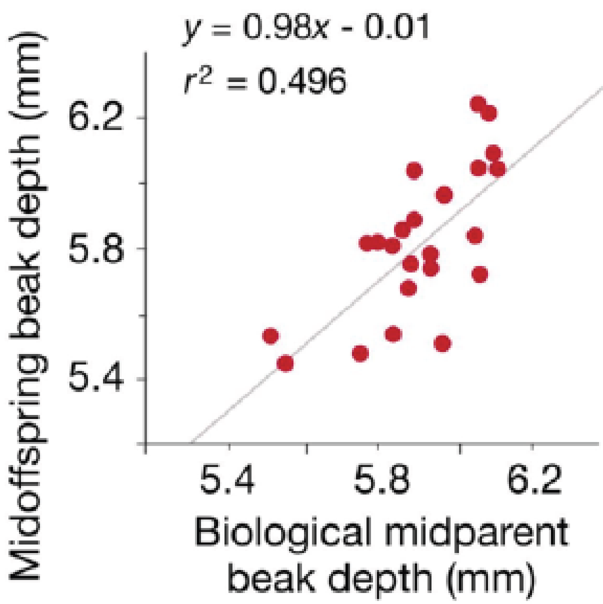
Lecture 2
Linear Regression

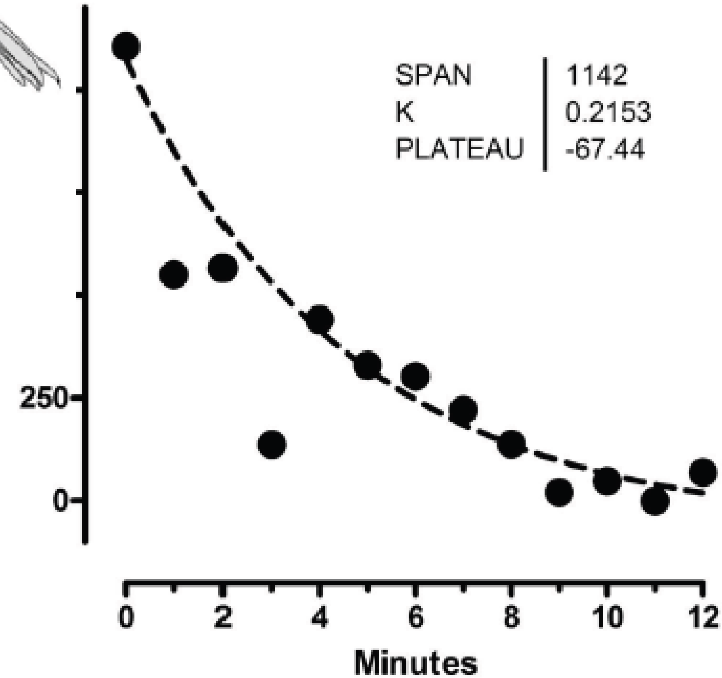
Rui Xia
School of Computer Science & Engineering
Nanjing University of Science & Technology
http://www.nustm.cn/member/rxia

Regression



linear

nonlinear



Data, Input, Output, Relation

- Training data set

Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

One training example $(x^{(i)}, y^{(i)})$, where i denotes the index of the example

Input: Feature Vector $\mathrm{x} = [x_1, x_2]$ Output: y

- Hypothesis: linear model

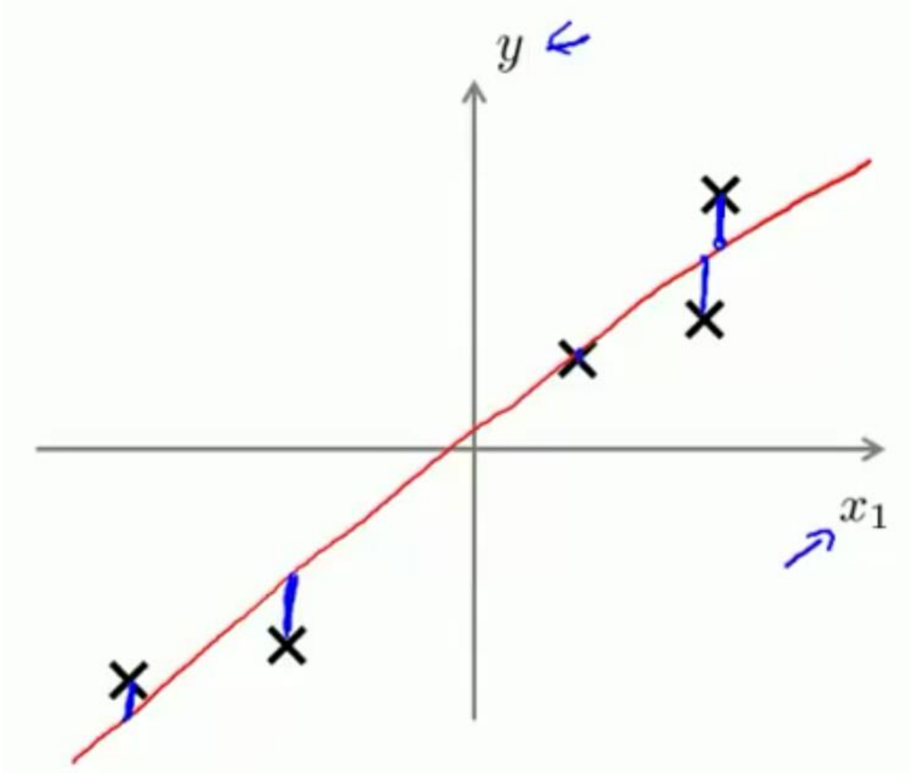
$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

The Least Mean Square (LMS) Algorithm

- Hypothesis

$$h_\theta(x) = \sum_{i=1}^{n} \theta_i x_i = \theta^{\mathrm{T}} x$$

- Parameters θ



- Cost function

$$J_l(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$
$$= \frac{1}{2}\sum_{i=1}^{m}(\theta^{\mathrm{T}} x^{(i)} - y^{(i)})^2$$

- Goal

$$\theta^* = \arg_\theta \min J_l(\theta)$$

Close-form Solution of LMS

- Define

$$X = \begin{bmatrix} -\left(x^{(1)}\right)^{\mathrm{T}} - \\ -\left(x^{(2)}\right)^{\mathrm{T}} - \\ \vdots \\ -\left(x^{(n)}\right)^{\mathrm{T}} - \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

- Then, we have

$$X\theta - y = \begin{bmatrix} \left(x^{(1)}\right)^{\mathrm{T}}\theta \\ \vdots \\ \left(x^{(n)}\right)^{\mathrm{T}}\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(n)}) - y^{(n)} \end{bmatrix}$$

- Now, the LMS cost function

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 = \frac{1}{2}(X\theta - y)^{\mathrm{T}}(X\theta - y)$$

Close-form of LMS Solution

- Calculating LMS gradient by matrix derivatives

$$\nabla_\theta J(\theta) = \nabla_\theta \frac{1}{2}(X\theta - y)^{\mathrm{T}}(X\theta - y)$$
$$= \frac{1}{2}\nabla_\theta\left(\theta^{\mathrm{T}}X^{\mathrm{T}}X\theta - \theta^{\mathrm{T}}X^{\mathrm{T}}y - y^{\mathrm{T}}X\theta + y^{\mathrm{T}}y\right)$$
$$= \frac{1}{2}\nabla_\theta \mathrm{tr}\left(\theta^{\mathrm{T}}X^{\mathrm{T}}X\theta - \theta^{\mathrm{T}}X^{\mathrm{T}}y - y^{\mathrm{T}}X\theta + y^{\mathrm{T}}y\right)$$
$$= \frac{1}{2}\nabla_\theta(\mathrm{tr}\theta^{\mathrm{T}}X^{\mathrm{T}}X\theta - 2\mathrm{tr}y^{\mathrm{T}}X\theta) = X^{\mathrm{T}}X\theta - X^{\mathrm{T}}y$$

- The close-form solution is obtain by letting the gradient equals zero

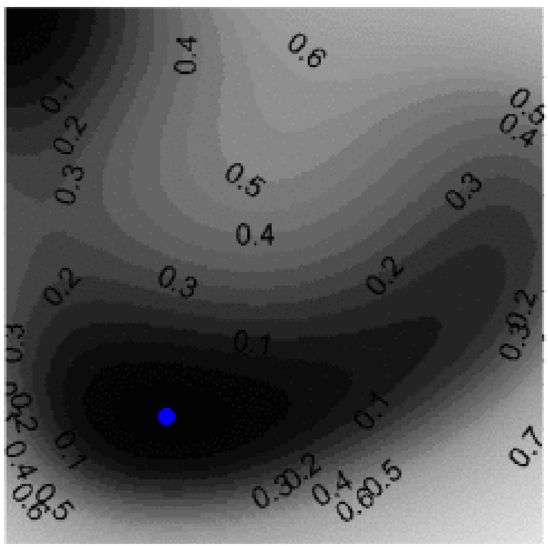
$$\theta^* = \boxed{(X^{\mathrm{T}}X)^{-1}}X^{\mathrm{T}}y$$

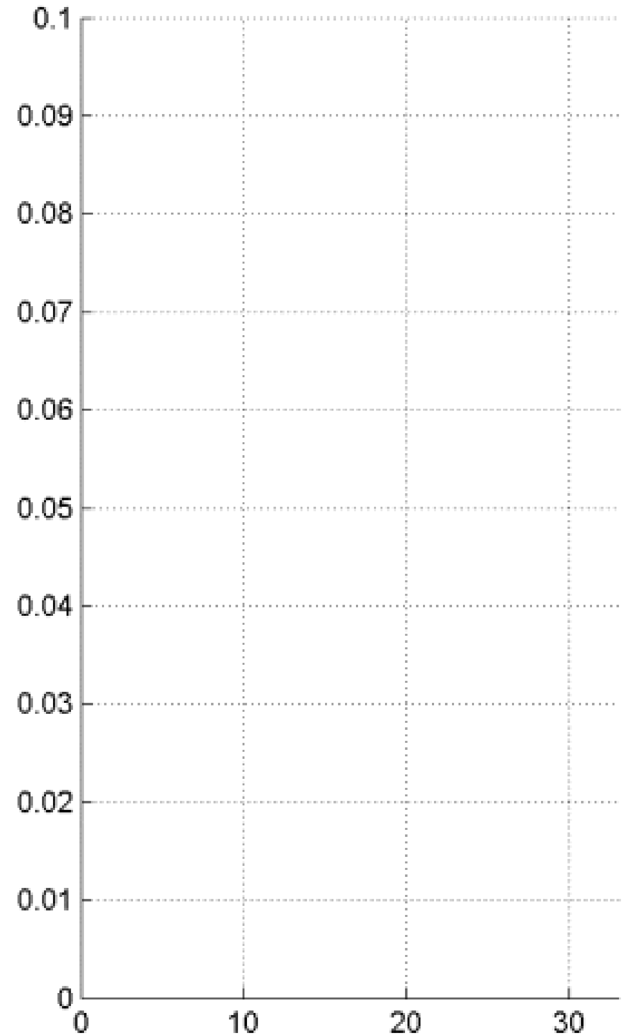
Sometimes very hard to compute!

Gradient Descent for Numeric Optimization

- Gradient descent is a first-order iterative optimization algorithm for finding the minimum of a function $f(\theta)$.
- Key idea:
  - The gradient direction is the direction that the function value increases the fastest.
- Optimization Process:
  - Start at a initial position (i.e., initial parameter $\theta^{(0)}$)
  - At current position $\theta^{(t)}$, repeat till convergence
    - Compute the gradient at current position: $\nabla_\theta f(\theta)|_{\theta=\theta^{(t)}}$
    - Move to the next position along the opposite direction of the gradient: $\theta^{(t+1)} = \theta^{(t)} - \alpha \cdot \nabla_\theta f(\theta)|_{\theta=\theta^{(t)}}$, where α is the learning rate
    - $t = t + 1$

A Dvnamic Illustration of Gradient Descent





Gradient Descent for Linear Regression

- Gradient

$$\frac{\partial J_l(\theta)}{\partial \theta} = \frac{1}{2}\frac{\partial}{\partial \theta}\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$= \frac{1}{2} \cdot 2 \sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot \frac{\partial}{\partial \theta}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)$$

$$= \sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\frac{\partial}{\partial \theta}\left(\theta^T x^{(i)}\right)$$

$$= \sum_{i=1}^{n}\boxed{\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}}$$

"Error · Feature"

- Gradient Descent (GD) Optimization

$$\theta := \theta - \alpha\frac{\partial}{\partial \theta}J_l(\theta) = \theta - \alpha\sum_{i=1}^{n}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

Project: Nanjing Housing Price Prediction

- Given history data

  Year x = [2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013]

  Price y = [2.000, 2.500, 2.900, 3.147, 4.515, 4.903, 5.365, 5.704, 6.853, 7.971, 8.561, 10.000, 11.280, 12.900]

- Assumption: the price and year are in a linear relation, thus they could be modeled by linear regression
- Task
  - To get the relationship of x and y by using linear regression, based on 1) close-form solution and 2) gradient descent;
  - To predict the Nanjing housing price in 2014.

Any Questions?