CS 331, Fall 2024
Lecture 18 (10/30)

Today:
- PSD matrices
- Convexity in $\mathbb{R}^d$
- Linear regression
- Algos for LP

PSD matrices (Part VI, Section 4.3)

Matrix factorization crash course:

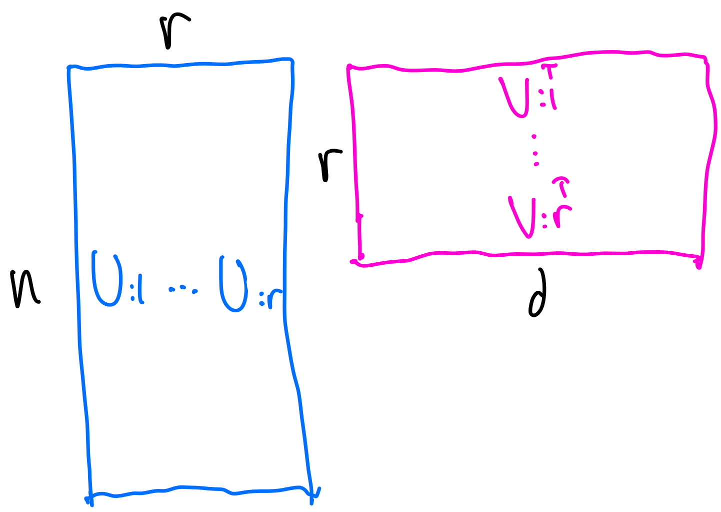
1) diag × anything

$$\underbrace{\Lambda}_{\text{diag}(\lambda)} V = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{bmatrix} \begin{bmatrix} V_{1:}^T \\ V_{2:}^T \\ \vdots \\ V_{d:}^T \end{bmatrix}$$

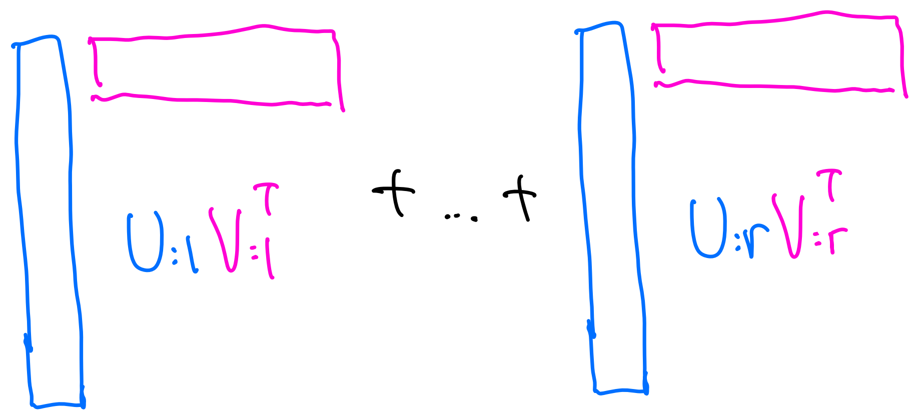
$$= \begin{bmatrix} \lambda_1 V_{1:}^T \\ \lambda_2 V_{2:}^T \\ \vdots \\ \lambda_d V_{d:}^T \end{bmatrix}$$

2) anything $\times$ (anything else)^T

$UV^T =$



$=$



Hence: SVD $\quad A = U \Sigma V^T, \quad \Sigma = \text{diag}(\sigma)$

nonnegative

$$A = \sum_{i \in [d]} \sigma_i u_i v_i^T$$

column of U column of V

Spectral theorem: $\forall$ symmetric $M \in \mathbb{R}^{d \times d}$

$$M = U \Lambda U^T \qquad U^T U = I$$

"orthonormal"

$$= \underbrace{\sum_{i \in [d]} \lambda_i u_i u_i^T}_{\text{eigendecomposition}}$$

$$U = \begin{pmatrix} u_1 & u_2 & \cdots & u_d \end{pmatrix}$$

$$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \\ & & & \lambda_d \end{pmatrix}$$

Compare to SVD:

$$M = U \Sigma V^T$$

$$= \sum_{i \in [d]} \sigma_i u_i v_i^T$$

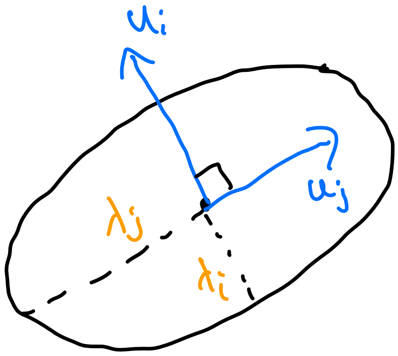
idea: if $\lambda_i < 0$

$$\Rightarrow \sigma_i = -\lambda_i$$
$$v_i = -u_i$$

If all $\lambda_i \geq 0$ already, SVD = eigendecomp.

We call such M PSD

"Every PSD matrix is an ellipse"



We say (u, λ) is $\underbrace{\text{eigvec/eigval}}_{\text{"eigpair"}}$ of M

if: $Mu = \lambda u$

Claim: if $M = U \Lambda U^T = \sum_{i \in [d]} \lambda_i u_i u_i^T$

then (λ_j, u_j) is eigpair $\forall j \in [d]$

Proof: $\left(\sum_{i \in [d]} \lambda_i u_i u_i^T \right) u_j = \lambda_j u_j$

$$\lambda_1 u_1 \underbrace{u_1^T u_j}_{=0} + \ldots + \lambda_j u_j \underbrace{u_j^T u_j}_{=1} + \ldots$$

Main claim: let M be symmetric, $d \times d$.

It's PSD iff $v^T M v \geq 0 \quad \forall v \in \mathbb{R}^d$

Proof: let $M = \sum_{i \in [d]} \lambda_i u_i u_i^T$

M not PSD $\Rightarrow \lambda_i < 0$ for some i

$\Rightarrow u_i^T M u_i = u_i^T (\lambda_i u_i) < 0$

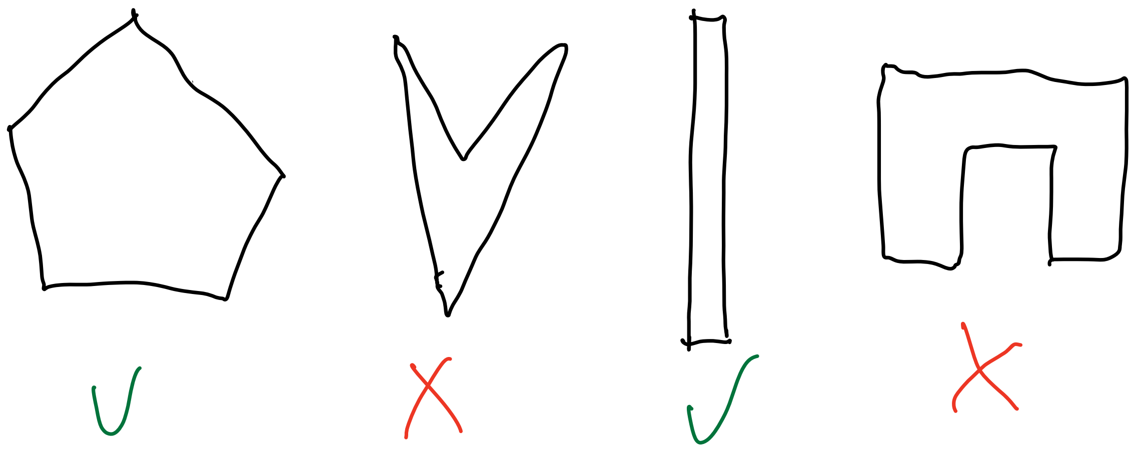
M is PSD $\Rightarrow v^T \left(\sum_{i \in [d]} \lambda_i u_i u_i^T \right) v$

$$= \sum_{i \in [d]} \underbrace{\lambda_i}_{\geq 0} \underbrace{(u_i^T v)^2}_{\geq 0} \geq 0.$$

e.g. M PSD means $M_{ii} \geq 0$: let $v = e_i$

Convexity in $\mathbb{R}^d$ (Part VI, Section 5.1)

Let $\mathcal{X} \subseteq \mathbb{R}^d$. We say $\mathcal{X}$ is <u>convex</u> if it contains all lines:



We say $f: \underset{\subseteq \mathbb{R}^d}{\mathcal{X}} \to \mathbb{R}$ is convex if:

- $\mathcal{X}$ convex
- All <u>1-d restrictions</u> of f are convex (stay at/below the line)

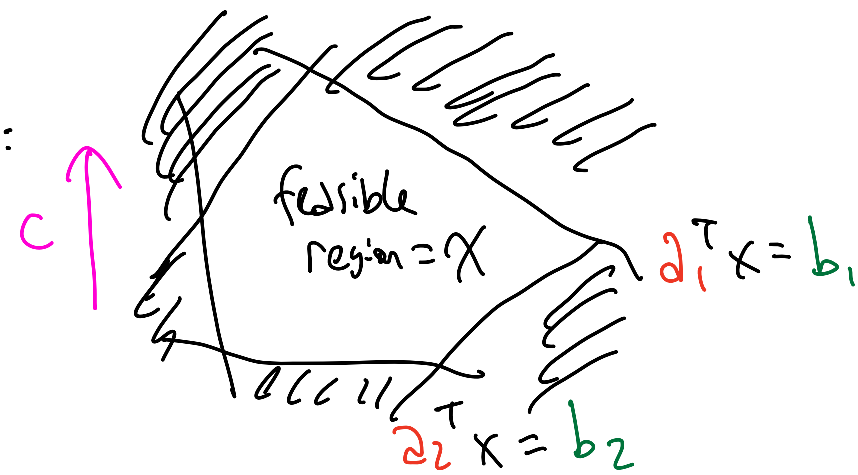
Two examples so far:

Section 3 • $\mathcal{X} = \mathbb{R}$ • $d = 1$

Section 2 • $\mathcal{X} = \{ x \in \mathbb{R}^d \mid Ax \le b \}$

• $f(x) = c^T x$ (LP)

$\mathcal{X}$ is convex:



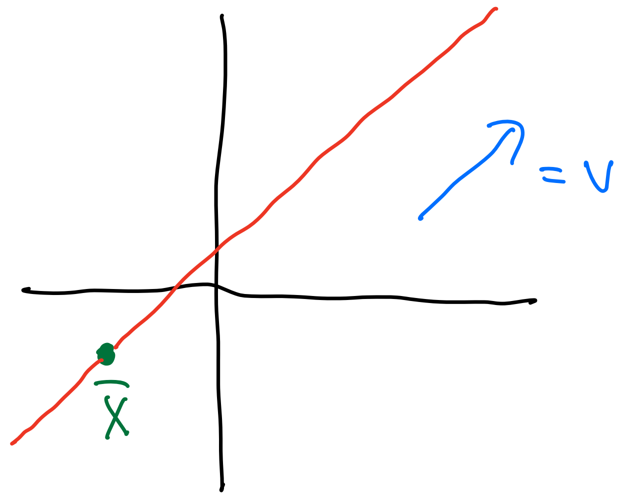
If $a_i^T x \le b_i \quad \forall i \in [d]$,

$a_i^T y \le b_i$

$$a_i^T \Big(\underbrace{(1-\lambda)x + \lambda y}_{\text{line between } x \,\&\, y} \Big) \le b_i \quad \forall \lambda \in [0,1]$$

Let $f : X \to \mathbb{R}$ be convex, so:

$f_{1d}(t)$
$= f(\bar{x} + t v)$
is convex.



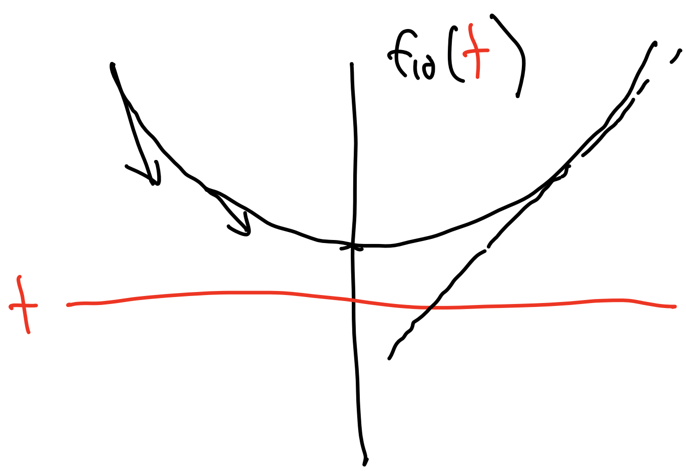
Fact 1:

$$f_{1d}(u) \geq f_{1d}(t) + f_{1d}'(t)(u - t)$$

"above the tangent line"

Fact 2:

$$f_{1d}''(t) \geq 0$$



So, what is $f_{\vec{v}}'(t)$?

$$\frac{\partial}{\partial t} f(\underbrace{\bar{x} + tv}_{x}) = \frac{\partial f}{\partial x_1}(x) \frac{\partial x_1}{\partial t} + \cdots$$

$$= \frac{\partial f}{\partial x_1}(x) v_1 + \ldots + \frac{\partial f}{\partial x_d}(x) v_d = \nabla f(x)^T v$$

Similarly, $f_{\vec{v}}''(t) = v^T \underbrace{\nabla^2 f(x)}_{d \times d \text{ matrix of } \frac{\partial^2 f}{\partial x_i \partial x_j}} v$

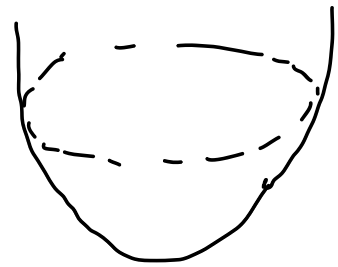
In $\mathbb{R}^d$: f is convex

Fact 1: $f(y) \geq f(x) + \underbrace{\nabla f(x)^T (y-x)}_{\text{the "tangent line"}}$

Fact 2: $v^T \nabla^2 f(x) v \geq 0 \quad \forall v \in \mathbb{R}^d$

Punchline: in $\mathbb{R}^d$, f is convex

$\iff \nabla^2 f(x)$ is PSD

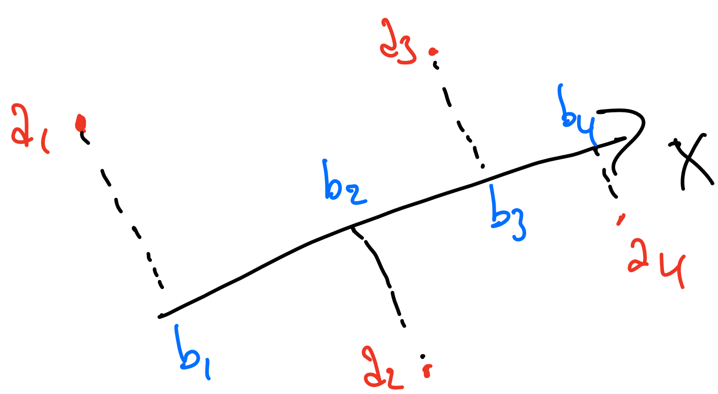


Linear regression (Part VI, Section 5.2)

Input: $\left\{ \left(\underset{\substack{\rotatebox{90}{$\in$} \\ \mathbb{R}^d}}{a_1}, \underset{\substack{\rotatebox{90}{$\in$} \\ \mathbb{R}}}{b_1} \right) \dots (a_n, b_n) \right\}$

"feature vector" "response"

Output: X that "explains" relationship



Goal: $\min_{x \in \mathbb{R}^d} \sum_{i \in [n]} \left(a_i^T x - b_i \right)^2$

model of relationship is <u>linear</u>

(HW: piecewise const.)

Another way to write: $A = \begin{pmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_n^T \end{pmatrix}$

$$f(x) = \| Ax - b \|_2^2$$

$$= (Ax - b)^T (Ax - b)$$

$$= x^T \underbrace{A^T A}_{M} x - 2 b^T A x + \text{const.}$$

Recall: M is PSD.

Fact: if $f(x) = x^T M x + V^T x$

then $\nabla^2 f(x) = 2M$

Regression is convex :)

For "nice" M we can solve fast using

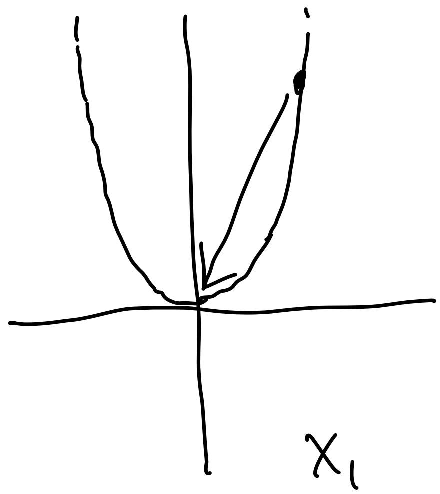
<u>gradient descent</u>

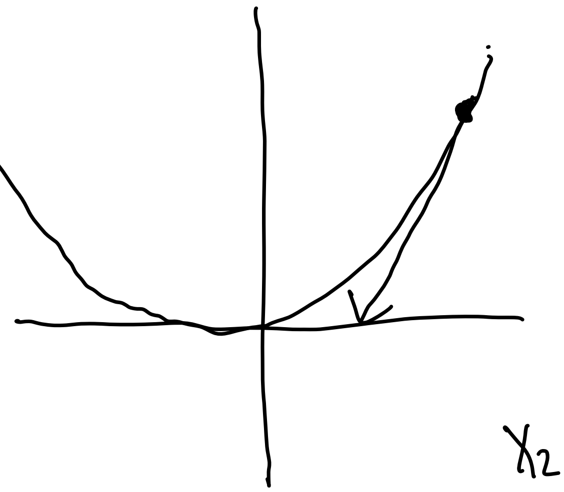
Sketch: Suppose $M = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$ (diagonal)

$$f(x) = \left(\lambda_1 x_1^2 + V_1 x_1\right) + \left(\lambda_2 x_2^2 + V_2 x_2\right)$$

only get to pick one step size...

$\eta = \frac{1}{2\lambda_1}$ perfect for x_1





perfect

undershot by factor $\frac{\lambda_2}{\lambda_1}$

Takes $\approx \frac{\lambda_1}{\lambda_2}$ iterations to converge.

General case: $M = U \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{pmatrix} U^T$

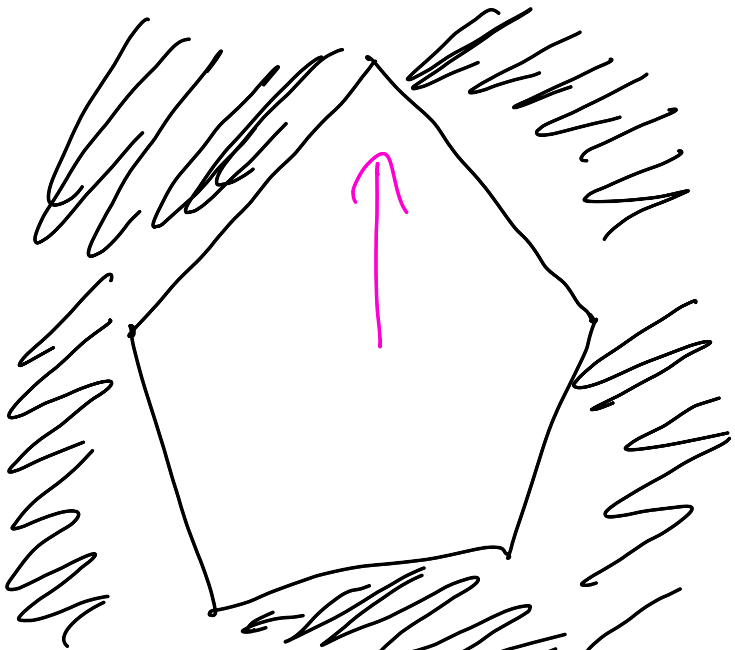
- GD converges in $\approx \frac{\lambda_1}{\lambda_d}$ iterations

  "how spherical is M"?

- No problem if $U \neq I$

  "input world = output world"

Algos for LP (Part VI, Section 5.3)



It's convex, but...

- constrained
- non-smooth

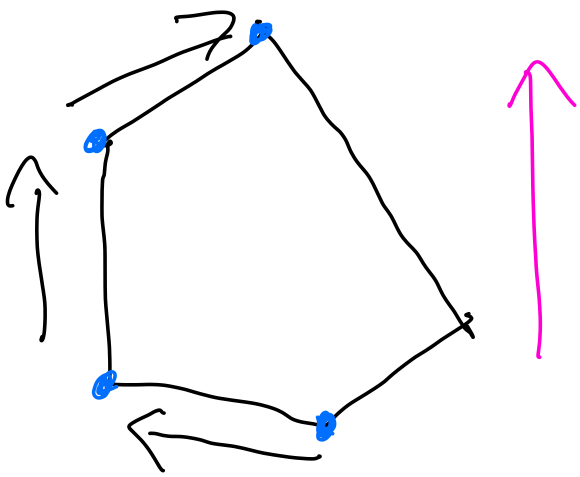
how to solve?

Strongly polynomial: open "Smale's 9th problem"

Weakly polynomial: ✓ great in practice!

- 10 most influential of 20th century
- NYT headline: "Breakthrough in Problem Solving"

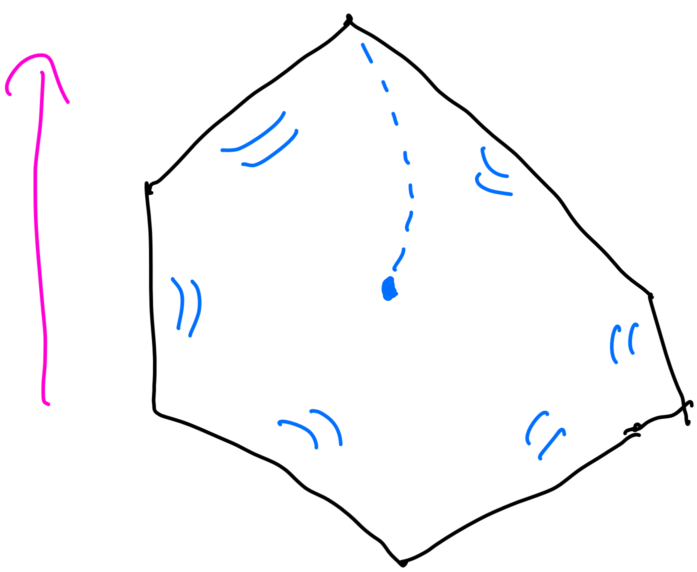
Idea 1: Simplex "walk along the boundary"



- decent in practice
- worst-case exp time
- "most instances" polytime

Idea 2: IPM "decrease the force fields"

$$x_t = \arg\max\ c^T x + t\,\underbrace{b(x)}_{\text{"barrier"}}$$



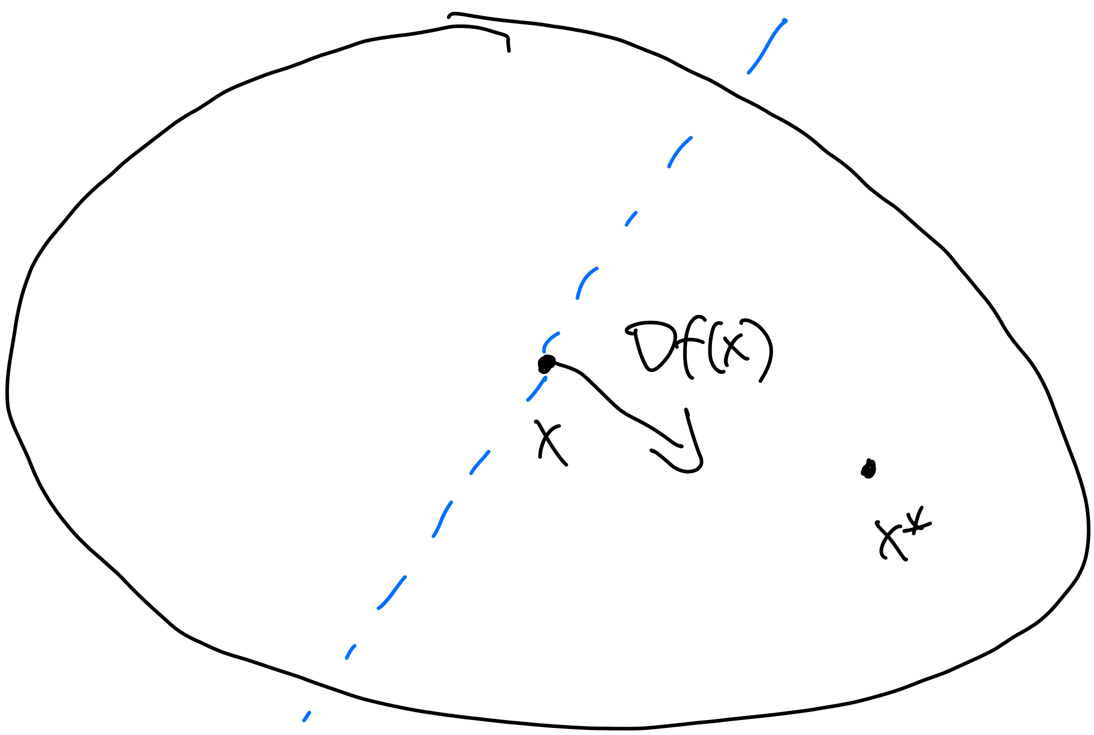
- polytime
- SOTA in theory
- SOTA (mostly) in practice

Idea 3: CPM "binary search in $\mathbb{R}^d$"

- works for all convex functions
- my favorite algo ☺

In 1-d: $f'(x)$ lets you binary search

In $\mathbb{R}^d$: use $\nabla f(x)$



$\forall$ planes through center-of-gravity x,

$\geq 30\%$ on both sides. Decrease volume!