# EECS 545 Homework 1

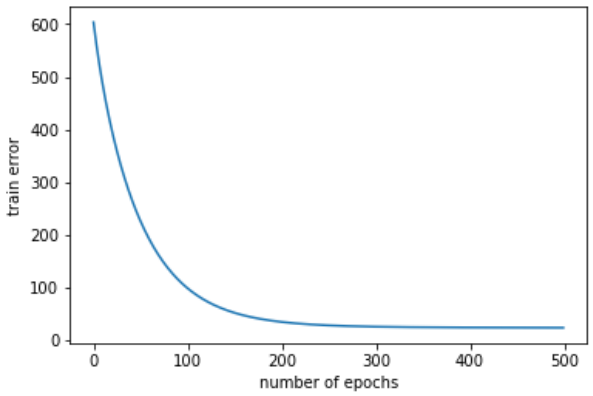
Hui Cai

1.

(b)

First, I want to mention that if the loss function is MSE, then in the formula of calculating gradient, there will be an additional term of $\frac{2}{N}$ , which causing the gradient to be smaller. So, I have to pick a learning rate to be larger than 5e-4 to ensure convergence.

For SGD, we choose learning rate: 5e-3, number of epochs: 500, that's enough to ensure convergence.




Learned weight vector: [-0.67530658, 0.70382235, -0.37334718, 0.79480359, -1.06177625, 3.25893624, 0.02689369, -2.31172154, 0.85456308, -0.62708085, -1.75778374, 0.92143721, -3.66165276] (without bias term)

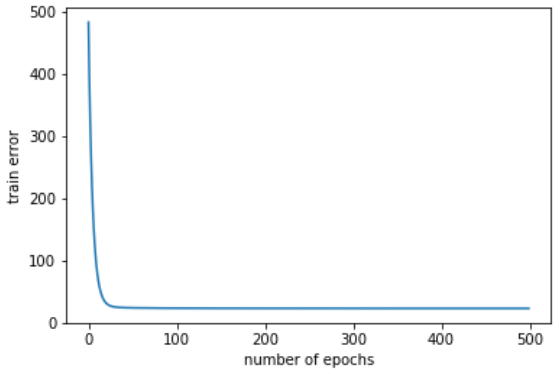
The bias term: 22.7863437344

Train error: 24.1243560454

Test error: 9.44147412663

(c)

For BGD, we choose learning rate: 5e-2, number of epochs: 500, that's enough to ensure convergence. The learning rate for BGD could be larger than SGD is that it use the whole dataset at a time and so, it would be more precise.



Learned weight vector: [-0.92826143, 1.17704793, 0.18132017, 0.67650394, -2.10542302, 2.76149471, 0.29854946, -3.13368946, 2.8202938, -2.30406969, -2.00068877, 0.90568455, -4.04887366] (without bias term)
The bias term: 22.9410087719
Train error: 23.1941398613
Test error: 10.7483532289

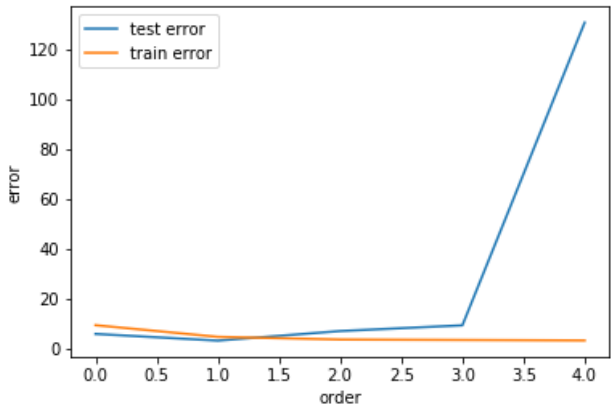
d)
Learned weight vector: [-0.93652728, 1.18983479, 0.2180906, 0.66954197, -2.10545149, 2.75102471, 0.30777503, -3.12356704, 2.96148512, -2.45469868, -2.00737039, 0.90552685, -4.05749492] (without bias term)
The bias term: 22.9410087719
Train error: 23.1915564692
Test error: 10.9665431668

We can see that the weight vector derived from the closed form is almost the same with the parameters we get before. Also, train error and test error are similar.

e)
Mean training error: 21.7053296964
Mean test error: 24.7917599242

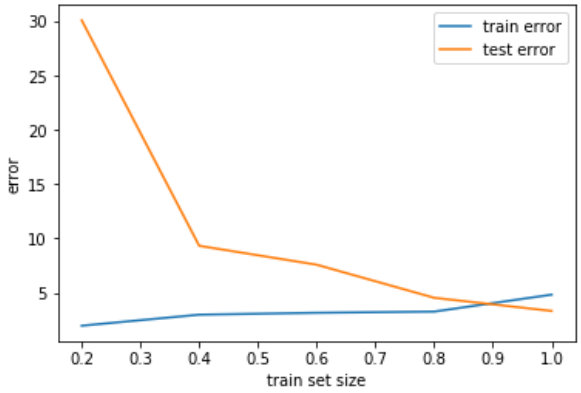
By constructing 100 random train/test splits, it's true that on average, training error is slightly smaller than test error.

2.
(a)




We can see that as the polynomial orders increase, the train error becomes smaller. That's because as the features increase, we can better fit the line. However, we can also see that, the test error is smallest at order 1. That's because at order 0, there's underfitting, as order increase, there exists problem of overfitting.

(b)




As the train set size become larger, the train error increase. That's because with more data add in, it becomes harder to fit it well with the same number of features. However, we can also see the decrease of the test error with the increase of train set size. That's because with more data to train, we actually get closer to the real underlying function (or relationship) that generates label. So, when it comes to predict, we can get better results.

3.

(a)

$$\mathrm{E}(\omega) = \frac{1}{2N}\sum_{i=1}^{N}(\omega^T\phi(x_n) - t_n)^2 + \frac{\lambda}{2}\|\omega\|^2$$

$$= \frac{1}{2N}\omega^T\Phi^T\Phi\omega - \frac{1}{N}t^T\Phi\omega + \frac{1}{2N}t^Tt + \frac{\lambda}{2}\omega^T\omega$$

where

$$\Phi = \begin{bmatrix}\phi(x_1)^T \\ \phi(x_2)^T \\ \vdots \\ \phi(x_n)^T\end{bmatrix}$$

In order to minimize this loss function, we compute the derivative,

$$\nabla_\omega E(\omega) = \frac{1}{N}\Phi^T\Phi\omega - \frac{1}{N}\Phi^Tt + \lambda\omega$$

$$= \frac{1}{N}(\Phi^T\Phi + \mathrm{N}\lambda\mathrm{I})\omega - \frac{1}{N}\Phi^Tt$$

We set $\nabla_\omega E(\omega)$ to be zero, to get $\hat{\omega}$,

$$\hat{\omega} = (\Phi^T\Phi + N\lambda I)^{-1}\Phi^Tt$$

This is the closed form solution for the weight vector to this regularized least squares objective function.

(b)

Best lamda is: 0.3

The validation error for this lamda is: 4.48444830133
The test error for this lamda is: 5.11613124943

So, the lowest RMSE identified on the validation set is 4.48 and the corresponding value for $\lambda$ is 0.3. The test error for this $\lambda$ is 5.12. By the way, when $\lambda$ is 0.2, the validation error is quite close to that of 0.3.

4. Weighted Linear Regression
We want to minimize

$$\mathrm{E}(\omega) = \frac{1}{2}\sum_{n=1}^{N} r_n(\omega^T x_n - t_n)^2$$

(a) The function above can be written as

$$\mathrm{E}(\omega) = (X\omega - t)^T R(X\omega - t)$$

where

$$\mathrm{X} = \begin{pmatrix} {x_1}^T \\ {x_2}^T \\ \vdots \\ {x_N}^T \end{pmatrix}_{N\times p}$$

$\mathrm{x}_i$ is the vector of the $\mathrm{i}_{\mathrm{th}}$ feature,

$$\mathrm{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}_{N\times 1}$$

t is the label (or response) vector derived from $t_i$,

$$\mathrm{R} = \begin{pmatrix} r_1/2 & & \\ & \ddots & \\ & & r_N/2 \end{pmatrix}_{N\times N}$$

R is a diagonal matrix derived from $r_i/2$.
We can check it equals to the loss function at the beginning just by the multiplication of matrix.

$$\mathrm{E}(\omega) = (X\omega - t)^T R(X\omega - t)$$

$$= \left(\begin{bmatrix} {x_1}^T \\ {x_2}^T \\ \vdots \\ {x_N}^T \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}\right)^T \frac{1}{2}\begin{pmatrix} r_1 & & \\ & \ddots & \\ & & r_N \end{pmatrix}\left(\begin{bmatrix} {x_1}^T \\ {x_2}^T \\ \vdots \\ {x_N}^T \end{bmatrix}\begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}\right)$$

$$= \left(\begin{pmatrix} \sum x_{1i}\omega_i \\ \sum x_{2i}\omega_i \\ \vdots \\ \sum x_{Ni}\omega_i \end{pmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}\right)^T \frac{1}{2}\begin{pmatrix} r_1 & & \\ & \ddots & \\ & & r_N \end{pmatrix}\left(\begin{pmatrix} \sum x_{1i}\omega_i \\ \sum x_{2i}\omega_i \\ \vdots \\ \sum x_{Ni}\omega_i \end{pmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}\right)$$

$$= \frac{1}{2}\sum_{n=1}^{N} r_n(\omega^T x_n - t_n)^2$$

(b)

$$\begin{aligned}\mathrm{E}(\omega) &= (X\omega - t)^T R(X\omega - t) \\ &= \omega^T(\sqrt{R}\mathrm{X})^T(\sqrt{R}\mathrm{X})\omega - t^T\mathrm{RX}\omega + t^T Rt\end{aligned}$$

In order to minimize this loss function, we compute the derivative,

$$\nabla_\omega E(\omega) = (\sqrt{R}\mathrm{X})^T(\sqrt{R}\mathrm{X})\omega - (\sqrt{R}\mathrm{X})^T\sqrt{R}t$$

We set it to be zero, then we get that,

$$\begin{aligned}\omega^* &= \left(\left(\sqrt{R}\mathrm{X}\right)^T\left(\sqrt{R}\mathrm{X}\right)\right)^{-1}\left(\sqrt{R}\mathrm{X}\right)^T\sqrt{R}\mathrm{t} \\ &= (X^T RX)^{-1}X^T Rt\end{aligned}$$

(c)

$$\mathrm{p}(t_i|\mathrm{x}_i;\omega) = \frac{1}{\sqrt{2\pi}\sigma_i}\exp(-\frac{(t_i - \omega^T x_i)^2}{2{\sigma_i}^2})$$

We compute the log likelihood function,

$$\log(\mathrm{L}) = \log\prod_{i=1}^{N} p_i = \sum_{i=1}^{N}\log p_i$$

$$= -\sum\log\sqrt{2\pi} - \sum\log\sigma_i - \sum\frac{(t_i - \omega^T x_i)^2}{2{\sigma_i}^2}$$

The first two terms in the formula is constant and not related to ω,

$$\max_\omega \log L \Leftrightarrow \min_\omega -\log L \Leftrightarrow \min_\omega \frac{1}{2}\sum_{i=1}^{N}\frac{1}{{\sigma_i}^2}(\omega^T x_i - t_i)^2$$

If we set $\mathrm{r}_i = \frac{1}{{\sigma_i}^2}$ , then this objective function is the same as the loss function stated at the beginning.