Gems of TCS

Linear Regression

Sasha Golovnev

May 6, 2021

Classes of Learning Problems

- Classification

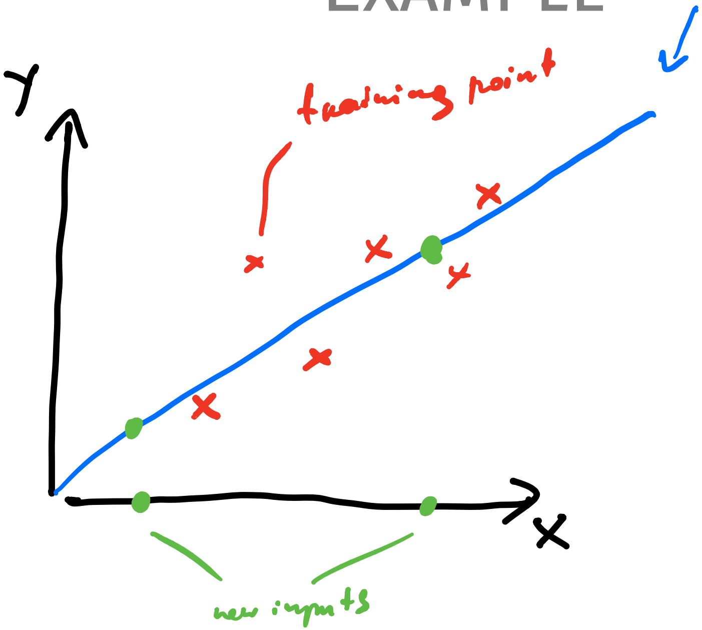
0	non-spam
1	spam
2	phishing

- Ranking
- Regression

→ real-valued output

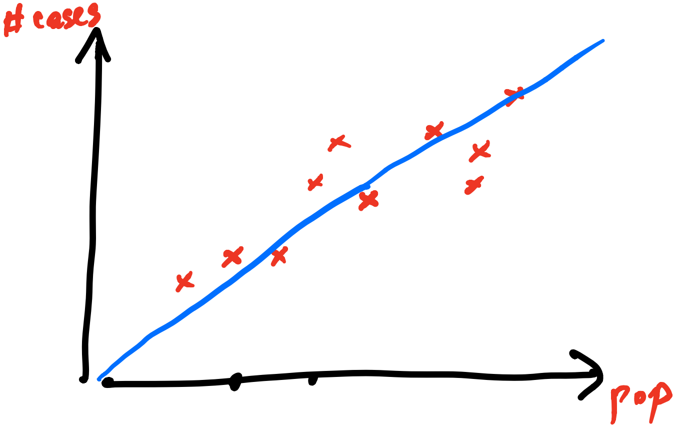
- Clustering
- . . .

Example



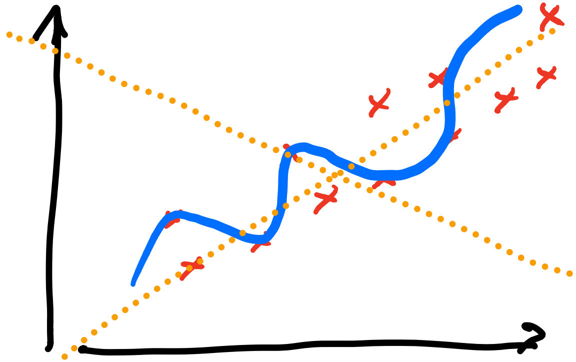
Example

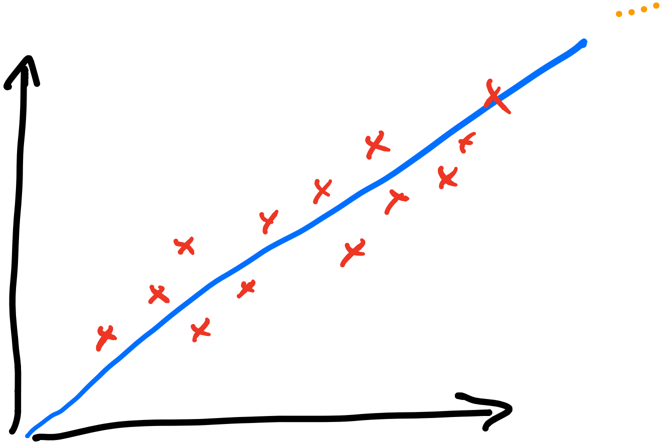
- Predict # of covid cases given city population



Linear Regression

predictor (hypothesis) is a linear function





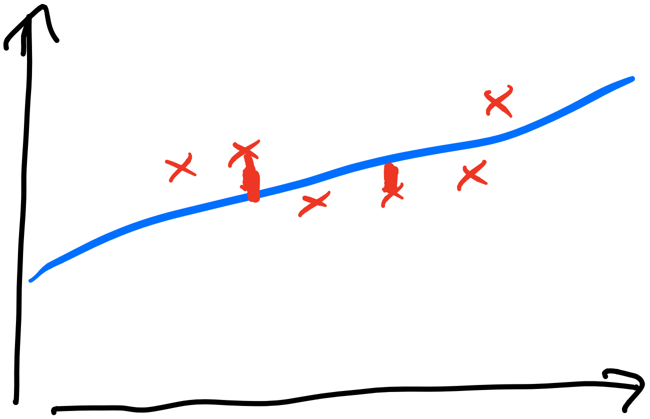
X_1	Y_1
x_2	Y_2
$\vdots$	
X_m	Y_m

Line

$y = \theta_0 + \theta \cdot x$

$\theta = (\theta_0, \theta_1)$

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



Cost Function

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(\underline{h_\theta(x_i)} - \underline{y_i}\right)^2$$

Linear Regression

Parameters $\theta = (\theta_0, \theta_1)$

Hypothesis

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

Cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2$$

Goal:

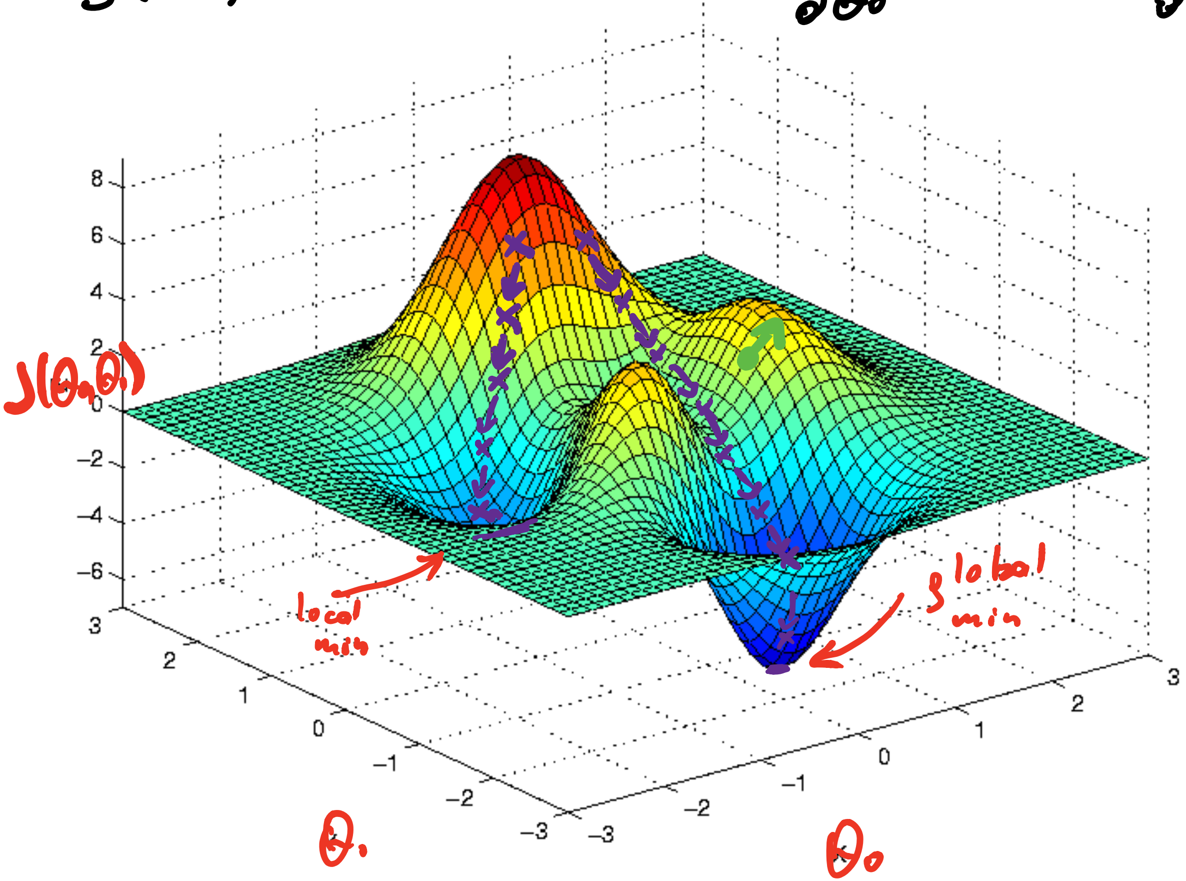
minimize $J(\theta)$

Find $\theta = (\theta_0, \theta_1)$

Gradient Descent

$J(\theta_0, \theta_1)$

$$\frac{\partial J(\boxed{\theta_0, \theta_1})}{\partial \theta_0} \qquad \frac{\partial J(\boxed{\theta_0, \theta_1})}{\partial \theta_1}$$

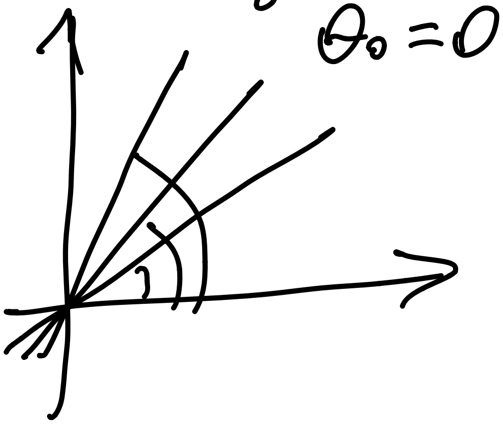


Toy Example

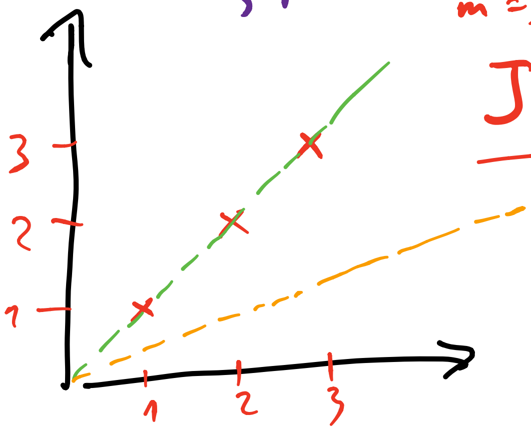
consider lines pass (0,0)

$y = \theta_1 \cdot x$

$\theta_0 = 0$



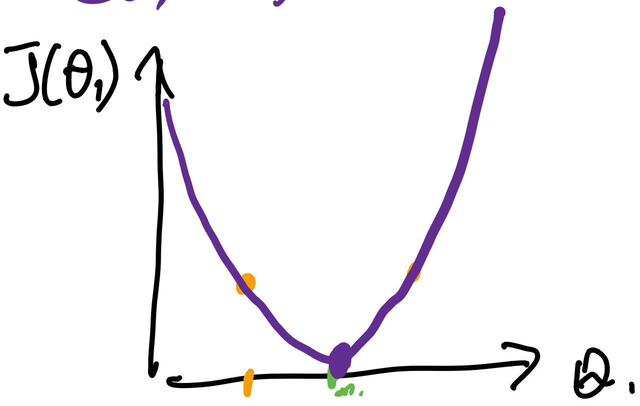
Training points



$m = 3$

$$J(\theta_1) = \frac{1}{6} \sum_{i=1}^{3} (\theta_1 x_i - y_i)^2$$

Cost Function



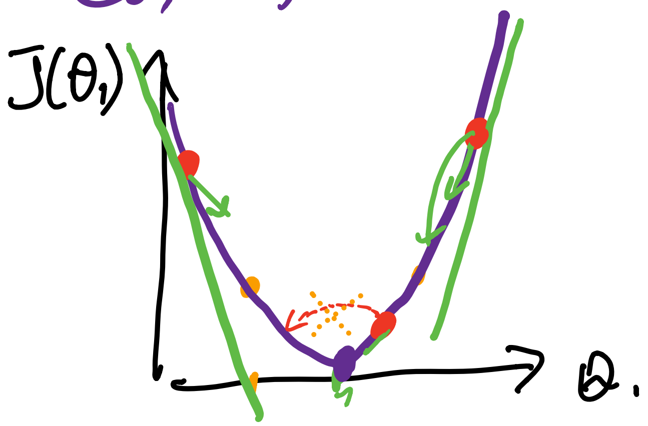
$\theta_1 = 1 \quad J(\theta_1) = 0$

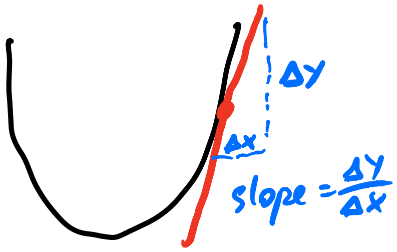
$\theta_1 = 0.5 \quad J(\theta_1) =$

$= \frac{1}{6}\left(\left(\frac{1}{2}\right)^2 + 1^2 + \left(\frac{3}{2}\right)^2\right) \approx$

0.58

Cost function





- function increases (decreases), we want to decrease (increase) θ_1

- function increases rapidly, we want to decrease θ_1 a lot
  slowly, we want to decrease θ a bit

$J'(\theta_1)$ is positive ⇔ J increases at θ_1
negative ⇔ J decreases at θ_1

increases slowly ⇔ derivative is small
fast ⇔ large

$$\theta_1 = \theta_1 - \boxed{\alpha} \cdot J'(\theta_1)$$

Learning rate

$J(\theta_0, \theta_1)$

increases fast $\left(\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}, \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} \right)$

decreases fast

$$\left(\underline{\underline{-\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}}}, \underline{\underline{-\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}}} \right)$$

Gradient Descent

Pick θ_0, θ_1 (Say, $\theta_0 = \theta_1 = 0$)

repeat until converge:

simultaneously update

$$\begin{cases} \theta_0 = \theta_0 - \boxed{\alpha \cdot \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}} \\ \theta_1 = \theta_1 - \alpha \cdot \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} \end{cases}$$

repeat until converge:

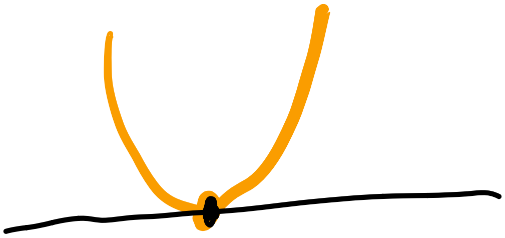
$$\text{update}_0 = \boxed{\alpha} \cdot \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$$

$$\text{update}_1 = \alpha \cdot \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$$

$$\theta_0 = \theta_0 - \text{update}_0$$

$$\theta_1 = \theta_1 - \text{update}_1$$

learning rate



Gradient Descent for Linear Regression

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i)^2$$

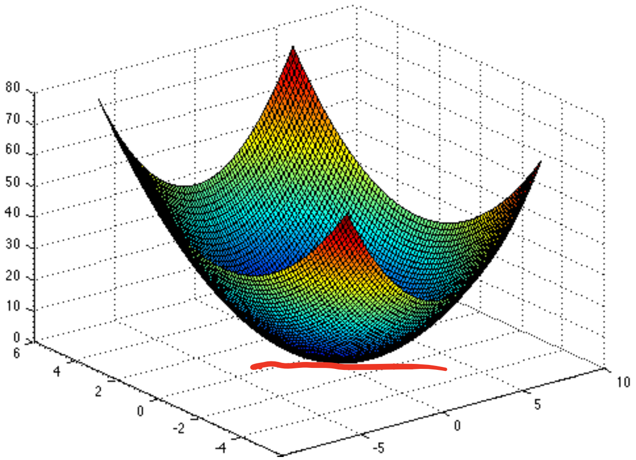
$$\left\{ \begin{aligned} \underline{\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}} &= \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i) \\ \underline{\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}} &= \frac{1}{m} \sum_{i=1}^{m} (\theta_0 + \theta_1 x_i - y_i) \cdot x_i \end{aligned} \right.$$

$\varepsilon = \frac{1}{10^3}$

in practice, converge $\equiv$ update $< \varepsilon$

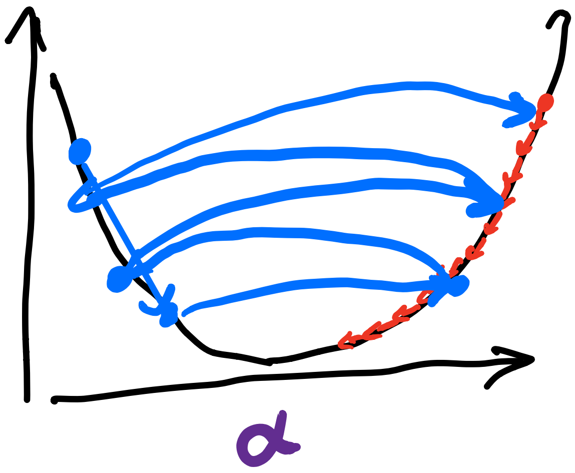
Quadrtic Loss Function

Cost function for linear regression has one local minimum



Learning Rate

Learning rate α



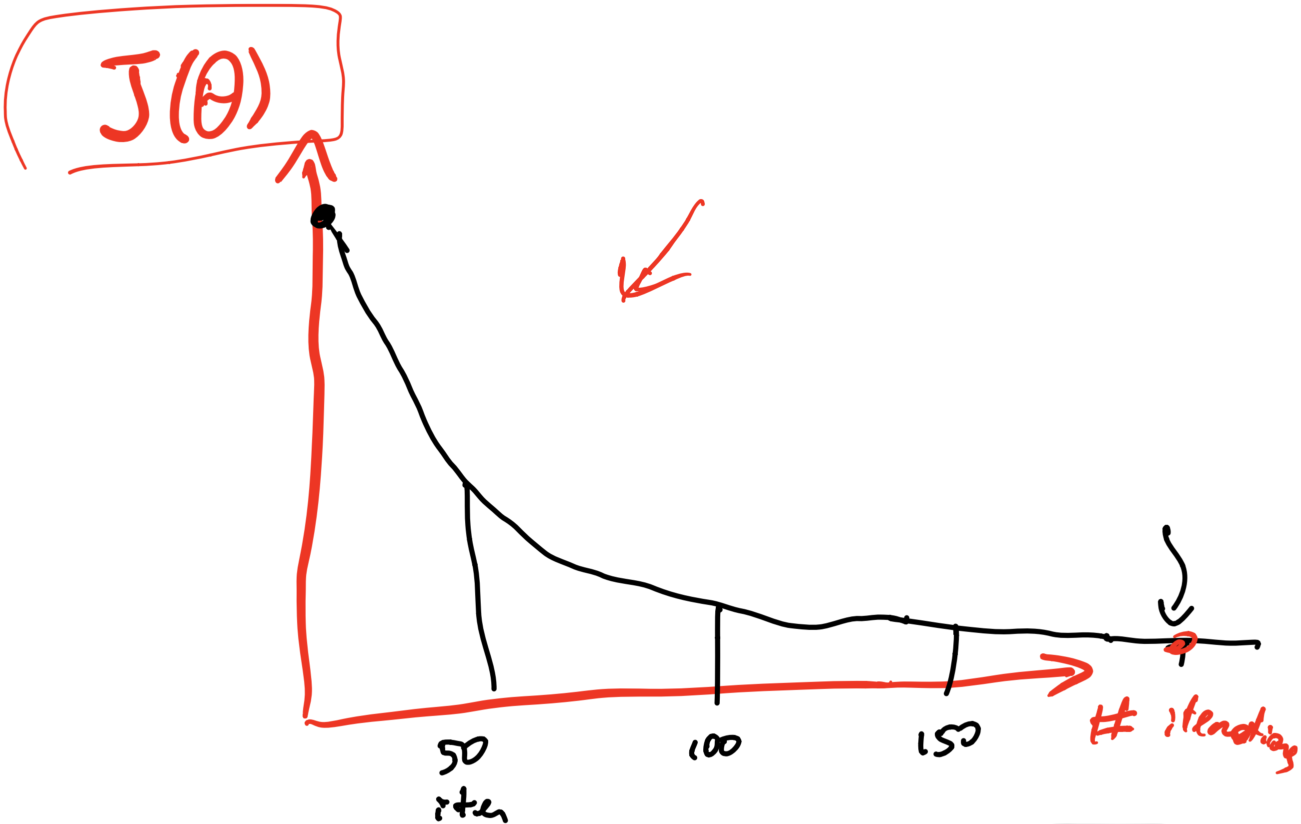
Thm For small enough α, gradient descent will converge.

too small α - slow alg.

too big α - doesn't converge

- - - . .	$\frac{1}{1000}$	$\frac{1}{100}$	($\frac{1}{10}$	$\frac{1}{5}$	$\frac{1}{3}$	$\frac{1}{2}$	1)	10	100	1000 . - - -
	slow	slow						doesnt conv		

"Debug" Gradient Descent



α is too big