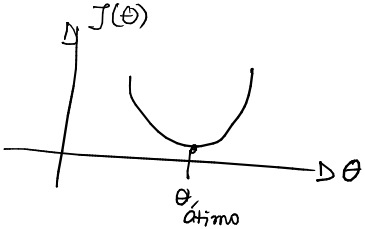
normal equation

P/ alguns problemas de regressão linear, fornece uma maneira melhor de resolver p/ o valor ótimo de θ

Em 1 step (s/ iterações) - chegamos no θ ótimo



Intuition:

If 1D ($\theta \in \mathbb{R}$) - θ é um escalar

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

$\frac{dJ(\theta)}{d\theta} = 0$ Solve for θ - se θ é real, ok

mas θ é $\mathbb{R}^{n+1}$, n+1 dimensional

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

$\frac{\partial}{\partial \theta_j} J(\theta) = \dots = 0 \quad \forall j$

Solve for $\theta_0, \theta_1, \dots, \theta_n$

n features

m - nº de training examples

x_0	x_1	x_2	x_3	x_4	y
1	-	-	-	-	a
1	-	-	-	-	b
1	-	-	-	-	c
1	-	-	-	-	d

$$X = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot \end{bmatrix}_{M \times (n+1)} \qquad Y = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}_{M \times 1}$$

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

$x^{(i)}$ → training example i

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix} \qquad X = \begin{bmatrix} -(x^{(1)})^T- \\ -(x^{(2)})^T- \\ \vdots \\ -(x^{(m)})^T- \end{bmatrix}_{m \times (n+1)}$$

Se eu só tenho 1 feature:

$$x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$$

$$X = \begin{matrix} 1 & x_1^{(1)} \\ 1 & x_1^{(2)} \end{matrix}$$

normal equation

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

feature scaling is not necessary!

↳ Só p/ gradient descent!

m training examples, n features

GD	NE
- preciso de um α.	- não preciso escolher α
- preciso de mtas iterações	- não preciso iterar
- funciona bem p/ mtas features	- p/ muitas features preciso calcular $(X^T X)^{-1}$ (custo da inversa: $O(n^3)$) nxn se tenho n features
	- slow if n is large

se n=1000 ainda OK inverter, mas acima de 10.000, problema!

inverter a million x million matrix - B.O.

normal equation é bom e rápido no caso de linear regression e com poucas features. Caso contrário, tem que ser outro método, como GD.

When n exceeds 10.000, good times to go from normal solution to iterative process.

E se X^TX for singular?

Causas:

- redundant features (L. Dependency.)

  x_1 = size in feet

  x_2 = size in m

  $x_1 = 3{,}28\, x_2$

- too many features ($m \leq n$)

  ex: $m = 10$ training ex.

  $n = 100$ features $\quad \theta \in \mathbb{R}^{101}$

  Passo deletar algumas features ou usar regularização

Vetorização

Ex: hipótese p/ regressão linear

$$h_\theta(x) = \sum_{j=0}^{n} \theta_j x_j = \underbrace{\theta^T x}_{\text{inner product}}$$

Unvectorized:

prediction = 0.0;

for j = 1:n+1,

prediction = prediction + theta(j) * x(j)

end;

Vectorized: double prediction = theta. transpose() * x;

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} \qquad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$$