19 Linear models

Note: NN and SVM are linear classifiers

Univariable LR: $W = [w_0(b), w_1]$
$X = [1, x_1]$

$$\boxed{\begin{array}{l}\text{find a line:} \\ \hat{y} = h_W(x) = W \cdot X \\ \text{fits best the data}\end{array}}$$ 原定义

$$L(y,\hat{y}) = L_1(y,\hat{y}) = |y-\hat{y}|$$
$$L_2(y,\hat{y}) = (y-\hat{y})^2$$

加上 Loss function，改写为：

$$\boxed{\begin{aligned} W^* &= \underset{W}{\operatorname{argmin}} L(h_W) \\ &= \underset{W}{\operatorname{argmin}} \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2 \end{aligned}}$$

其中，N为sample的~~数量~~

w_1 和 $w_0(b)$ 可以直接算出来，见slides

general learning: $\boxed{W^* = \underset{W}{\operatorname{argmin}} L(h_W)}$

gradient descent: for every w_i:

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} L(w)$$

update rule　　learning rate　　loss func along w_i axis

eg. $w_0 \leftarrow w_0 - \alpha \sum_j (y_j - h_W(x_j))$
$w_1 \leftarrow w_1 - \alpha \sum_j (y_j - h_W(x_j)) \times x_j$

batch gd : use all train data
promise to converge but slow

stochastic gd :
- random
- apply single point update
- decreasing learning rate (like simulate annealing)

not guarantee to converge but faster

<u>multi-variable linear regression</u> :

$$w^* = \underset{w}{\mathrm{argmin}} \sum_j L_2(y_j, w \cdot x_j)$$

<u>Classification</u> : decision boundary : path (surface) that separates two classes

linear separater ✓ ⇒ linearly saparable

$$\boxed{\begin{aligned} h_w(x) &= \mathrm{Threshold}(w \cdot x) \\ w^* &= \underset{w}{\mathrm{argmin}}\ L(h_w) \end{aligned}}$$

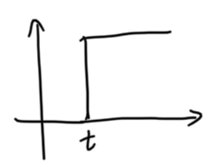
loss target

Regression : 多项式 ...
classifier : sigmoid ...

perceptron learing rule :

$$w_i \leftarrow w_i + \alpha(y - h_w(x)) \times x_i$$

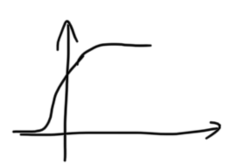
hard threshold :



un-predictable
not robust to noise

Logistic regression :

Soft threshold : (sigmoid)



more predictable
robust to noise