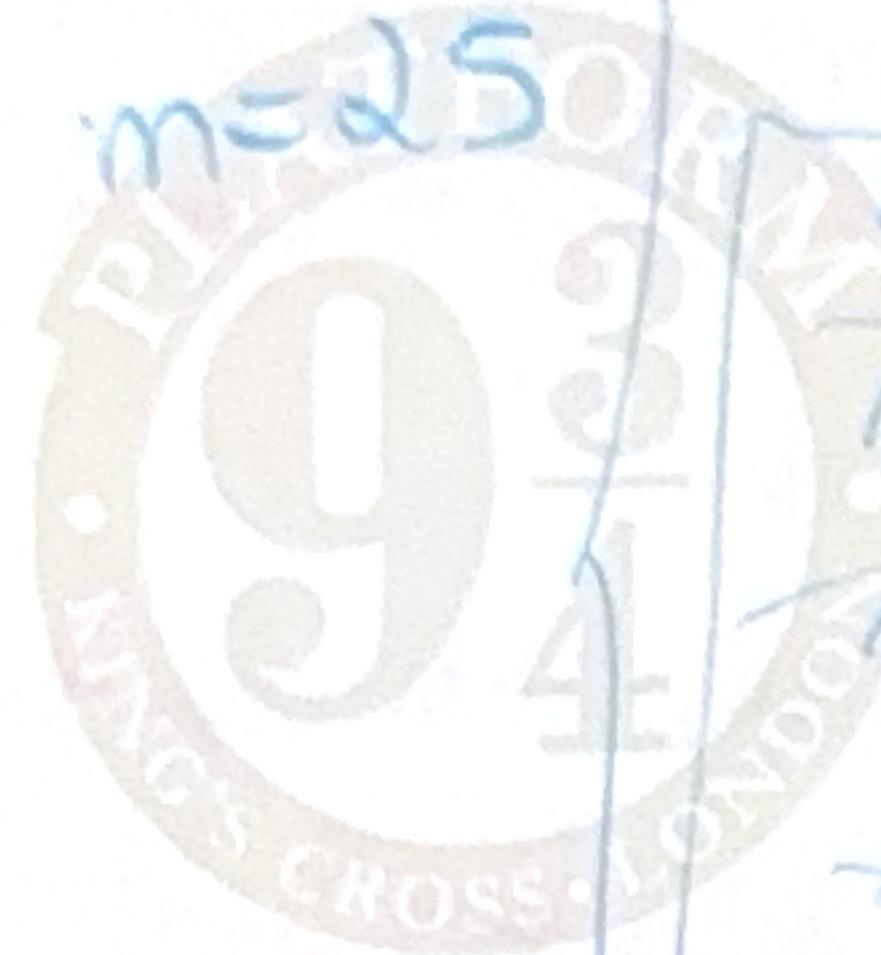


$x_0 \quad x_1 \quad x_2 \quad x_3 \quad y$



$$\begin{array}{c}
 m=25 \\
 \begin{array}{|c|c|c|c|c|} \hline & x_0 & x_1 & x_2 & x_3 & y \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n=4 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} \\
 A \quad (25 \times 4) \quad (4 \times 1) \quad (25 \times 1)
 \end{array}$$

$$\Theta = (\underbrace{A^T A}_{B})^{-1} \underbrace{A^T}_{C} \cdot Y$$

$$\Theta_0 = 0.14210$$

$$\Theta_1 = 8.64061$$

$$\Theta_2 = -2.00702$$

$$\Theta_3 = 5.00001$$

$$\Theta_0 = 100$$

$$\Theta_1 = 7$$

$$\Theta_2 = -2$$

$$\Theta_3 = 5$$

$$J = 2215.47$$

$$J = 0$$

Solução encontrada: grafos muitíssimo próximos

Gradient Descent X Normal Equation

- Needs to choose α • No need for α
 - Needs many iterations • Doesn't need to iterate
 - Works well even when n is large (even thousands of features)
 - Need to compute $(X^T X)^{-1}$ $O(n^3)$
 - Slow if n is very large
- $n = 100 \text{ OK}$
- $n = 10^4 \text{ or greater } O(K \cdot n^2)$
- $n = 1000 \text{ OK}$
- $n = 10000 \text{ hmm... maybe}$

Normal equation
usually doesn't work
with more sophisticated learning algorithms

in this case,
we apply the gradient descent

~~$X^T X$ não é invertível se...~~

1) houver colunas redundantes (proporcionais)

Eg: $x_1 = \text{size}(f\{)^2$, $x_2 = \text{size}(m^2)$

2) too many features $m \leq n$

↳ delete some features or use REGULARIZATION