HW2

Thursday, February 10, 2022 11:25 PM

1. [10 points] In Module 2, we gave the normal equation (i.e., closed-form solution) for linear regression using MSE as the cost function. **Prove that the closed-form solution for Ridge Regression** is $\boldsymbol{w} = (\lambda I + X^T \cdot X)^{-1} \cdot X^T \cdot \boldsymbol{y}$, where I is the identity matrix, $X = (x^{(1)}, x^{(2)}, \dots, x^{(m)})^T$ is the input data matrix, $x^{(i)} = (1, x_1, x_2, \dots, x_n)$ is the i-th data sample, and $\boldsymbol{y} = (y^{(1)}, y^{(2)}, \dots, y^{m})$. Assume the hypothesis function $h_w(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$, and $y^{(j)}$ is the measurement of $h_w(x)$ for the j-th training sample. The cost function of the Ridge Regression is $E(\boldsymbol{w}) = \sum_{i=1}^{m} (\boldsymbol{w}^T \cdot \boldsymbol{x}^{(i)} - y^{(i)})^2 + \lambda \sum_{i=1}^{m} w_i^2$.

Ridge Regression Closed Form: $w = (\lambda I + X^T \cdot X)^{-1} \cdot X^T \cdot y$

Ridge Regression Cost Function: $E(w) = \sum_{i=1}^{m} (w^T \cdot x^{(i)} - y^{(i)})^2 + \lambda \sum_{i=1}^{m} w_i^2$

Shape of $X \rightarrow m \times n+1$, Shape of $y \rightarrow m$ (column vector)

Shape of $w \rightarrow n+1$ (column vector), Shape of $x \rightarrow n+1$ (column vector)

$$
\begin{aligned}
E(w) &= \sum_{i=1}^{m} (w^T \cdot x^{(i)} - y^{(i)})^T \cdot (w^T \cdot x^{(i)} - y^{(i)}) + \lambda \sum_{i=1}^{m} w_i^T \cdot w_i \\
&= \sum_{i=1}^{m} (x^{(i)T} \cdot w - y^{(i)T}) \cdot (w^T \cdot x^{(i)} - y^{(i)}) + \lambda \sum_{i=1}^{m} w_i^T \cdot w_i \\
&= \sum_{i=1}^{m} x^{(i)T} \cdot w \cdot w^T \cdot x^{(i)} - x^{(i)T} \cdot w \cdot y^{(i)} - y^{(i)T} \cdot w^T \cdot x^{(i)} + y^{(i)T} \cdot y^{(i)} + \lambda \sum_{i=1}^{m} w_i^T \cdot w_i \\
&= \sum_{i=1}^{m} x^{(i)T} \cdot w \cdot w^T \cdot x^{(i)} - 2 w^T x^{(i)} \cdot y^{(i)} + y^{(i)T} \cdot y^{(i)} + \lambda \sum_{i=1}^{m} w_i^T \cdot w_i \\
&= w^T \left[\sum_{i=1}^{m} x^{(i)T} \cdot x^{(i)}\right] \cdot w - 2 w^T \sum_{i=1}^{m} x^{(i)} \cdot y^{(i)} + \sum_{i=1}^{m} y^{(i)T} \cdot y^{(i)} + \lambda \sum_{i=1}^{m} w_i^T \cdot w_i \\
&= w^T \cdot X \cdot X^T \cdot w - 2 w^T X \cdot y + y^T \cdot y + \lambda w^T \cdot w
\end{aligned}
$$

$\nabla E(w) = 2 X \cdot X^T w - 2 X \cdot y + 2 \lambda \cdot w = 0$ * set = to 0 to optimize

$\rightarrow 2 X \cdot X^T \cdot w + 2 \lambda w = 2 X^T \cdot y$

$\rightarrow X \cdot X^T \cdot w + \lambda \cdot w = X^T \cdot y$

$\rightarrow w(X \cdot X^T + \lambda I) = X^T \cdot y \Rightarrow \boxed{w = (X^T X + \lambda I)^{-1} \cdot X^T \cdot y}$

2. [10 points] Assume we have K different classes in a multi-class Softmax Regression model. The posterior probability is $\hat{p}_k = \delta(s_k(x))_k = \frac{\exp(s_k(x))}{\sum_{j=1}^{K} \exp(s_j(x))}$ for $k = 1, 2, \dots, K$, where $s_k(x) = {\theta_k}^T \cdot x$, input x is an n-dimension vector, and K the total number of classes.
   1) To learn this Softmax Regression model, how many parameters we need to estimate? What are these parameters?
   2) Consider the cross-entropy cost function $J(\Theta)$ of m training samples $\{(x_i, y_i)\}_{i=1,2,\dots,m}$ as below. Derive the gradient of $J(\Theta)$ regarding to θ_k.

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} {y_k}^{(i)} \log({\hat{p}_k}^{(i)})$$

where ${y_k}^{(i)}$ = 1 if the ith instance belongs to class k; 0 otherwise.

Posterior Probability $\hat{p}_k = \delta(s_k(x))_k = \frac{e^{s_k(x)}}{\sum_{j=1}^{K} e^{s_j(x)}}$ ← softmax function, $s_k(x) = \theta_k^T \cdot x$

Shape of $x \rightarrow n$ (column vector), K = # of classes

1) We have that $s_k(x) = \theta_k^T \cdot x$ * x is a n-dimension vector. This means that each θ_k is an n-dimension vector. Since there are K classes, we need to estimate $n \times K$ parameters. The parameters θ_k are vectors of parameters that are used to transform the input feature vectors x to $z_k(s_k)$, after which the softmax is applied to get a final prediction.

2) Cost Function: $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \cdot \log(\hat{p}_k^{(i)})$

$$\nabla J(\theta) = -\frac{1}{m}\sum_{i=1}^{m} \nabla \sum_{k=1}^{K} y_k \cdot \log(\hat{p}_k)$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\sum_{k=1}^{K} y_k \cdot \nabla \log(\hat{p}_k)\right] = -\frac{1}{m}\sum_{i=1}^{m}\left[\sum_{k=1}^{K} \frac{y_k}{\hat{p}_k} \cdot \frac{\partial \hat{p}_k}{\partial s_j}\right]$$

Need to calculate derivative of softmax: $\frac{\partial}{\partial s_j}\log(\hat{p}_k)$ ← log derivative to simplify

$\frac{\partial}{\partial s_j}\log(\hat{p}_k) = \frac{1}{\hat{p}_k} \cdot \frac{\partial \hat{p}_k}{\partial s_j}$ * $\frac{\partial}{\partial x}\log(x) = \frac{1}{x}$ * chain rule

$\frac{\partial \hat{p}_k}{\partial s_j} = \hat{p}_k \cdot \frac{\partial}{\partial s_j}\log(\hat{p}_k)$ * rearrange

$= \hat{p}_k \cdot \frac{\partial}{\partial s_j}\log\left(\frac{e^{s_k}}{\sum_{k=1}^{K} e^{s_k}}\right) = \hat{p}_k \cdot \left[\frac{\partial}{\partial s_j}\left(s_k - \log\left(\sum_{k=1}^{K} e^{s_k}\right)\right)\right]$ * substitute $\hat{p}_k$ & find log

$= \hat{p}_k \cdot \frac{\partial s_k}{\partial s_j} - \frac{\partial}{\partial s_j}\log\left(\sum_{k=1}^{K} e^{s_k}\right)$ * distribute

$= \hat{p}_k \cdot \left[\mathbb{1}\{k=j\} - \frac{\partial}{\partial s_j}\log\left(\sum_{k=1}^{K} e^{s_k}\right)\right]$ * $\frac{\partial s_k}{\partial s_j} = \begin{cases} 1, & \text{if } k=j \\ 0, & \text{otherwise} \end{cases}$

$= \hat{p}_k \cdot \left[\mathbb{1}\{k=j\} - \frac{1}{\sum_{k=1}^{K} e^{s_k}} \cdot \left(\frac{\partial}{\partial s_j}\sum_{k=1}^{K} e^{s_k}\right)\right]$ * chain rule

$= \hat{p}_k \cdot \left[\mathbb{1}\{k=j\} - \frac{e^{s_j}}{\sum_{k=1}^{K} e^{s_k}}\right]$ * partial derivative of $\sum = e^{s_j}$

$= \hat{p}_k \cdot \left[\mathbb{1}\{k=j\} - p_j\right]$ * substitute p_j for posterior

Now plug derivative of softmax back into $\nabla J(\theta)$

$$\nabla J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[\sum_{k=1}^{K} y_k \cdot (\mathbb{1}\{k=j\} - p_j)\right] \cdot x^{(i)}$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\sum_{k=1}^{K} y_k \cdot \mathbb{1}\{k=j\} - \sum_{k=1}^{K} y_k \cdot p_j\right] x^{(i)}$$ * distribute

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y_k - \sum_{k=1}^{K} y_k \cdot p_j\right] \cdot x^{(i)}$$ * $\sum \mathbb{1}\{k=j\} = 1$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y_k - p_j \sum_{k=1}^{K} y_k\right] \cdot x^{(i)}$$ * p_j comes out of $\sum$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y_k - p_j(1)\right] \cdot x^{(i)}$$ * $\sum y_k = 1$ (one-hot)

$$\Rightarrow \boxed{\frac{1}{m}\sum_{i=1}^{m}\left[\hat{p}_k^{(i)} - y_k^{(i)}\right] \cdot x^{(i)}}$$