Regressão Linear

Dados os pontos

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad [y] = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Obter a e b que melhor aproximam $\bar{y}(x) = ax + b$

Função custo: mínimos quadrados $\rightarrow J(a,b) = \frac{1}{2n} \sum_{i=1}^{n} (\bar{y}(x_i) - y_i)^2 = \frac{1}{2n} \sum_{i=1}^{n} (ax_i + b - y_i)^2$

O gradiente de uma função indica o sentido de máximo incremento em seu valor

$$\nabla J(a,b) = \left[\frac{\partial J(a,b)}{\partial a}, \frac{\partial}{\partial b} J(a,b) \right]^t$$

Solução de MAT-27:

$$\underbrace{[A]}_{\begin{bmatrix} a \\ b \end{bmatrix}} = (\underbrace{X}_{\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}} X^t)^{-1} X^t \underbrace{Y}_{\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}}$$

Gradient descent

for $i = 1:MAX$

$$(a_{i+1}, b_{i+1}) = (a_i, b_i) - \alpha . \nabla J(a_i, b_i)$$

Com múltiplas variáveis

$\vec{x} = [x_1 \ \ x_2 \ \ x_3 \dots \ x_n]^t$

y é escalar

$A = [a_0 \ \ a_1 \ \dots \ a_n]$

$$\bar{y}(\vec{x}) = A . \begin{bmatrix} 1 \\ \vec{x} \end{bmatrix} = a_0 + a_1 x_1 + \dots + a_n x_n$$

$$J(A) = \frac{1}{2n} \sum_{i=1}^{n} (\bar{y}(\vec{x}_i) - y_i)^2$$

$$\frac{\partial J(A)}{\partial a_k} = \frac{1}{n} \sum_{i=1}^{n} (A . \vec{x}_i - y_i) . x_k^{(i)}$$

Regressão Polinomial

- Adaptação da regressão linear
- Em vez de aproximar por retas, aproximamos por polinômios de qualquer grau
- Cada potência é uma nova variável

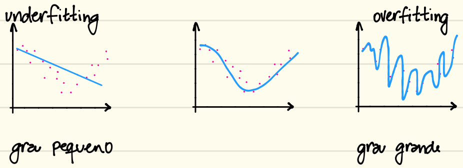
$$\bar{y}(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

$$J(A) = \frac{1}{2m} \sum_{i=1}^{m} (\bar{y}(x_i) - y_i)^2$$

$$\frac{\partial J(A)}{\partial a_k} = \frac{1}{m} \sum_{i=1}^{m} (a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_n x_i^n - y_i) x_i^k$$

```
for i=1:MAX
    A_{i+1} = A_i - α . ∇J(A_i)
end
```

O grau escolhido do polinômio pode não ser adequado



Regressão Logística

É como a regressão linear, mas o resultado da previsão é binério ($\emptyset$ ou 1) e não real.

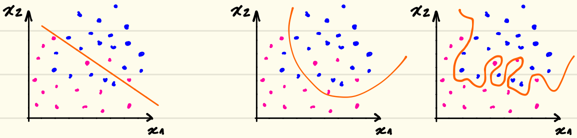
2 passos:

1) Dividir o espaço em regiões de decisão
2) Associar 0 ou 1 para cada região de decisão

Passo 1

Dado o ponto $\vec{x}$, fazemos $\theta(\vec{x})$, em que θ é uma função $\theta: \mathbb{R}^n \to \mathbb{R}$ e pode ser tanto uma regressão linear como polinomial

Os pontos em que $\theta(\vec{x}) = 0$ são os decision boundary



$\theta_0 + \theta_1 x_1 + \theta_2 x_2 = 0$

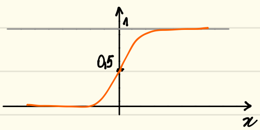
$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 = 0$

$\theta_0 + \ldots + \theta_6 x_1^3 x_2 + \ldots = 0$

Passo 2

Separa em $\emptyset$s e 1s

Função sigmoid: $g(x) = \dfrac{1}{1 + e^{-x}}$



$x < 0$: converge rápido para $\emptyset$

$x > 0$: converge rápido para 1

é contínua e suave

Passo 1 + Passo 2: $h_\theta(x) = g(\theta(\vec{x}))$

Temos um modelo que varia os parâmetros em θ. Definimos θ minimizando a função custo com os pontos conhecidos

Qual a função custo?

Acertou $\emptyset$ ou 1: custo zero $\rightarrow -\ln(1-h_\theta(x^i))$ se $y^i=0$

Errou: custo $\rightarrow \infty$ $\rightarrow -\ln(h_\theta(x^i))$ se $y^i=1$

$\rightarrow cost_\theta(x^i) = -y^i \ln(h_\theta(x^i)) - (1-y^i)\ln(1-h_\theta(x^i))$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^i \ln(h_\theta(x^i)) - (1-y^i)\ln(1-h_\theta(x^i))\right]$$

Também vamos utilizar gradient descent:

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

Regularização

Evita overfitting $\rightarrow$ pode ser utilizado em qualquer regressão estudada aqui

Overfitting é causado por parâmetros θ_i muito grandes, especialmente se i representar o coeficiente de um grau grande do polinômio

Penalizamos θ_i's grandes na função de custo

$$J_{reg}(\theta) = J(\theta) + \underbrace{\frac{\lambda}{2m}}_{\text{peso da regularização}}\sum_{i=1}^{m}\theta_i^2$$

$$\frac{\partial J_{reg}}{\partial \theta_j}(\theta) = \frac{\partial J(\theta)}{\partial \theta_j} + \frac{\lambda}{m}\theta_j$$

Encontrar o λ ideal pode evitar também underfitting