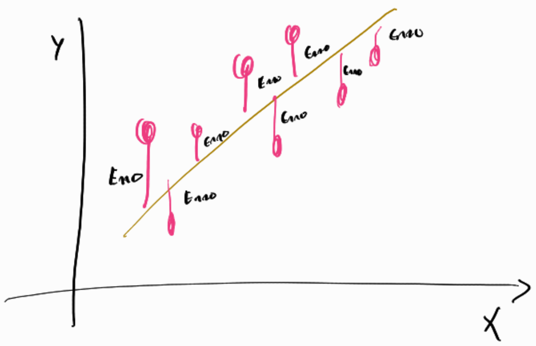
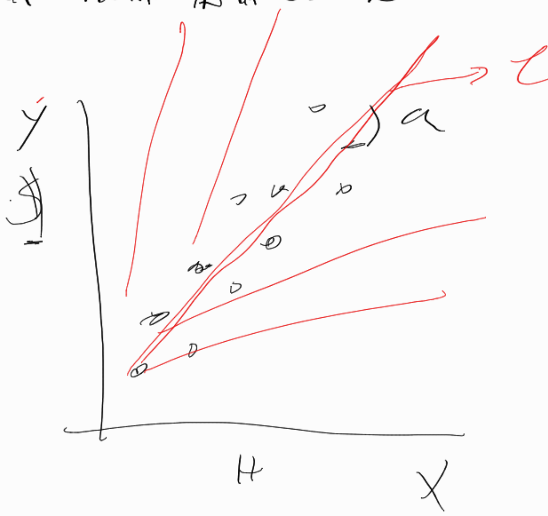
A ideia é calcular a qualidade de uma reta através do erro quadrático





Podemos descrever uma reta sabendo apenas dois parâmetros a e b, sendo respectivamente o coeficiente angular e linear.

Matematicamente construindo uma reta como:

$$\boxed{y = ax + b.} \longrightarrow \$ = aH + b$$

Sabendo o erro, podemos então atualizar os valores de "a" e "b" utilizando a taxa de variação do erro, também chamada de função custo. Podemos escrever a mesma como

1. $$J(\theta) = \frac{1}{2m}\sum (\underbrace{h_\theta(x^i)}_{\theta_0 + \theta_1 x} - \underbrace{y^i}_{\text{valor real}})^2 , \text{ onde } \theta_0 = b \text{ e } \theta_1 = a$$

b a

A fç "custo" determina, de certa forma, o quão longe estamos do valor real, dado um valor qualquer de θ. Utilizando a ideia de cálculo variacional, podemos então atualizar os parâmetros θ através da $\frac{d}{d\theta}J(\theta)$.

$$\frac{\partial h_\theta}{\partial \theta_1} = \frac{\partial(\theta_0 + \theta_1 x)}{\partial \theta_1} = x$$

$$\frac{d}{d\theta} J(\theta) = \frac{1}{2m} \sum 2(h(\theta) - y) \cdot x = \boxed{\sum (h(\theta) - y)\, x}$$

Logo

$$\theta = \theta - \alpha \frac{dJ}{d\theta}$$

↳ Taxa de aprendizado.

o OBS: Para implementar de forma eficiente é necessário vetorizar, trabalhando na forma matricial, podemos aproveitar o produto de matrizes e evitar loops que aumentam o custo computacional;

Fazendo $x^i = [x_1 \ x_2 \ x_3 \ \ldots] = x^i = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \end{bmatrix}$

e Theta $= [\theta_0 \ \ \theta_1]$

$$X \cdot \theta^T = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ \vdots & \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \theta_0 + \theta_1 x_1 \\ \theta_0 + \theta_2 x_2 \\ \vdots \end{bmatrix}$$

Para uma regressão em várias variáveis é preciso apenas acrescentar mais coeficientes na eq da reta

$$\boxed{h(\theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \ldots}$$