[ML_04] Linear_Regression

Data Representation

feature vector : $\underline{x} = [x_1, x_2, \cdots x_n]^T$

Training dataset : $D = \{(\underline{x}^i, y^i)\}_{i=1}^m = \{(\underline{x}^1, y^1), \cdots, (\underline{x}^m, y^m)\}$

$$\underline{X} = \begin{bmatrix} \underline{x}^{1T} \\ \underline{x}^{2T} \\ \vdots \\ \underline{x}^{mT} \end{bmatrix} \longrightarrow \boxed{h_\theta} \longrightarrow \underline{y} = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix}$$

$$\underline{\hat{y}} = h_\theta(\underline{X}) = \begin{bmatrix} h_\theta(\underline{x}^1) \\ h_\theta(\underline{x}^2) \\ \vdots \\ h_\theta(\underline{x}^m) \end{bmatrix}$$

Linear Model Representation

$\underline{x} \triangleq [x_0\ x_1\ x_2 \cdots x_n]^T$, $\underline{\theta} = [\theta_0\ \theta_1 \cdots \theta_n]^T$

For a single training example

$$h_\theta(\underline{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$= [\theta_0\ \theta_1\ \theta_2 \cdots \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \underline{\theta}^T \underline{x}$$

For a batch of training examples

$$h_\theta(\underline{X}) = \begin{bmatrix} \underline{\theta}^T \underline{x}^1 \\ \underline{\theta}^T \underline{x}^2 \\ \vdots \\ \underline{\theta}^T \underline{x}^m \end{bmatrix} = \underline{X}\underline{\theta} \quad \rightarrow \underline{X} = \begin{bmatrix} \underline{x}^{1T} \\ \underline{x}^{2T} \\ \vdots \\ \underline{x}^{mT} \end{bmatrix} = \begin{bmatrix} x_0^1 & x_1^1 & x_2^1 & \cdots & x_n^1 \\ & & \vdots & & \\ x_0^m & x_1^m & x_2^m & \cdots & x_n^m \end{bmatrix}$$

$\underline{a}\underline{b}^T = \underline{a}^T\underline{b}$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(1)T} \\ \mathbf{x}^{(2)T} \\ \vdots \\ \mathbf{x}^{(m)T} \end{bmatrix} \rightarrow \boxed{\mathbf{h}_\theta} \rightarrow \mathbf{h}_\theta(\mathbf{X}) = \begin{bmatrix} h_\theta(\mathbf{x}^{(1)}) \\ h_\theta(\mathbf{x}^{(2)}) \\ \vdots \\ h_\theta(\mathbf{x}^{(2)}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}^T\mathbf{x}^{(1)} \\ \boldsymbol{\theta}^T\mathbf{x}^{(2)} \\ \vdots \\ \boldsymbol{\theta}^T\mathbf{x}^{(m)} \end{bmatrix} = \mathbf{X}\boldsymbol{\theta}$$

$$\mathbf{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Linear Model

MSE cost for Linear Model

classic form:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(\underline{x}^i) - y^i)^2$$

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

Vector form:

$$J(\theta) = \frac{1}{2m} \| h_\theta(\underline{X}) - \underline{y} \|_2^2$$

$$= \frac{1}{2m} \| \underline{X}\underline{\theta} - \underline{y} \|_2^2 = \frac{1}{2m} (\underline{X}\underline{\theta} - \underline{y})^T (\underline{X}\underline{\theta} - \underline{y})$$

Gradient of classic form

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (\underline{\theta}^T \underline{x}^i - y^i) x_j^i$$

Gradient of vector form

$$\nabla J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (\underline{\theta}^T \underline{x}^i - y^i) \underline{x}^i$$

$$= \frac{1}{m} \underline{X}^T (\underline{X}\underline{\theta} - \underline{y})$$

$* \ \nabla_\theta \| \underline{X}\underline{\theta} - \underline{y} \|_2^2$

$$\| \underline{X}\underline{\theta} - \underline{y} \|_2^2 = (\underline{X}\underline{\theta} - \underline{y})^T (\underline{X}\underline{\theta} - \underline{y})$$

$$= (\underline{\theta}^T \underline{X}^T - \underline{y}^T)(\underline{X}\underline{\theta} - \underline{y})$$

$$= \underline{\theta}^T \underline{X}^T \underline{X}\underline{\theta} - \underline{y}^T \underline{X}\underline{\theta} - \underline{\theta}^T \underline{X}^T \underline{y} + \underline{y}^T \underline{y}$$

$$= \underline{\theta}^T \underline{X}^T \underline{X}\underline{\theta} - 2\underline{y}^T \underline{X}\underline{\theta} + \underline{y}^T \underline{y}$$

$$\nabla_\theta \| \underline{X}\underline{\theta} - \underline{y} \|_2^2 = \nabla_\theta (\underline{\theta}^T \underline{X}^T \underline{X}\underline{\theta}) - 2\nabla_\theta (\underline{y}^T \underline{X}\underline{\theta})$$

$$= 2\underline{X}^T \underline{X}\underline{\theta} - 2\underline{X}^T \underline{y}$$

$$= 2\underline{X}^T (\underline{X}\underline{\theta} - \underline{y})$$

$\nabla_{\underline{x}} (\underline{x}^T \underline{A} \underline{x}) = 2\underline{A}^T \underline{A} \underline{x}$

$\nabla_{\underline{x}} (\underline{b}^T \underline{A} \underline{x})$

$= \nabla_{\underline{x}} \underline{C}^T \underline{x} = \underline{C}$

$= \underline{A}^T \underline{b}$

Parameter Update by GD

classic form

$$\theta_j = \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (\underline{\theta}^T \underline{x}^i - y^i) x_j^i$$

vector form

$$\underline{\theta} = \underline{\theta} - \alpha \nabla J(\underline{\theta}) = \underline{\theta} - \alpha \frac{1}{m} \underline{X}^T (\underline{X}\underline{\theta} - \underline{y})$$

Gradient Descent vs Normal Equation

$\nabla J(\underline{\theta}) = \frac{1}{m} \underline{X}^T (\underline{X}\underline{\theta} - \underline{y})$ $\quad\quad$ $\underline{\theta} = (\underline{X}^T \underline{X})^{-1} \underline{X}^T \underline{y}$

- Need to choose alpha
- Needs many iterations
- $O(kn^2)$
- works well when n is large

- No need to chose alpha
- No need to iterate
- $O(n^3)$
- slow if n is very large

Feature Normalization

Feature scaling

→ Feature 간의 size 차이가 크면 cost function이 특정 feature의 영향을 많이 받게 됨.

→ 이것은 모든 feature를 공평하게 고려하려는 의도와 어긋남

→ 그래서 feature 간의 size를 맞추기 위해 scaling 진행

→ 각 Feature의 Max 값으로 나누어줌

→ 즉, 각 feature의 Max값을 1로 통일해줄 수 있음.