COMP 540 Statistical Machine Learning

Xiang Zhou (xz58) - Guangyuan Yu ()

2017.01.17

1 Locally weighted linear regression

- Part 1

$$J(\theta) = \sum_{i=1}^{m} w^{(i)}(\theta^T x^{(i)} - y^{(i)})^2 = (X\theta - y)^T W (X\theta - y)$$

$$A = X\theta - y = \begin{pmatrix} x_1^{(1)} & \cdots & x_D^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(N)} & \cdots & x_D^{(N)} \end{pmatrix}$$

$$J(\theta) = A^T W A = ({x^{(1)}}^T - y^{(1)} \cdots {x^{(N)}}^T\theta - y^{(N)}) \begin{pmatrix} w^{(1)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w^{(n)} \end{pmatrix} \begin{pmatrix} {x^{(1)}}^T\theta - y^{(1)} \\ \vdots \\ {x^{(N)}}^T\theta - y^{(N)} \end{pmatrix}$$

$$= \sum_{i=1}^{m} w^{(i)}({x^{(i)}}^T\theta - y^{(i)})^2 = w^{(i)}(\theta^T x^{(i)} - y^{(i)})^2$$

- Part 2

$$\begin{aligned} J(\theta) &= (X\theta - y)^T W (X\theta - y) \\ &= (\theta^T X^T - yT) W (X\theta - y) \\ &= \theta^T X^T W X\theta - \theta^T X^T W X y - y^T W X\theta + y^T W y \end{aligned}$$

Because $(\theta^T X^T W)$ and y are $1\times N$ and $N\times 1$ respectively, $\theta^T X^T W y = y^T W X\theta$.

$$\frac{dJ(\theta)}{d\theta} = 0 = \frac{d}{d\theta}[\theta^T X^T W X\theta] - 2\frac{d}{d\theta}[\theta^T X^T W y] + \frac{d}{d\theta}[y^T W y]$$

By taking the matrix derivative, we get:

$$0 = 2X^T W X\theta - 2X^T W y$$

$$\hat{\theta} = (X^T W X)^{-1} X^T W y$$

- Part 3

Algorithm 1 Calculating θ by Batch Gradient Descent

Input: Data Matrix $X \in m \times d+1$, vector $y \in m \times 1$, learning rate $\alpha \in \mathbb{R}$, input vector $x \in \mathbb{R}^{d+1}$, bandwidth of sphere of influence around x τ
Output: Vector $\theta \in \mathbb{R}^{d+1}$ that minimizes weighted LSE

$w \leftarrow m \times n$ zeros matrix
$\theta \leftarrow d \times 1$ zeros matrix
$grad \leftarrow d \times 1$ zeros matrix
for $j = 0$ to m **do**
    $w_j^{(j)} \leftarrow \frac{(x - X^{(j)})^T (x - X^{(j)})}{2\tau^2}$
end for
for $j = 0$ to 5000 **do** ▷ arbitrary number of iterations
    $grad \leftarrow \frac{X^T w (X\theta - y)}{m}$
    $\theta \leftarrow \theta - \alpha * grad$
end for
return θ

2 Properties of the linear regression estimator

- Under the assumption that $E[\epsilon] = 0$,

$$\begin{aligned} \theta &= (X^T X)^{-1} X^T y \\ E[\theta] &= E[(X^T X)^{-1} X^T y] \\ &= (X^T X)^{-1} X^T E[X\theta^* + \epsilon] \\ &= (X^T X)^{-1} (X^T X)\theta^* + (X^T X)^{-1} X^T E[\epsilon] \\ &= \theta^* + 0 = \theta^* \end{aligned}$$

- Under the assumption that $Var(\epsilon) = \sigma^2$,

$$\begin{aligned} Var(\theta) &= Var((X^T X)^{-1} X^T y) \\ &= Var((X^T X)^{-1} X^T (X\theta^* + \epsilon)) \\ &= Var(\theta^*) + Var((X^T X)^{-1} \epsilon) \\ &= 0 + (X^T X)^{-1} Var(\epsilon) \\ &= (X^T X)^{-1} \sigma^2 \end{aligned}$$

3 Implementing linear regression and regularized linear regression

3.1 Implementing linear regression

3.1.1 Linear regression with one variable

Prediction of 5% LSTAT: 298224.50151
Prediction of 50% LSTAT: -5787.31930418

3.1.2 Linear regression with multiple variables

- Prediction of 'average tract': 225328.063241
- The predictions using gradient descent and the normal equation match up. The LSE we're minimizing has a unique minimum that both methods find. The θs provided for the two solutions were different, and this is due to the fact that we normalized the feature data for the gradient descent.
- After running the four different learning rates provided (.01, .03, .1, and .3), we found that all converged, none with oscillation, on the same value. Thus, the best choice for a learning rate out of these four is 0.3 due to the fact that it converges the fastest and thus requires the least amount of iterations to finish training. We went a step further and increased the scale logarithmically again, testing out learning rates = {1,3}. Here I ran into issues where the loss climbed incredibly until it reached NAN by the 300th iteration and 200th iteration with 1 and 3 respectively. Thus, we found that .3 is the most suitable learning rate. See Figures **??**-**??**.

3.2 Implementing regularized linear regression

- As λ increases, the fit becomes resistant to changes to θ. We decrease variance and increase bias, so the shape of the fit becomes more regular and less curvy. Because of this, the training error constant increases. Validation error decreases from $\lambda = 0$, but then begins to increase as our bias becomes greater and we make more assumptions about the fit that we shouldn't be making. See Figures **??-??**

- Based on our error curve, We find a clear minimum between 0.1 and 1. Give our λ range, we conclude that 0.3 is the best choice of λ for this problem. It is also interesting to note that this gives us the highest training error. See Figure **??**

- Test error: 4.68
  Based on the error curve from the previous part, this is a good error compared to the validation and training error using our optimal λ.