REGRESSÃO LINEAR
COM MÚLTIPLAS VARIÁVEIS

DECOM

Rogério Gomes e Paulo Almeida

Múltiplas características (variáveis).

Até agora vimos:

Size (feet^2)	Price ($1000)
x	y
2104	460
1416	232
1534	315
852	178
…	…

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Múltiplas características (variáveis).

Size (feet^2)	Number of bedrooms	Number of floors	Age of home (years)	Price ($1000)
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315
852	2	1	36	178
…	…	…	…	…

Notação:

n = número de características (features)

$x^{(i)}$ = Entrada (features) do i^{th} exemplo de treinamento

$x_j^{(i)}$ = valor da característica j do i^{th} exemplo de treinamento

Hipótese:

Anterior: $h_\theta(x) = \theta_0 + \theta_1 x$

Agora: $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$

Hipótese:

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

Por conveniência de notação, definimos $x_0 = 1$.

Assim:

Hipótese:

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \Re^{n+1} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} \in \Re^{n+1}$$

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

$$= \theta^T X$$

Gradiente descendente para múltiplas variáveis

Hipótese: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parâmetros: $\theta_0, \theta_1, \ldots, \theta_n$

Função Custo: $J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$

Gradiente descendente:

Repeat $\{$

$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$

$\}$

(Atualize simultaneamente para todo $j = 0, \ldots, n$)

Gradiente Descendente

Anteriormente (n=1):

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \underbrace{\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})}_{\frac{\partial}{\partial\theta_0}J(\theta)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

(Atualize simultaneamente θ_0, θ_1)

$\}$

Novo algoritmo $(n \geq 1)$:

Repeat $\{$

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

(Atualize simultaneamente $\theta_j, j = 0, \ldots, n$)

$\}$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

...

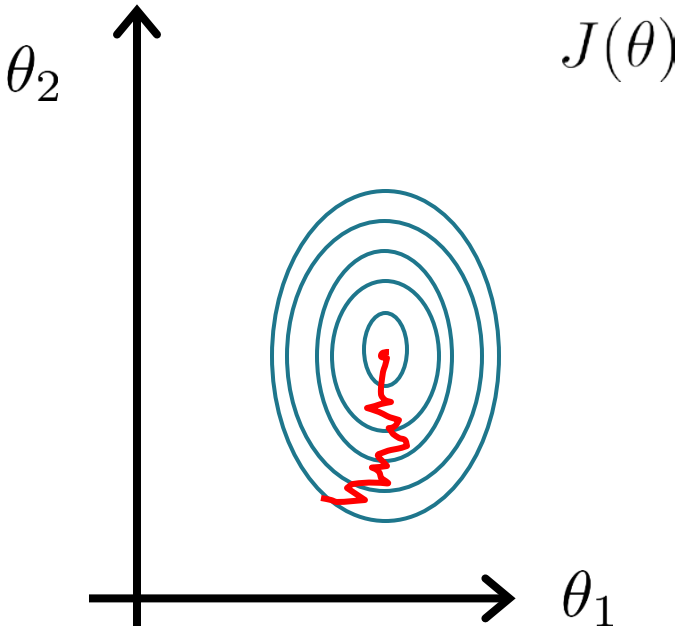
Gradiente descendente: Feature Scaling

Feature Scaling

Ideia: Colocar as características dentro da mesma escala.

E.g. x_1 = size (0-2000 feet^2)

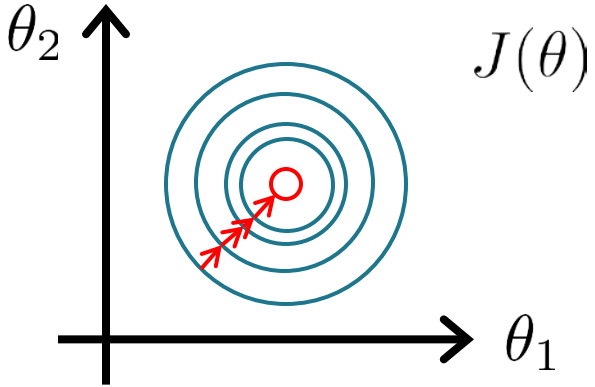
x_2 = number of bedrooms (1-5)



$$x_1 = \frac{\text{size (feet}^2\text{)}}{2000}$$

$$x_2 = \frac{\text{number of bedrooms}}{5}$$

$$0 \le x_1 \le 1 \quad 0 \le x_2 \le 1$$



Feature Scaling

Coloque as características (features) dentro do intervalo

$$-1 \leq x_i \leq 1$$

$x_0 = 1$

$0 \leq x_1 \leq 3$

$-2 \leq x_2 \leq 0.5$

$-100 \leq x_3 \leq 100$

$-0.0001 \leq x_4 \leq 0.0001$

$-3 \leq x \leq 3$

$-\frac{1}{3} \leq x \leq \frac{1}{3}$

Mean normalization

Troque x_i por $x_i - \mu_i$ - (Não faça isso para $x_0 = 1$).

E.g. $x_1 = \frac{size - 1000}{2000}$

$x_2 = \frac{\#bedrooms - 2}{5}$

$-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5$

$x_1 \leftarrow \frac{x_1 - \mu_1}{s_1}$ $x_2 \leftarrow \frac{x_2 - \mu_2}{s_2}$ S – Range (Max-Min) ou Desvio padrão.

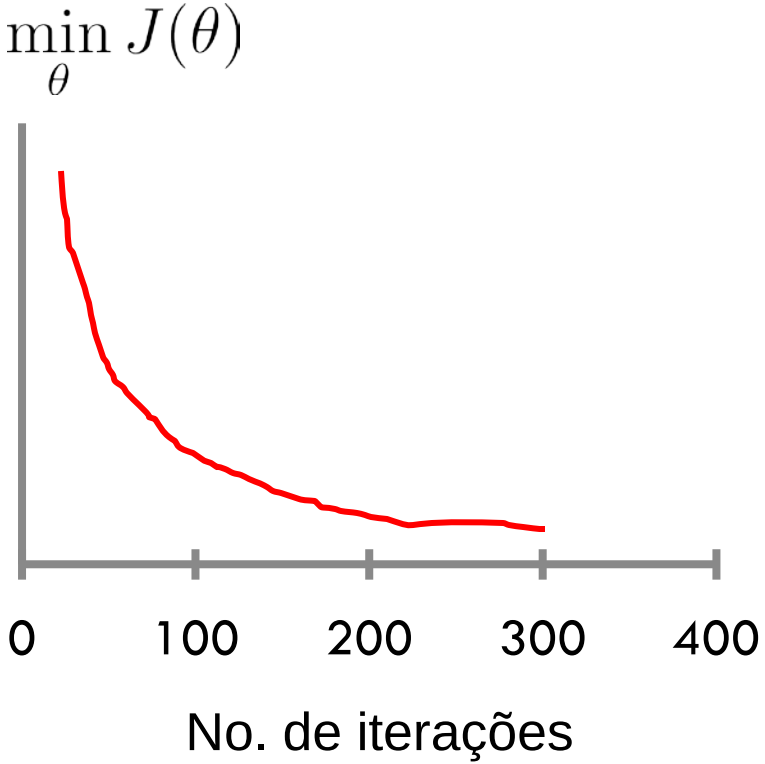
Gradiente descendente: Taxa de Aprendizado

Gradiente descendente

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

- “Debugging”: Como saber se o Gradiente descendente está funcionando corretamente.
- Como escolher a taxa de aprendizado α .

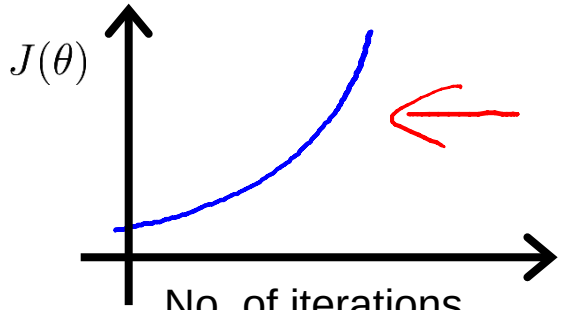
O gradiente descendente está funcionando corretamente?



Examplo de teste automático de convergência:

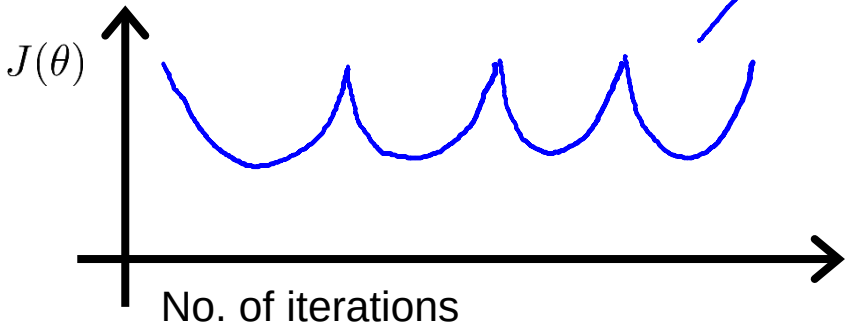
Considere convergente se $J(\theta)$ decresce pelo menos 10^{-3} em cada iteração.

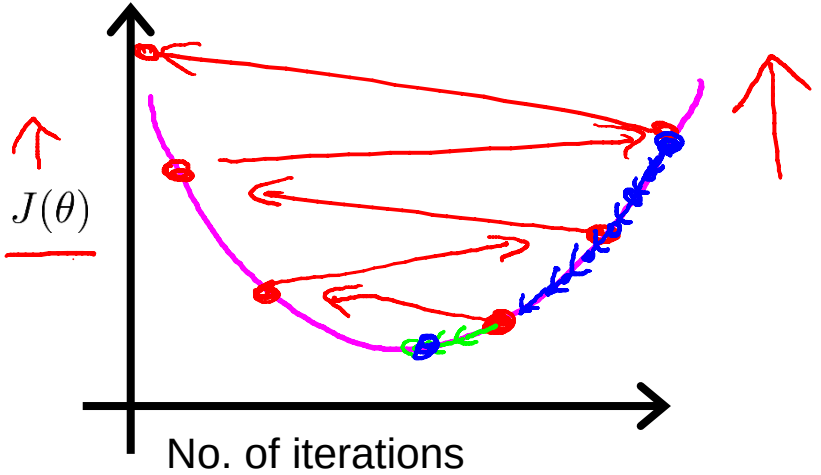
O gradiente descendente está funcionando corretamente?



Gradiente descendente não está funcionando?

use um α pequeno.





- Para um valor pequeno de α, $J(\theta)$ deve decrescer a cada iteração.
- Mas se α é muito pequeno, o gradiente descendente pode convergir muito lentamente.

Resumo:

- Se α for muito pequeno: convergência Lenta.
- Se α for muito grande: $J(\theta)$ poderia não decrescer em cada iteração, ou mesmo, não convergir.

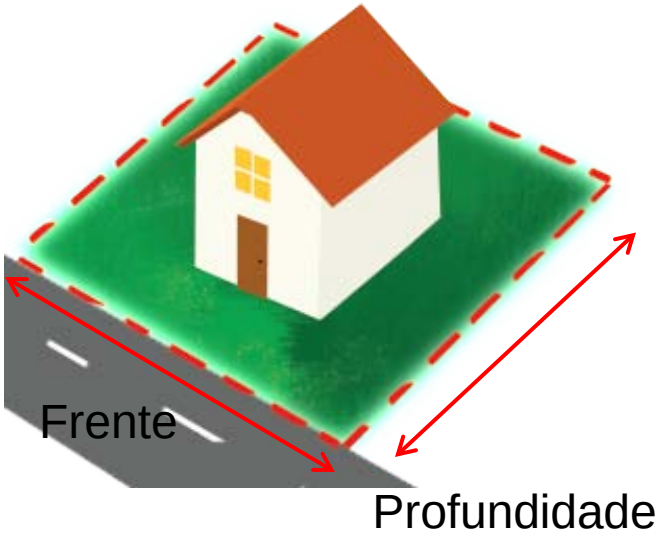
Para escolher α, tente:

…, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, …

Características (Features) e Regressão polinomial

Predição do preço de casas

$$h_\theta(x) = \theta_0 + \theta_1 \times frente + \theta_2 \times \mathrm{Pr}ofundidade$$

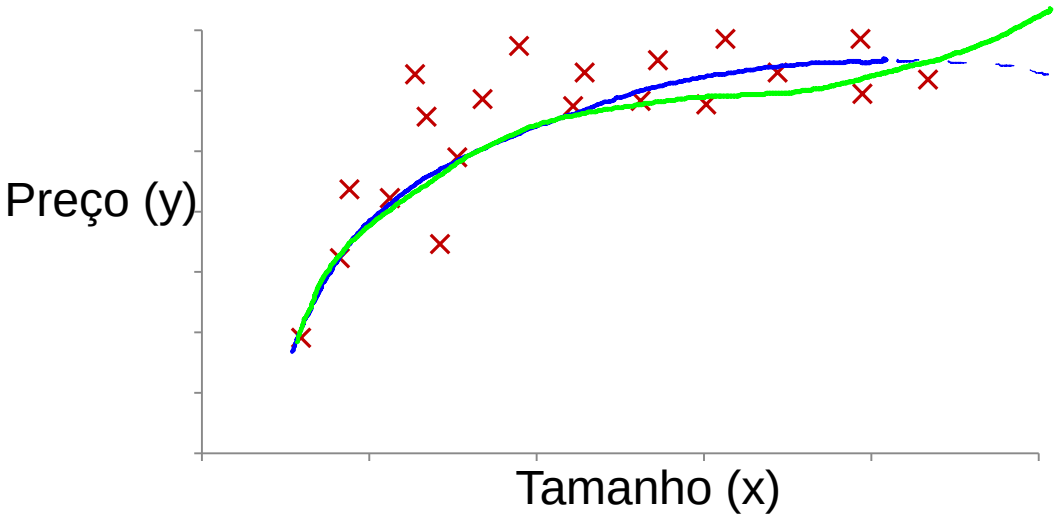


Área:

X=frente x profundidade

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Regressão Polinomial



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

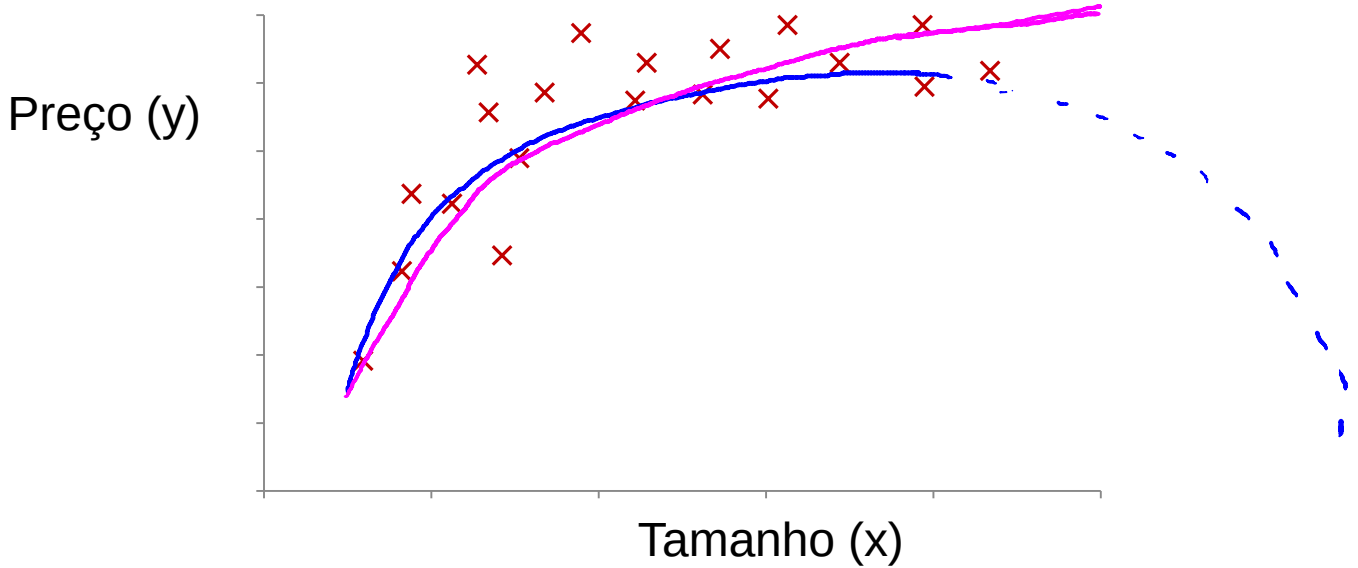
$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

$$= \theta_0 + \theta_1 (size) + \theta_2 (size)^2 + \theta_3 (size)^3$$

$$x_1 = (size)$$
$$x_2 = (size)^2$$
$$x_3 = (size)^3$$

Escolha de características (features)

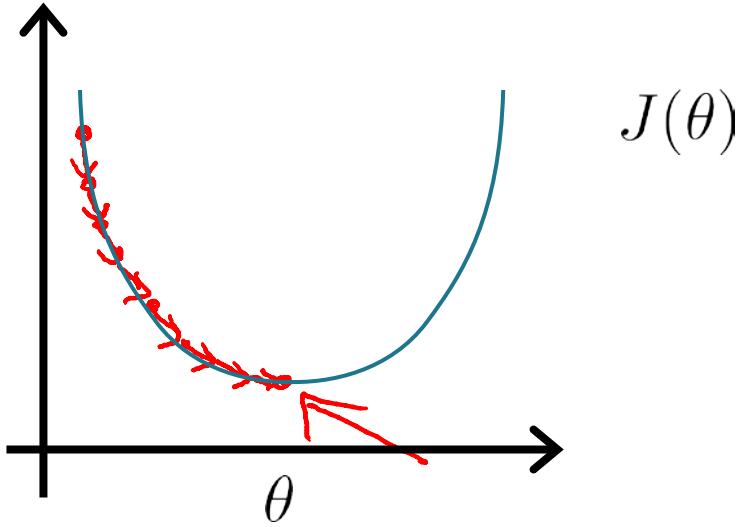


$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2(size)^2$$

$$h_\theta(x) = \theta_0 + \theta_1(size) + \theta_2\sqrt{(size)}$$

Equação Normal

Gradiente Descendente



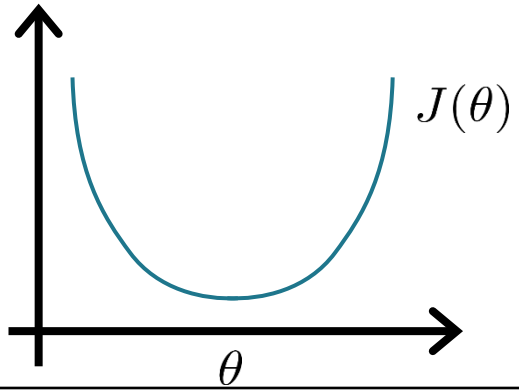
Equação Normal: Método para resolver θ analiticamente.

Se 1D $(\theta \in \mathbb{R})$

$$J(\theta) = a\theta^2 + b\theta + c$$

$$\frac{d}{d\theta} J(\theta) = 0$$

Resolva em função de θ



$\theta \in \mathbb{R}^{n+1}$

$$J(\theta_0, \theta_1, \ldots, \theta_m) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \cdots = 0 \quad \text{(para cada } j\text{)}$$

Resolva para cada $\theta_0, \theta_1, \ldots, \theta_n$

Exemplos: $m = 4$.

x_0	Size (feet^2) x_1	Number of bedrooms x_2	Number of floors x_3	Age of home (years) x_4	Price ($1000) y
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 & 45 \\ 1 & 1416 & 3 & 2 & 40 \\ 1 & 1534 & 3 & 2 & 30 \\ 1 & 852 & 2 & 1 & 36 \end{bmatrix} \qquad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

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

m **exemplos** $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ **;**

n **características.**

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \in \mathbb{R}^{n+1}$$

$$X = \begin{bmatrix} - & (X^{(1)})^T & - \\ - & (X^{(2)})^T & - \\ - & \vdots & - \\ - & (X^{(m)})^T & - \end{bmatrix} \qquad Y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^m \end{bmatrix}$$

Mx(n+1)

E.g. Se $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$

$$X = \begin{bmatrix} 1 & x^1 \\ 1 & x^2 \\ \vdots & \vdots \\ 1 & x^m \end{bmatrix}$$

Mx2

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

$(X^T X)^{-1}$ É a inversa da matrix $X^T X$.

Octave ou Matlab: `pinv(X'*X)*X'*y`

m exemplos de treinamento, n características.

Gradiente Descendente

- Precisa escolher α .
- Muitas iterações.
- Trabalha bem mesmo quando n é grande.

Equação Normal

- Não precisa escolher α .
- Não é iterativo
- Precisa calcular $(X^T X)^{-1}$
- Lento se n é muito grande

Equação Normal e não inversabilidade

Equação Normal

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

- Se $X^T X$ é não inversível (singular/degenerada)

Faça:

- Octave ou matlab: `pinv(X'*X)*X'*y`

E se $X^T X$ não for inversível?

- Redundante features (linearmente dependente).

  E.g. $x_1 =$ Tamanho em feet^2

  $x_2 =$ Tamanho em m^2

- Muitos features (e.g. $m \leq n$).

  - Exclua alguns features ou faça regularização.