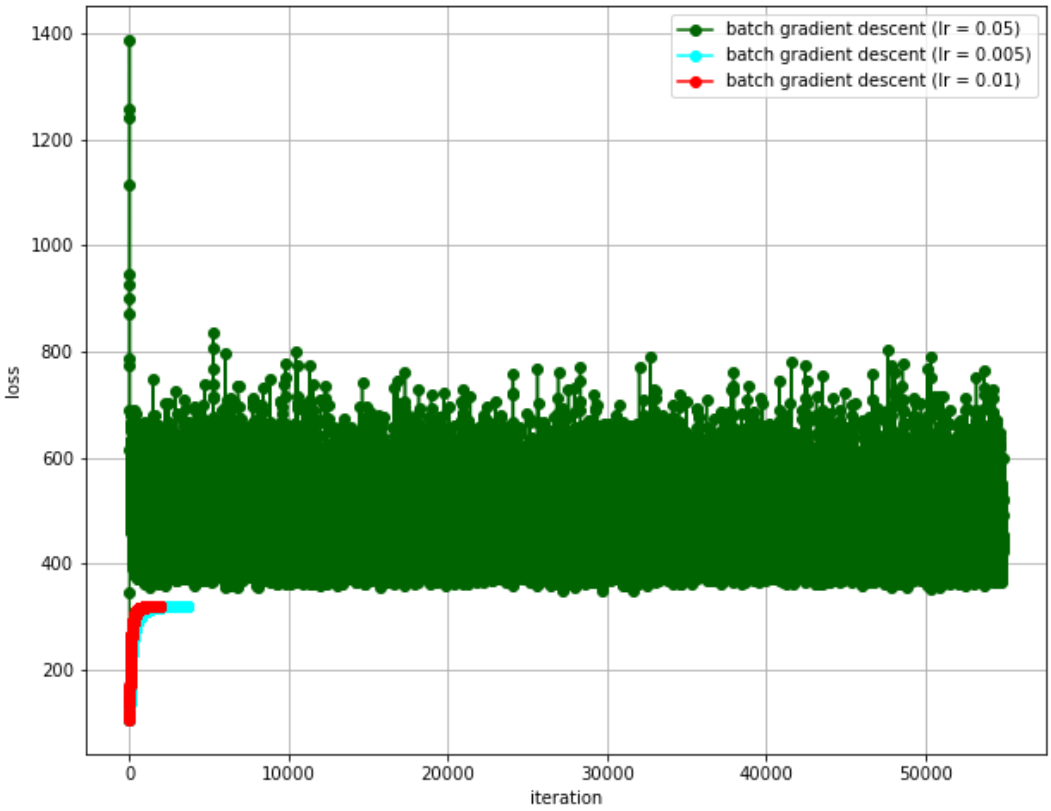
(b)

I was able to generate a plot for learning rates of $5.0e-3$ and $1.0e-2$ that seemed reasonable, but when attempting to plot the learning rate of $5.0e-2$ yielded seemingly garbage results.

The plot immediately below include only the two seemingly 'valid' learning rates:



The plot immediately below includes all 3 learning rates to demonstrate the seemingly invalid results with $5.0e-2$:



For the two results that seem valid, below is a table of their accuracy when evaluated against the training and test data.

Learning Rate	Training Precision	Test Precision
0.005	0.4254	0.449
0.01	0.4254	0.449

(c)

Given the plots above, and ignoring the garbage results for $lr = 0.05$, it seems that having a smaller learning rate results in a longer computation (more iterations) to reach a steady-state solution.

(d)

Note: Did not attempt