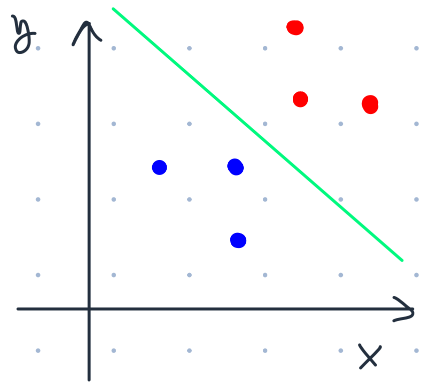
ANNOUNCEMENTS

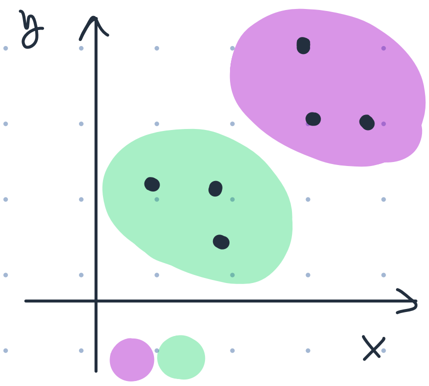
1. HW3 due Friday, 11:59 pm late due Sunday – **NO** extensions beyond late due
2. Prelim conflict form – fill by 3:30 pm today!
3. P3 due changed from 03/04 → 03/05 11:59 pm
4. Prelim logistics – will also be posted to Ed later!
5. HW4 to be released w/ solutions (soon!)

TIME TO TURN YOUR NON-NOTE-TAKING DEVICES OFF!

So FAR



vs.

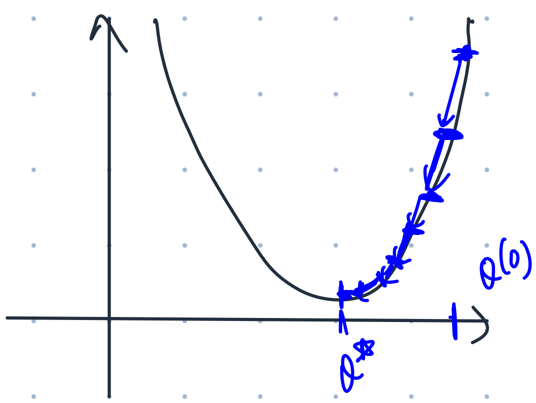


REGIME

Supervised learning — has labels
Unsupervised learning — no labels

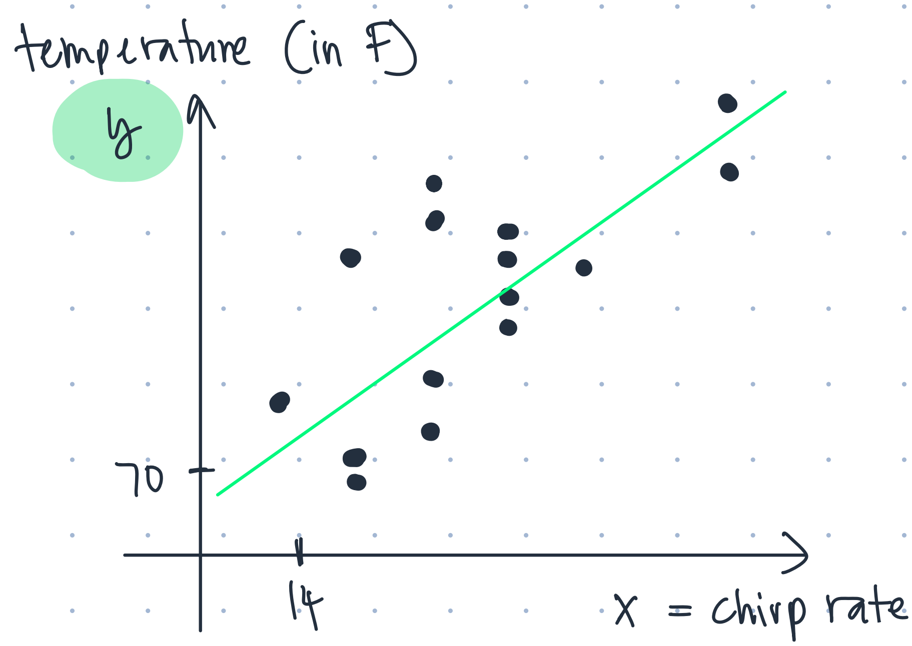
Estimating "cost" function — MLE, MAP

OPTIMIZATION — Iterative approaches : GD, AdaGrad, momentum, Newton



<u>TODAY</u>: Linear regression — instead of a discrete class, we wish to predict a continuous "y"

(supervised setting)



temperature = $\theta_0 + \theta_1$ chirp rate + ϵ — unmodeled noise

<u>GOAL</u>: Can we recover the "green" line?

temperature = $\theta_0 + \theta_1 \times$ chirp rate

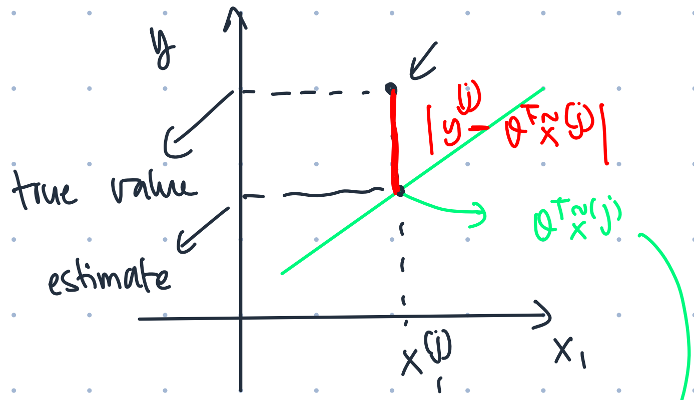
we are trying to learn some "h"

$$h(x^{(j)}; \theta) = \theta_0 \cdot 1 + \theta_1 x_1^{(j)} + \theta_2 x_2^{(j)} + \ldots + \theta_d x_d^{(j)}$$

$$= \sum_{i=0}^{d} \theta_i x_i^{(j)} = \theta^T \tilde{x}^{(j)}, \quad \tilde{x}^{(j)} = \begin{bmatrix} x_0^{(j)} = 1 \\ x_1^{(j)} \\ \vdots \\ x_d^{(j)} \end{bmatrix}$$

$x_0^{(j)} = 1$

COST FUNCTION $J(\theta)$



<u>GOAL</u>: minimize

$$|y^{(j)} - \theta^T \tilde{x}^{(j)}|$$

for all "$\underline{j}$":

$$\theta_0 + \theta_1 \text{ chirp rate} = \theta^T \begin{bmatrix} 1 \\ \text{chirp rate} \end{bmatrix}$$

$$J(\theta) = \frac{1}{2n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)} \right)^2$$

average squared deviation b/w observed $y^{(j)}$, predicted $\theta^T \tilde{x}^{(j)}$

HOW DO WE FIND "OPTIMAL" θ? to minimize $J(\theta)$

Nick's idea – use GD

starting with some $\theta^{(0)}$

$$\theta^{(k+1)} = \underbrace{\theta^{(k)} + \alpha}_{\text{step size!}} \left(-\nabla J(\theta^{k})\right) \longrightarrow \text{move in steepest descent direction}$$

NEED: $\nabla J(\theta^{k})$, or more generally, $\nabla J(\theta)$

$$J(\theta) = \frac{1}{2n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right)^2$$

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

$$\frac{\partial J}{\partial \theta_i} = \frac{\partial}{\partial \theta_i} \frac{1}{2n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right)^2$$

$$= \frac{1}{\cancel{2}n} \sum_{j=1}^{n} \frac{\partial}{\partial \theta_i} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right)^2$$

$$= \cancel{2}\left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right) \underbrace{\frac{\partial}{\partial \theta_i}\left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right)}_{-\frac{\partial}{\partial \theta_i} \theta^T \tilde{x}^{(j)}}$$

$$=$$

$$\frac{\partial}{\partial \theta_i} \theta^T \tilde{x}^{(j)} = \theta_0 + \theta_1 x_1^{(j)} + \dots + \theta_i x_i^{(j)} + \dots + \theta_d x_d^{(j)}$$

$$= x_i^{(j)}$$

$$\frac{\partial J}{\partial \theta_i} = \frac{-1}{n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right) x_i^{(j)}$$

$$\theta_i^{(k+1)} = \theta_i^{(k)} + \alpha \left(\frac{1}{n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right) x_i^{(j)}\right) \longrightarrow \text{for one "}\theta_i\text{"}$$

$$\Rightarrow \theta^{(k+1)} = \theta^{(k)} + \frac{\alpha}{n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right) x^{(j)}$$

GD might not converge to global optima if >1 optima

$$J(\theta) = \frac{1}{2n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T x^{(j)}\right)^2$$ is "nice"

Convex, ONE minima, GD will converge!

Q. What is the cost of running one update $(k \to k+1)$ of GD, in terms of n = dataset size, d = feature dimension?

$$\theta^{(k+1)} = \theta^{(k)} + \frac{\alpha}{n} \boxed{\sum_{j=1}^{n} \left(y^{(j)} - \theta^T x^{(j)}\right) x^{(j)}}$$

$\left(y^{(j)} - \theta^T x^{(j)}\right) x^{(j)} \to O(d)$

$O(nd)$

Aditya says "$O(nd)$"

$\Rightarrow$ Seasons change, GD No update :(

Computational complexity of $\underline{\text{one}}$ update step of GD

Q: CAN WE DO BETTER?

the problem!!

$$\Rightarrow \theta^{(k+1)} = \theta^{(k)} + \frac{\alpha}{n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T x^{(j)}\right) x^{(j)}$$

It's <u>ok</u> to take an "approximate" step, so long as it's faster!

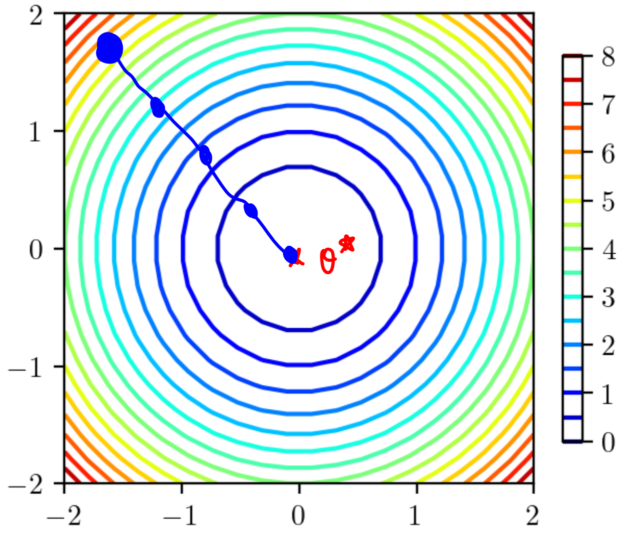
$$\nabla J = \frac{1}{n} \sum_{j=1}^{n} \left(y^{(j)} - \theta^T \tilde{x}^{(j)}\right) \tilde{x}^{(j)}$$

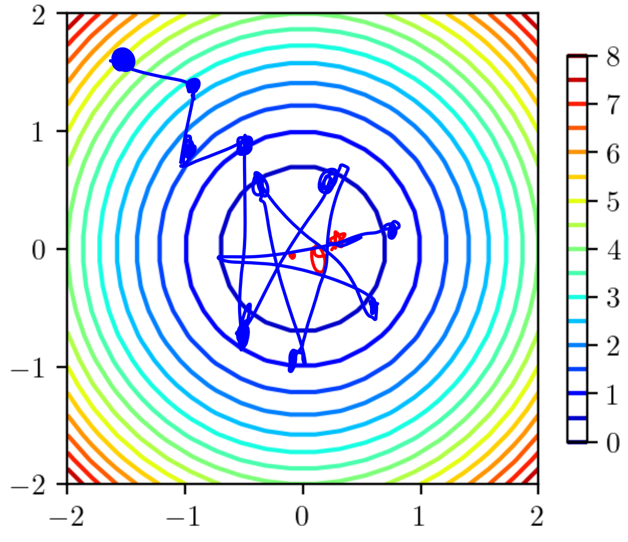
1. Sample some $\tilde{j}$ uniformly at random from traing set
2. Compute $\nabla_{\tilde{j}} J$ as $\left(y^{(\tilde{j})} - \theta^T \tilde{x}^{(\tilde{j})}\right) \tilde{x}^{(\tilde{j})}$

Jay says "$O(d)$" for this new approach!

<u>"stochastic"</u> GD

GD vs. "stochastic" GD





"Stochastic"

Another idea: closed-form solution to "least-squares" objective

"Alex" — take derivative of J w.r.t θ set 0, solve for θ?

design matrix

$$X = \begin{bmatrix} — x^{(1)T} — \\ \vdots \\ — x^{(n)T} — \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}$$

$x^{(1)T}$ — d^{th}-dimensional

$n \times d+1$

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

$$\|u\|_2^2 = u^T u = u_1^2 + \cdots + u_n^2$$

$$\frac{1}{2n}(y - X\theta)^T(y - X\theta) = \frac{1}{2n}\left[\left(y^{(1)} - x^{(1)T}\theta\right)^2 + \cdots + \left(y^{(n)} - x^{(n)T}\theta\right)^2\right]$$

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

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

$$= \cancel{y^T y} - \theta^T X^T y - y^T X\theta + \theta^T X^T X\theta$$

ignore for now

0

$$\nabla_\theta J \overset{\text{set}}{=} 0 \Rightarrow \nabla_\theta J = \nabla_\theta \; -\theta^T X^T y - y^T X\theta + \theta^T X^T X\theta$$

$\nabla_\theta\, \theta^T x = x$

$\nabla_\theta\, \theta^T A\theta = (A + A^T)\theta$

$a \cdot b = b \cdot a = a^T b = b^T a$

$\theta^T X^T y = y^T X\theta$

$$\nabla_\theta \left(-2\theta^T X^T y + \theta^T X^T X\theta\right)$$

$$= -2X^T y + \left(X^T X + (X^T X)^T\right)\theta$$

$$= -2X^T y + 2X^T X\theta$$

$$\nabla_\theta J \overset{\text{set}}{=} 0 \rightarrow -2X^T y + 2X^T X\theta = 0$$

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

Q: why choose GD / SGD if we have closed-form solution?

Inverting $X^T X$ is problematic — can be near singular or singular

$(X^T X)^{-1} \longrightarrow O(d^3)$

forming $X^T X \longrightarrow O(nd^2)$

$\theta^* \rightarrow O(nd^2 + d^3)$

vs SGD $= O(d)$

Revisiting least squares $J(\theta)$ — probabilistic view

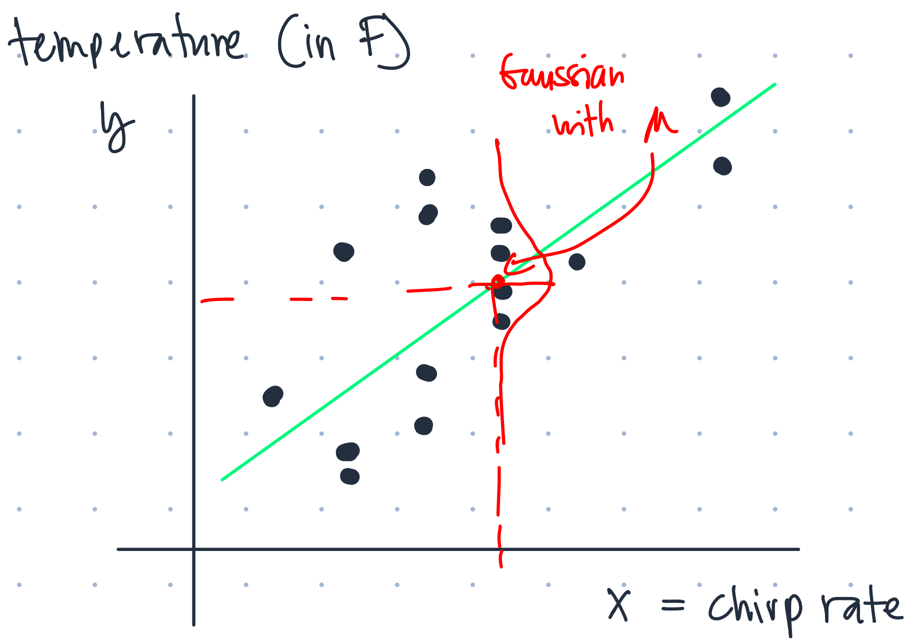
How was data generated?

$$y^{(j)} = \theta^T \tilde{x}^{(j)} + \epsilon^{(j)}$$

Assume: $\epsilon^{(j)} \sim \mathcal{N}(0, \sigma^2)$

$$p(\epsilon^{(j)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^{(j)} - 0)^2}{2\sigma^2}\right)$$

$$p(y^{(j)} \mid x^{(j)}; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(j)} - \theta^T \tilde{x}^{(j)})^2}{2\sigma^2}\right)$$



GOAL: Estimate "θ" to maximize likelihood of data

$$L(\theta; X, y) = P(D;\theta) = \prod_{j=1}^{n} P(x^{(j)}, y^{(j)}; \theta)$$

constant

$$= \prod_{j=1}^{n} p(y^{(j)} | x^{(j)}; \theta)\, p(x^{(j)})$$

under optimization $\equiv \prod_{j=1}^{n} p(y^{(j)} | x^{(j)}; \theta)$

$$L(\theta) = \arg\max_{\theta} \prod P(y^{(j)} | x^{(j)}; \theta)$$

$\sim \mathcal{N}(\theta^T \tilde{x}^{(j)}; \sigma^2)$

$$\ell(\theta) = \log(L(\theta)) = \arg\max_{\theta} \sum_{j=1}^{n} \log p(y^{(j)} | x^{(j)}; \theta)$$

$$= \sum_{j=1}^{n} \log\left(\frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(-\frac{(y^{(j)} - \theta^T \tilde{x}^{(j)})^2}{2\sigma^2}\right)\right)$$

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

constant

arg max

$$= \frac{-1}{2\sigma^2} \sum_{j=1}^{n} (y^{(j)} - \theta^T \tilde{x}^{(j)})^2$$

$$\equiv \arg\min_{\theta} \frac{1}{2\sigma^2} \sum_{j=1}^{n} (y^{(j)} - \theta^T \tilde{x}^{(j)})^2$$

$$\equiv \arg\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} (y^{(j)} - \theta^T \tilde{x}^{(j)})^2$$

cost function $J(\theta)$ (factor of $1/n$)

Also, doesn't depend on $\underline{\sigma^2}$!