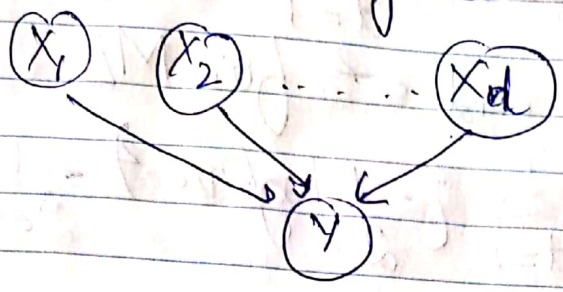
Lecture 09

Case II fixed DAG, complete data, parametrized CPTs

IIA Linear Regression



$\vec{x} \in \mathbb{R}^d$
$y \in \mathbb{R}$

- Gaussian CPT

$$P(Y=y \mid \vec{X}=\vec{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y-\vec{\omega}\cdot\vec{x})^2\right\}$$

$\vec{\omega}$ ↓ weight vector

Note: $E[y|\vec{x}] = \vec{\omega}\cdot\vec{x}$

* Training Data:

$$\left\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_T, y_T)\right\}$$

- Probability of IID data:

$$P(y_1, y_2, \ldots y_T \mid \vec{x}_1, \vec{x}_2, \ldots, \vec{x}_T)$$

(conditional likelihood)

$$= \prod_{t=1}^{T} P(y_t \mid \vec{x}_t)$$

* Log conditional likelihood:

$$\mathcal{L} = \log P(\text{data}) = \sum_{t=1}^{T} \log P(y_t | \vec{x}_t)$$

- Estimate $\vec{\omega}$ and σ^2 for ML.

$$\mathcal{L}(\vec{\omega}, \sigma^2) = \sum_{t=1}^{T} \left\{ -\frac{1}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left[y_t - \vec{\omega} \cdot \vec{x}_t \right]^2 \right\}$$

same as minimizing sum-squared error of linear fit.

- To maximize:

$$0 = \frac{\partial \mathcal{L}}{\partial \omega_\alpha} \rightarrow \alpha = 1, 2, \cdots d = \sum_t \left[\left(\frac{1}{\sigma^2} \right) \cdot \left(y - \vec{\omega} \cdot \vec{x}_t \right) \right] x_{\alpha t}$$

d equations ($\alpha = 1, 2, \ldots d$)
d unknowns ($\omega_1, \omega_2, \ldots \omega_d$)

* Linear Equations:

$$\sum_t y_t x_{\alpha t} = \sum_t (\vec{\omega} \cdot \vec{x}_t) x_{\alpha t} = \sum_{t=1}^{T} \left(\sum_{\beta=1}^{d} (\omega_\beta x_{\beta t}) \right) x_{\alpha t}$$

- In matrix vector form:

$d \times d$ matrix $A_{\alpha\beta} = \sum_{t=1}^{T} x_{\alpha t} x_{\beta t}$ (symmetric)

$d \times 1$ vector $\rightarrow b_\alpha = \sum_{t=1}^{T} y_t x_{\alpha t}$

$$\Rightarrow \boxed{\vec{A} = \sum_t \vec{x}_t \vec{x}_t^T \qquad \vec{b} = \sum_t y_t \vec{x}_t}$$

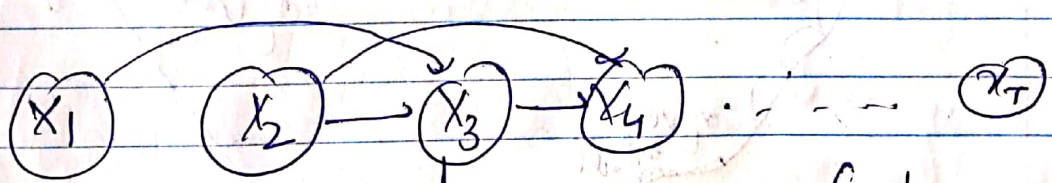
- Set of linear equations

$$\vec{b} = A\vec{w} \xrightarrow{\text{solve}} \boxed{\vec{w}_{ML} = A^{-1}\vec{b}}$$

- Ill-conditioned problems arise when:
  - Input dimensionality exceeds the no. of examples ($d > T$)
  - Inputs not in 'general' position.
  - option: minimum norm solution:
    find $\min \|\vec{w}\|$ such that $\frac{dL}{d\vec{w}}$ vanishes
    (always exist, unique)

- example: time series prediction.

time series $\{x_1, x_2, \ldots, x_T\}$ $x_t \in \mathbb{R}$.

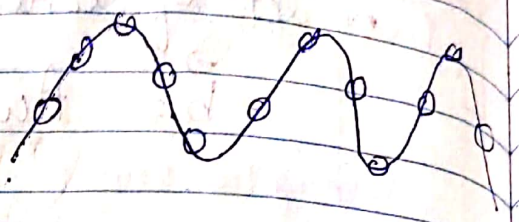


model $x_t \approx \sum_{\alpha=1}^{d} w_\alpha x_{t-\alpha}$ (above $d=2$)

- **Question:** if x_t is a linear combination of (say) x_{t-1} and x_{t-2}, is x_t a linear function of time? **NO**

Ex. $x_t = \sin(\Omega t)$

$$\boxed{x_t = 2(\cos\Omega)\, x_{t-1} - x_{t-2}}$$

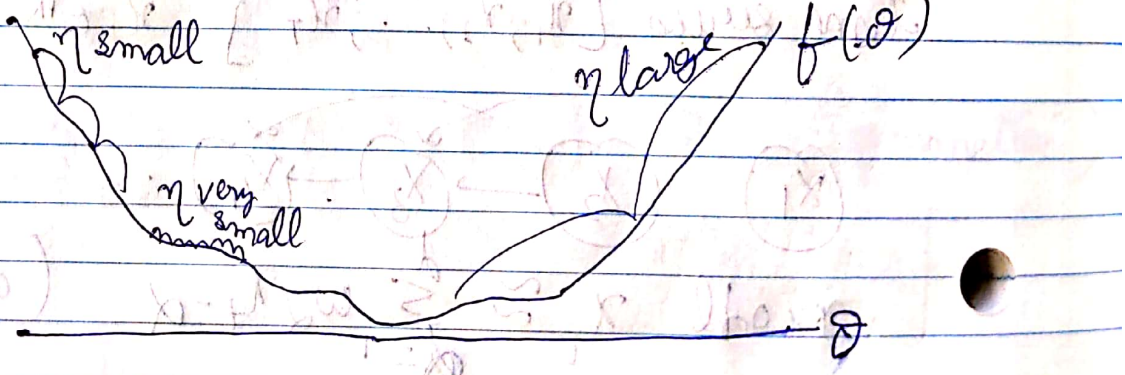


→ **Detour - numerical optimization!**

- How to maximize (or minimize) function $f(\vec{\theta})$ over $\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_d) \in \mathbb{R}^d$
- Not always possible to analytically solve: $\frac{df}{d\vec{\theta}} = \left(\frac{df}{d\theta_1}, \ldots, \frac{df}{d\theta_d}\right) = (0, 0, \ldots 0)$ in closed form.

+ Turn to numerical methods:

1) gradient descent (or ascent)



iterative update rule:

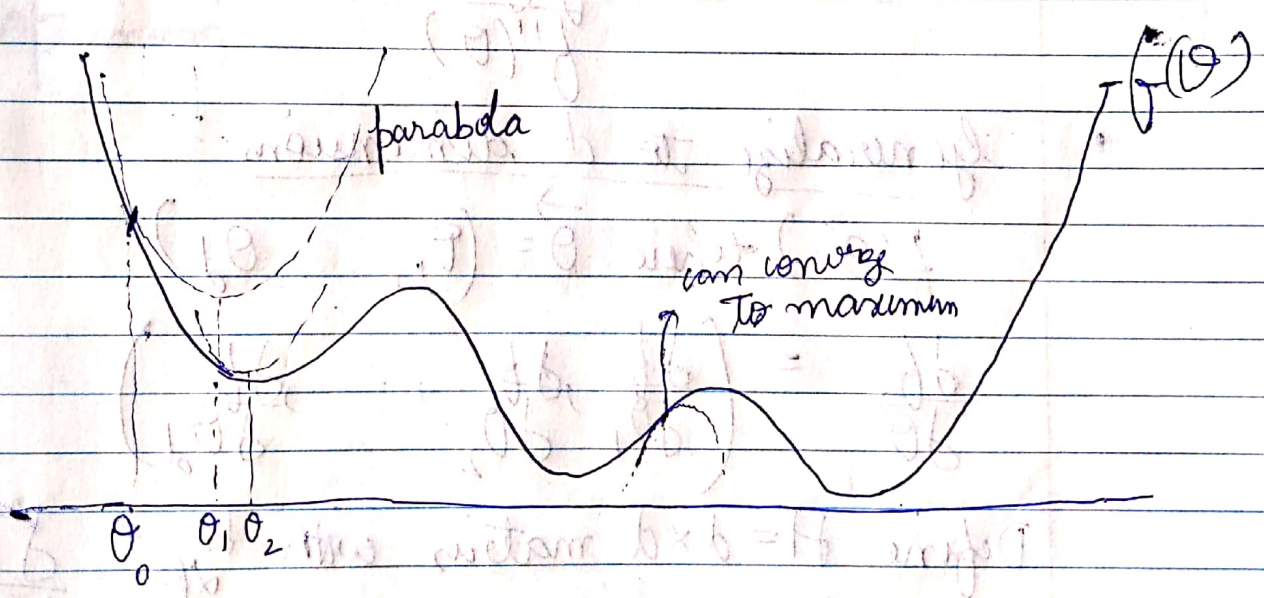
$$\vec{\theta} \leftarrow \vec{\theta} - \eta \left(\frac{df}{d\vec{\theta}} \right) \quad \eta > 0$$

step size, learning rate

Cons: tuning $\eta > 0$ can be tricky.
- no guarantee of monotonic convergence.
- local vs global maximum (minimum)

Pros: - simple, generic procedure for any differentiable function.
- asymptotically converges to some local minimum (or maximum)

2.2 Newton's method:



- Approximate $f(\theta)$ near $\theta = \theta_0$ using parabola

- Taylor series

$$f(\theta) \approx f(\theta_0) + f'(\theta_0)(\theta-\theta_0) + \frac{1}{2} f''(\theta_0)(\theta-\theta_0)^2$$

- Minimize quadratic approximation:

$$0 = \frac{d}{d\theta}\left[f(\theta_0) + f'(\theta_0)(\theta-\theta_0) + \frac{1}{2} f''(\theta_0)(\theta-\theta_0)^2\right]$$

$$0 = f'(\theta_0) + f''(\theta_0)(\theta-\theta_0)$$

$$\Rightarrow \quad \theta_{min} = \theta_0 - \frac{f'(\theta_0)}{f''(\theta_0)}$$

- Iterative update rule:

$$\theta \leftarrow \theta - \frac{f'(\theta)}{f''(\theta)}$$

- Generalize to d dimensions:

$f(\vec{\theta})$ where $\vec{\theta} = (\theta_1, \ldots, \theta_d)$

$$\frac{df}{d\vec{\theta}} = \left(\frac{df}{d\theta_1}, \frac{df}{d\theta_2}, \ldots, \frac{df}{d\theta_d}\right)$$

Define $H = d \times d$ matrix with $H_{\alpha\beta} = \frac{d^2 f}{d\theta_\alpha d\theta_\beta}$

↓

Hessian

(symmetric matrix)

update:

$$\vec{\theta} \leftarrow \vec{\theta} - H^{-1}\left(\frac{\partial L}{\partial \vec{\theta}}\right)$$

matrix inversion

matrix vector multiplication

Pros: 1.) no learning rate to be tuned

2.) converges very fast (when it converges)

Cons: 1.) unstable if far from optimum.

2.) often expensive computation ($O(d^2)$ or invert Hessian matrix.

3.) Can converge to local (not necessarily global) optimum.