

Formula

$$y = f(x) = w_0 + \sum_{j=1}^n w_j x_j = \theta^T x$$
$$x = (1, x_1, x_2, \dots, x_n)$$

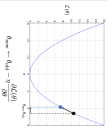
Learning Objective: minimize loss function

Take the quadratic loss as the corresponding loss

$$L(w, f(x)) = \frac{1}{2} (y - f(x))^2$$

Ways to minimize loss function

Gradient learning Methods


$$f_{k+1} = f_k - \eta \frac{\partial f(x)}{\partial w}$$

Strategies for Gradient Descent

Batch Gradient Descent

计算梯度: $\frac{\partial L(w)}{\partial w} = \frac{1}{2N} \sum_{i=1}^N 2(w - w_0)(x_i - w_0)$
 $= \frac{1}{N} \sum_{i=1}^N (w - w_0)(x_i - w_0)$
 $= \frac{1}{N} \sum_{i=1}^N (w x_i - w_0 x_i - w w_0 + w_0^2)$
 $= \frac{1}{N} \sum_{i=1}^N (w x_i - w_0 x_i - w w_0 + w_0^2)$
 $= \frac{1}{N} \sum_{i=1}^N (w x_i - w_0 x_i - w w_0 + w_0^2)$

Batch: Each type of gradient descent uses all the training examples.

Stochastic Gradient Descent 随机梯度下降

$$\frac{\partial L(w)}{\partial w} = \frac{1}{2N} \sum_{i=1}^N 2(w - w_0)(x_i - w_0)$$
$$w_{k+1} = w_k + \eta (x_k - w_k x_k)$$

Mini-Batch Gradient

Split the whole dataset into B partitions

1, 2, 3, ..., B

Randomly select b partitions for each iteration

Gradient Descent: $\frac{\partial L(w)}{\partial w} = \frac{1}{2N} \sum_{i=1}^N 2(w - w_0)(x_i - w_0)$

Batch: $L(w) = \frac{1}{2N} \sum_{i=1}^N (w - w_0)^2$ for each iteration

Stochastic GD if batch GD and stochastic GD

compare with BGD

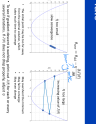
Faster learning
Uncertainty or fluctuation in learning

Basic Search Procedure

- Choose a method such as η
- Calculate iteratively with the data
- Stop the iteration if reaching a minimum

Choosing proper learning Rate

Learning rate η is a key factor in the gradient descent method. It determines the step size of the iteration. If the learning rate is too large, the iteration will diverge. If the learning rate is too small, the iteration will be slow.



1. Choose a proper learning rate η

2. Calculate iteratively with the data

3. Stop the iteration if reaching a minimum

Matrix form of linear equation

$$y = Xw$$
$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_1 & x_2 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & \dots & x_n \end{bmatrix}, w = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}$$

1. Choose a proper learning rate η

2. Calculate iteratively with the data

3. Stop the iteration if reaching a minimum

Matrix form of Gradient Descent

Objective: $\min_w \frac{1}{2} \|y - Xw\|^2 = \frac{1}{2} (y - Xw)^T (y - Xw)$

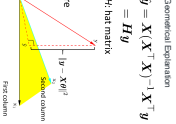
Gradient: $\frac{\partial L(w)}{\partial w} = -X^T (y - Xw)$

Station: $\frac{\partial L(w)}{\partial w} = 0 \Rightarrow -X^T (y - Xw) = 0$
 $\Rightarrow X^T X w = X^T y$

Geometrical Explanation

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

H : hat matrix



1. Choose a proper learning rate η

2. Calculate iteratively with the data

3. Stop the iteration if reaching a minimum

Probabilistic

Linear regression with Gaussian noise model

Formula

$$y = f(x) + \epsilon = w_0 + \sum_{j=1}^n w_j x_j + \epsilon$$
$$\epsilon \sim N(0, \sigma^2)$$
$$x = (1, x_1, x_2, \dots, x_n)$$

Learning objective

Data likelihood

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - f(x))^2}{2\sigma^2}}$$

Maximize the data log-likelihood

$$-\log p(y|x) = -\log \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - f(x))^2}{2\sigma^2}} \right)$$
$$= \frac{1}{2} \log(2\pi\sigma^2) + \frac{(y - f(x))^2}{2\sigma^2}$$
$$= \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - f(x_i))^2$$

1. Choose a proper learning rate η

2. Calculate iteratively with the data

3. Stop the iteration if reaching a minimum

Deterministic

Linear regression model

Algebra Perspective