Jan 21, 2020

CS 6241: Numerical methods for data science

Numerical methods:

- algo that we put on a computer to "solve a problem"
- usually continuous math, interplay with discrete

Data science

- ??
- analysis of some info that has already been collected for new insight

This class

(1) matrix methods — basic ML, opt; dimensionality redux

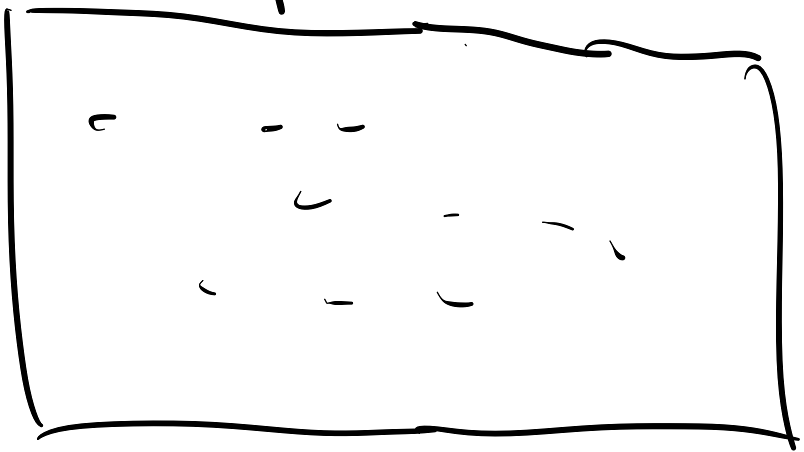
(2) network analysis — ranking; clustering; ML

special topics: GPs / kernel, time series, discrete choice

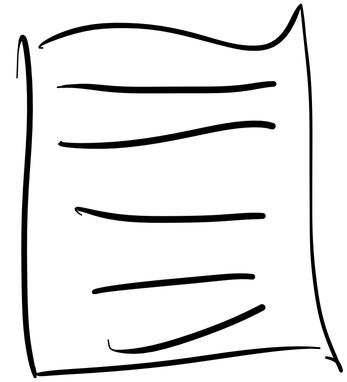
Two important objects: (1) Matrix
(2) Graph

samples	feats	problem
animals in Vet lab	height, weight, food	sick?
patients	prescriptions	cluster new Rx

sparse

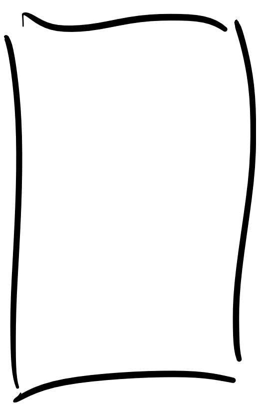


dense



feats

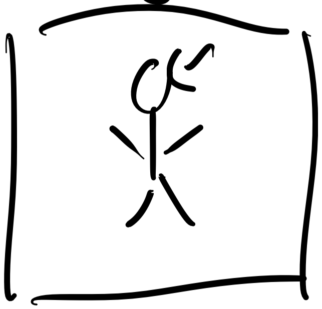
samp



data table

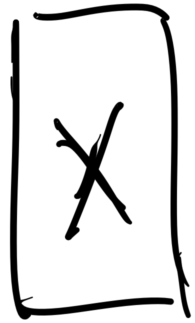
matrices with two-dim structure

images



feats

samps n



$$\bar{X} = \left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right)X$$

$$S = \frac{1}{n-1}\bar{X}^T\bar{X}$$

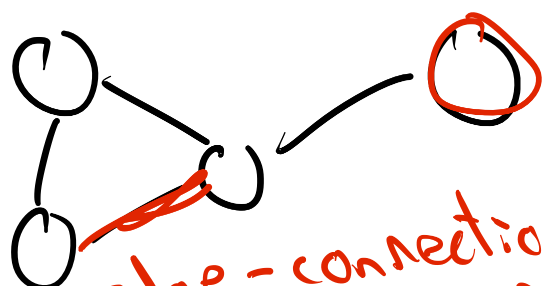
x_j

$$\bar{X}_j = \underbrace{\left(I - \frac{1}{n}\mathbb{1}\mathbb{1}^T\right)X_j}_{\sum_{i=1}^{n} x_{ij}}$$

$$\bar{X}_{ij} = x_{ij} - \frac{1}{n}\sum_{k=1}^{n} x_{kj}$$

---

graphs!



nodes - entities (person)

edge - connection (friendship)

domain	node	edge	problem
sociology	person	friends	recommending friends
ecology	species	carbon flow who-eats-whom	find hierarchy
biology	proteins	interact/ bond	new drug targets

Graphs as matrices

$$A_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \text{ exists} \\ 0 & \text{o/w} \end{cases}$$

$$A^k_{ij} = \#\text{paths from } i \text{ to } j \text{ of length } k$$

Administrivia

- PhD level (elective) course
  4220, 6210
- cs.cornell.edu/courses/cs6241/2020sp

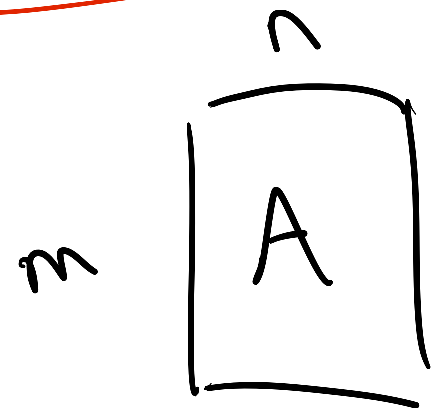
Coursework
- ~3 homeworks (25%)
- reaction paper (20%)
- project
  - proposal (10%)
  - progress report (15%)
  - final report (30%)

groups of 1, 2, or 3

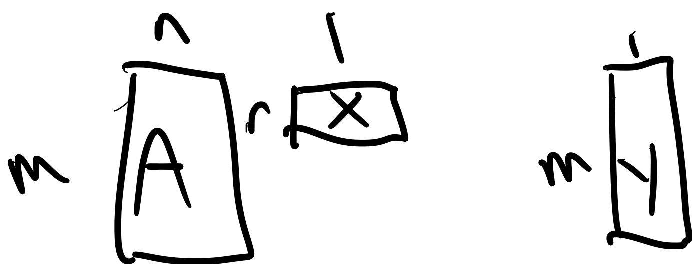
① math
② code
③ data analysis

Numerical linear algebra basics

A $m \times n$ matrix
real-valued



$y = Ax$



$A(x+y) = Ax + Ay$ $A(\alpha x) = \alpha A x$

$$\underset{m \times n}{C} = \underset{m \times k}{A} \cdot \underset{k \times n}{B}$$

$$C_{ij} = \sum_{r=1}^{k} a_{ir} b_{rj}$$

1. norms $\|x\|_2$ $\|A\|_2$
2. eigenvalues $Ax = \lambda x$

   $x \neq 0$ eigenvector

   λ eigenvalue
3. $Ax = b$

   known
4. Factorize $A = BC$ $\boxed{A} = \boxed{B}\boxed{C}$

Norms: how we measure

$x \in \mathbb{R}^n$ $\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

$A \in \mathbb{R}^{m \times n}$ $\quad \|A\|_F = \sqrt{\sum_{ij} A_{ij}^2}$

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$$

all norms: $\|x+y\| \leq \|x\| + \|y\|$

some norms are submultiplicative

Claim: $\|Ax\|_2 \leq \|A\|_2 \|x\|_2$

Proof: $x = 0$ ✓ $\quad x \neq 0$

$$\frac{\|Ax\|_2}{\|x\|_2} \leq \|A\|_2 = \sup_{y \neq 0} \frac{\|Ay\|_2}{\|y\|_2}$$

✓

Claim: $\|AB\|_2 \le \|A\|_2 \|B\|_2$

Proof: $$\|AB\|_2 = \sup_{x \neq 0} \frac{\|A(Bx)\|_2}{\|x\|_2} \le \|A\|_2 \sup_{x \neq 0} \frac{\|Bx\|_2}{\|x\|_2} = \|A\|_2 \|B\|_2$$

Eigenvalues

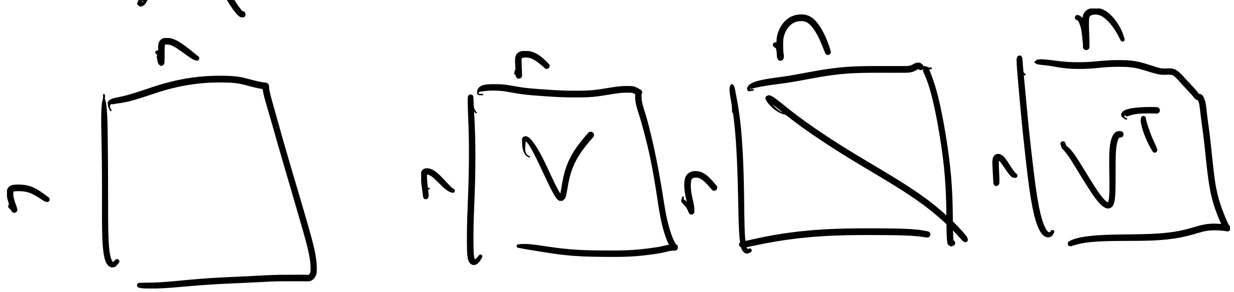
$$Ax = \lambda x \quad x \neq 0 \quad \lambda \in \mathbb{R}$$

$n \boxed{A}$ must be square

A is symmetric if $A^T = A$ $\quad (A^T)_{ij} = A_{ji}$

if A is symm, then

$$A = V \Lambda V^T \qquad V^T V = I$$



$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}$$

$$AV = V \Lambda V^T V = V \Lambda$$

$$AV_{:,j} = \lambda_j V_{:,j} \qquad A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

$$\min \frac{x^T A x}{x^T x} = \lambda_n \text{ (smallest eval)}$$

$$x = v_n$$

SVD singular value decomposition

$$\underset{m \times n}{A} = \underset{m \times n}{U}\ \underset{n \times n}{\Sigma}\ \underset{n \times n}{V^T}$$

$$U^T U = I$$

$$V^T V = I$$

$$\Sigma = \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{pmatrix} \qquad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$$

Numerical opt. basics

$$\min \; f(x)$$

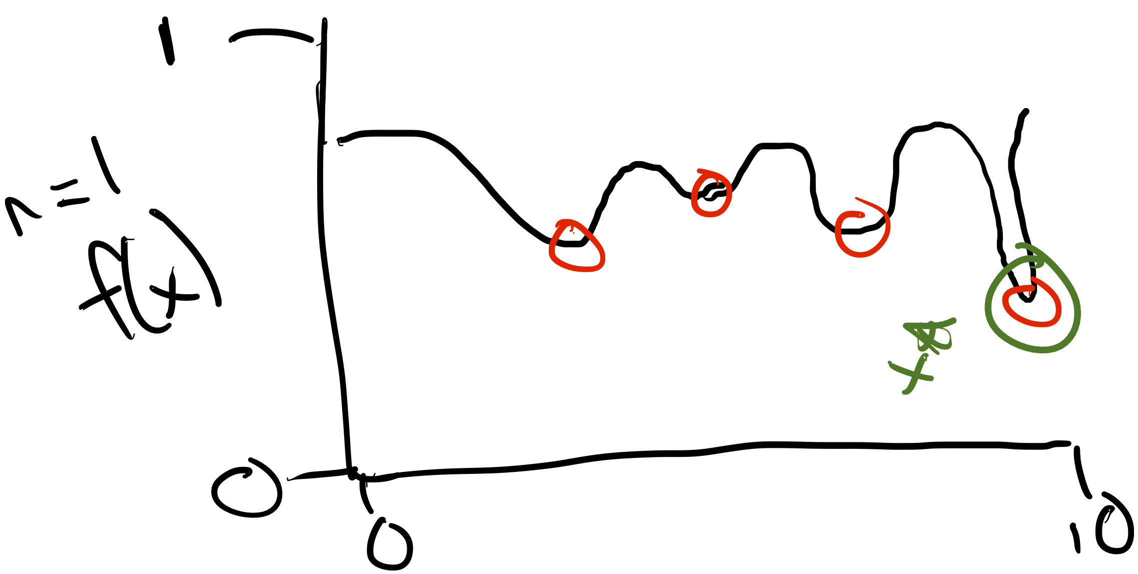
$f: \mathbb{R}^n \to \mathbb{R}$ objective function

$$\text{subject to} \quad x \in \Omega$$

constraint set

Example: $\min \; x^T c$ linear program

$$\text{s.t.} \quad Ax \le b$$



- local minimizer x^*

$f(x^*) \le f(x)$ nearby

- global minimizer

$f(x^*) \le f(x)$

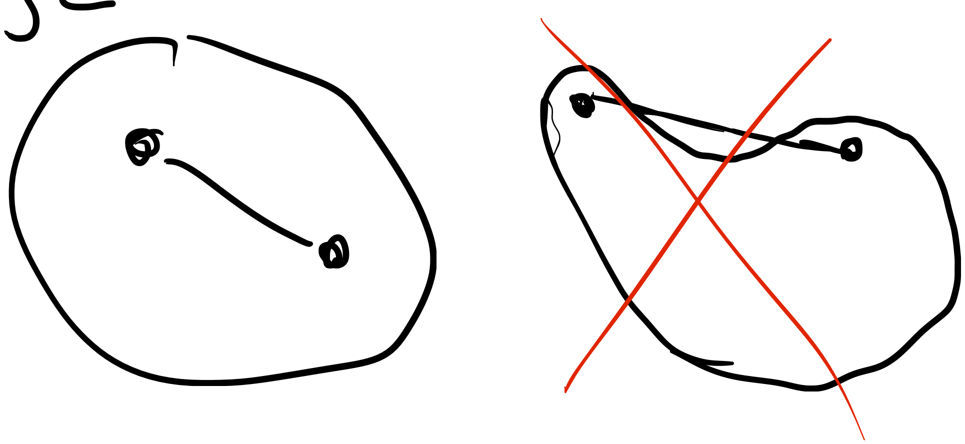
often hard to find global min

doing better with <u>structure</u>

Example: convexity

$$\min \; f(x) \quad \text{s.t.} \quad x \in \Omega$$

Ω

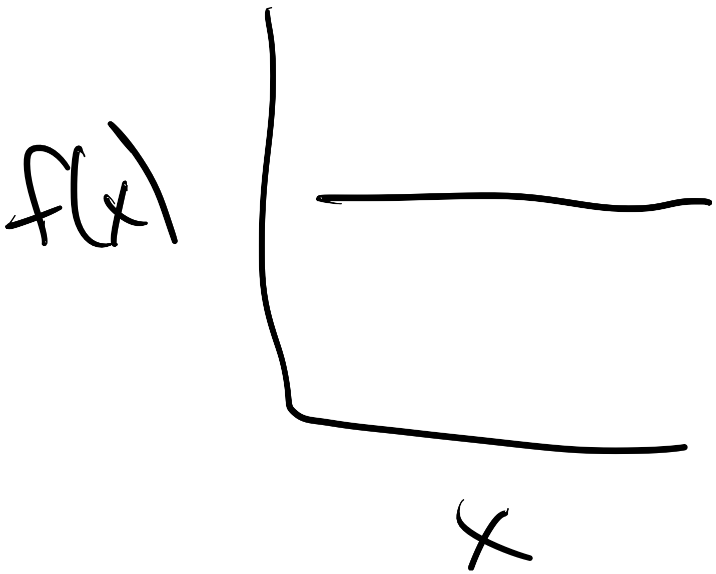


$\alpha \in (0,1)$

$f(\alpha x + (1-\alpha) y) \le \alpha f(x) + (1-\alpha) f(y)$

$x, y \in \Omega$

$\alpha x + (1-\alpha) y \in \Omega$



convex $\Rightarrow$
local minimizers are global minimizers

Ex: $\min \; c^T x$

s.t. $Ax \leq b$

can we compute them?

poly time algs:

- least squares (Thurs)
- LPs
- semi-definite programs

hard cases:

Example: $\Omega = \{ A \mid x^T A x \geq 0 \text{ if } x \geq 0 \}$

copositive matrices

hard to check membership