L 1

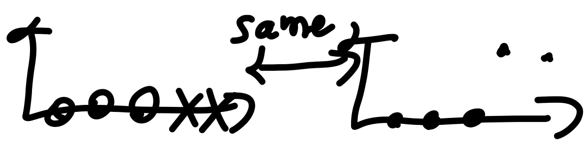
ML : machine learn with task, performance, experience

1. Supervised learning : (x,y) E, x → y

① regression : y → continuous

logistic regression : seperate good/bad

② classification : y → discrete



CV : digitize → learn neural

→ execute

ML decisions many : strategic important

never work first-time : debug strategy important

2. Deep learning:

3. unsupervised learning: label output X
→ find interesting clustering

e.g. cocktail party problem

4. reinforce learning: do stuff → reward

e.g. dog training

L2

Linear regression

$\boxed{\text{Train set } T}$

$\downarrow$

$\boxed{\text{feed algorithm}}$

$\downarrow$ function

eg.

size $\rightarrow$ $\boxed{h}$ $\rightarrow$ house price

hypothesis

How to represent h ?

$$h(x) = \sum_{j=0}^{n} \theta_j x_j \quad , \quad x_0 = 1$$

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

θ : parameter

m : # training examples

x : input / features

y : output / target variable

(x, y) : train examples

$(x^{(i)}, y^{(i)})$ = ith example

n : # features

Goal : ? classify performance?

Linear regression: $J(\theta) = \frac{\min}{2}\left(\sum_{i=1}^{m}(h(x) - y)^2\right)$

↓

Gradient descent : 1. start with θ

2. change θ to θ'

? no local minima?

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

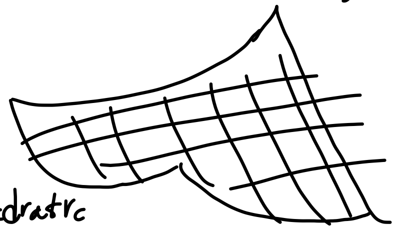
learning rate

$$\frac{\partial}{\partial \theta_j}(J(\theta)) = (h_\theta(x) - y)x_j$$

Repeat until converge

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

$J(\theta)$:

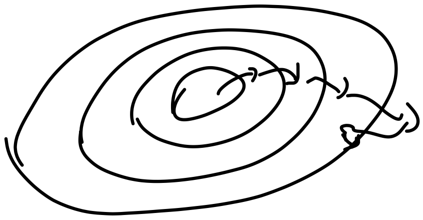


a bowl

↓

because quadratic

so no local minima



level surfaces

α value : ↓ → slow

↑ → overshoot → slow

? make sure one → one ? how?

no need I think

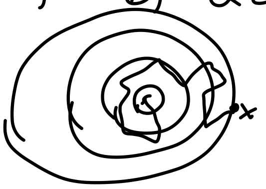
Batch gradient descent : disadvantage : m↑ → slow

Alternative : Stochastic gradient descent

Repeat

For i to m

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



Never converge !

switch stocastic to batch?

no. instead ↓α

$$\nabla_\theta J(\theta) = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_2} \end{bmatrix}$$

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{1,1}} & \frac{\partial f}{\partial A_{1,2}} \\ \vdots & \vdots \end{bmatrix} \text{ if } f : \mathbb{R}^{m \times n} \to \mathbb{R}$$

$$\nabla_\theta J(\theta) = \vec{0} \to \text{global min}$$

$$f(A) = \operatorname{tr} AB, \quad \nabla_A f(A) = B^T$$

$$\nabla_A \operatorname{tr} A A^T C = CA + C^T A$$

$$\vec{X} = \begin{bmatrix} — & x^{(1)T} & — \\ — & x^{(2)T} & — \\ & \vdots & \end{bmatrix}$$

similar for $\vec{y}$

$$J(\theta) = \frac{1}{2}(X\theta - y)^T (X\theta - y)$$

$$\nabla_\theta J(\theta) = X^T X\theta - X^T y \underline{\text{set}} \vec{0}$$

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

must exist

L3

Locally weighted regression $\Rightarrow$ fit non-linear

parametric / non-parametric learning algor-ithm

$\swarrow$ fixedset of (θ_i)

$\downarrow$ Size of cset of θ_i) increases

Locally weighted — regression

$$J(\theta) = \sum_{i=1}^{m} w^i (y^{(i)} - \theta^T x^{(i)})^2$$

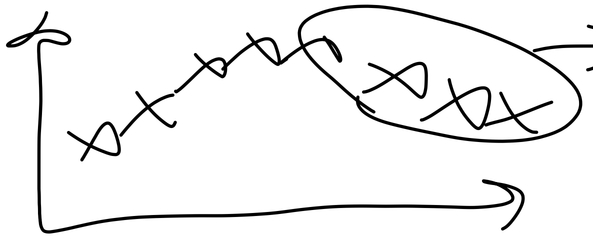
$$w^i = e^{-(x^i - x)^2 / 2\tau^2}$$

τ $\rightarrow$ bandwidth

$\rightarrow$ decide width

if $|x^{(i)} - x| \rightarrow 0$, $w^i \rightarrow 1$

if $|x^{(i)} - x| \rightarrow \infty$, $w^{(i)} \rightarrow 0$



Probabilistic Interpretation

why LS? Assume ① $y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$

error

② $\varepsilon^{(i)} \sim N(0, \sigma^2)$

Assume ② IID → independent & identically distributed

$$\Rightarrow p(y^{(i)} | x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

parametrization, θ not random variable

likelyhood

$$L(\theta) = p(\vec{y} | \vec{x}; \theta)$$

$$= \prod_{i=1}^{m} p(y^{(i)} | x^{(i)}; \theta)$$

likelyhood → fixed $\vec{x}$, vary θ → likelyhood of θ

prob → fixed θ, vary $\vec{x}$ → prob of data

Log likelyhood

$$\ell(\theta) = \log \mathcal{L}(\theta)$$

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} + \sum_{i=1}^{m} -\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}$$

maximize $\ell(\theta)$ → minimize

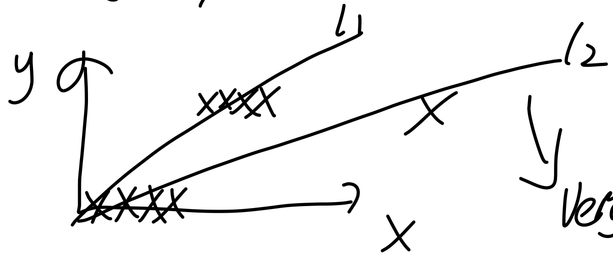
⇒ maximal likelyhood to LSR

Non IID → not bother to have better computation

Maximal likelyhood estimation

classification problem

$y \in \{0,1\}$



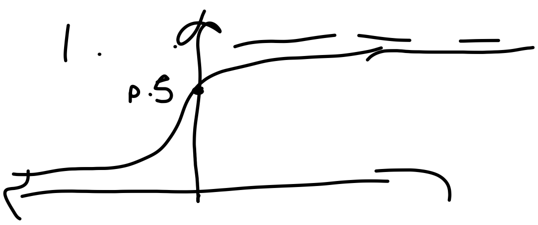
very bad

so linear regression bad

Logistic regression

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$ (generalized linear models)

"singmoid" or "logistic function"



$$P(y|x;\theta) = h^y (1-h)^{1-y}$$

for $y \in \{0,1\}$

$$\ell = \log \mathcal{L}(\theta) = \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))$$

To maximize likelyhood : use ~~Batch~~

$$\theta_j = \theta_j + \frac{\partial}{\partial \theta_j} \ell(\theta) \cdot \alpha$$

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

all for GLM

There is no local minima/maxima for the log function, so for logistic function

Sadly no normal equation

Newton's method

$\ell'(\theta) = 0 \Rightarrow$ maxima

$$\theta^{(1)} := \theta^{(0)} - \Delta$$

$$\Delta = \frac{f(\theta^{(0)})}{f'(\theta^{(0)})}$$

$$\theta^{(t+1)} := \theta^{(t)} - \frac{\ell(\theta^{(t)})}{\ell'(\theta^{(t)})}$$

"Quadratic error"

$\downarrow$

significant digits doubles on one iteration

when θ is a vector

$$\theta^{(t+1)} := \theta^{(t)} + H^{-1} \nabla_\theta \ell$$

actually

H : Hessian matrix $\rightarrow \nabla_\theta^2$

H_{ij} : $\dfrac{\partial \ell^2}{\partial \theta_i \partial \theta_j}$

H^{-1} equivalent to $\dfrac{1}{A}$

Newton pro: fast cons: matrix large then bad.

Created with Notewise

L4

Perceptron algorithm

$$g(z) = \begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$$

$$h_\theta(x) = g(\theta^T x) \qquad \theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)}$$

0 : right $\quad y^{(i)} = 1$

± 1 : if wrong $\quad y^{(i)} = 0$