Lecture 14: Linear programming and algorithms for compressive sensing.

Recall basis pursuit:

$$\min \|x\|_1 \quad \text{s.t.} \quad \underbrace{Ax = b.}$$

but how to deal with this?

linear system, can solve

We will develop a general paradigm for solving this via linear programming

A linear program is specified by:

1. Variables $x_1, \cdots, x_n \in \mathbb{R}$
2. Linear constraints, i.e. $i = 1, \cdots, m$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i$$

\# of constraints

or

$$\sum_{j=1}^{n} a_{ij} x_j = b_i$$

Note: this captures $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$ by negating: $\sum_{j=1}^{n} -a_{ij} x_j \geq -b_i$

technically, you don't even need equality:

$$\sum a_{ij} x_j = b_i \iff \sum a_{ij} x_j \leq b_i \text{ and } \sum -a_{ij} x_j \leq -b_i.$$

3. Objective function: again linear.

$$\min \sum_{j=1}^{n} c_j x_j.$$

Note: also captures max by negating as well.

What is not allowed: x_i^2, $\log x_i$, e^{x_i}, etc.

In general, we write this as

$$\min c^T x \quad \text{s.t.} \quad A x \geq b$$

this means coordinatewise.

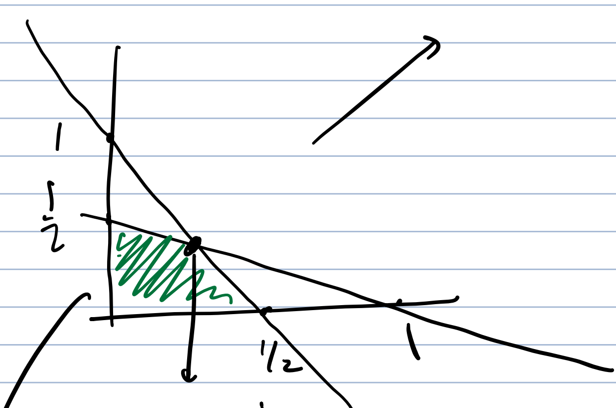
Example: max $x_1 + x_2$

s.t. $x_1 \geq 0$

$x_2 \geq 0$

$2x_1 + x_2 \leq 1$

$x_1 + 2x_2 \leq 1$



$(1/3, 1/3)$

a "polytope" in high dimensions.

<u>Example:</u> ℓ_1 minimization. $x \in \mathbb{R}^n$

$$\min \|x\|_1 \quad \text{s.t.} \quad \boxed{Ax = b} \text{ easy}$$

how to encode?

Idea: use additional variables. $y_1, \dots, y_n$

goal: $y_i = |x_i|$.

$$\min \|x_i\|_1 \Leftrightarrow \min \sum y_i .$$

Idea: enforce that $y_i \geq |x_i|$, so that minimizer will always take $y_i = |x_i|$.

Introduce $2n$ constraints $y_i \geq x_i, \; y_i \geq -x_i$.

$$\Leftrightarrow y_i \geq |x_i| .$$

$$\min \sum_{i=1}^{n} y_i \quad \text{s.t.} \quad y_i \geq x_i, \; y_i \geq -x_i \;\; \forall i$$
$$Ax = b .$$

$$Ax \geq b, \; -Ax \geq -b$$

Solving linear programs

LPs were introduced by Fourier in 1827

Kantorovich, Leontief 30s

Dantzig $\rightarrow$ Simplex Algorithm '47, but is worst case slow.

+ von Neumann $\rightarrow$ duality 48. min $\Leftrightarrow$ max

[Khachiyan '79]: A polynomial time algorithm for solving LPs called the ellipsoid method.

[Karmarkar '84]: Interior point methods (much faster)

hiding (many) $\log^c$ factors

Nowadays: $\tilde{O}(n^{2+1/18} L)$ [Jiang, Song, Weinstein, Zhang '20].

For sparse: $\tilde{O}(m\sqrt{n} L)$ [Lee-Sidford '15].

L = # bits needed to encode the input

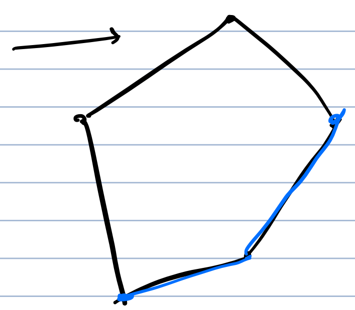
$\boxed{\text{Actual fastest: cvxpy}}$

Big open questions:

- Can L be removed? (strongly polynomial time)
- Can you get truly nearly-linear time?

(very) high level ideas:

simplex:



Fact: optimal solution is at a vertex of the polytope.

Fact: If you are at a suboptimal vertex, there is a neighboring vertex that improves your value.

worst case: exponential time.

However, for "most" instances, poly-time [Spielman-Teng '03].

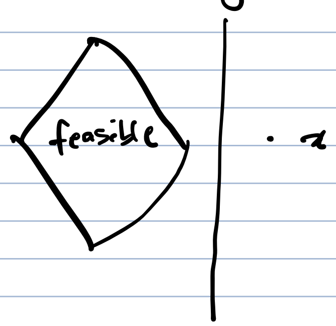
Smoothed analysis

Ellipsoid / cutting plane:

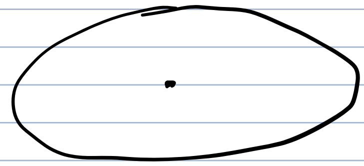
you can reduce optimization to feasibility:

does there exist a point that satisfies $Ax \geq b$?

Fact: for any point x which violates the constraints, one can easily find a separating hyperplane

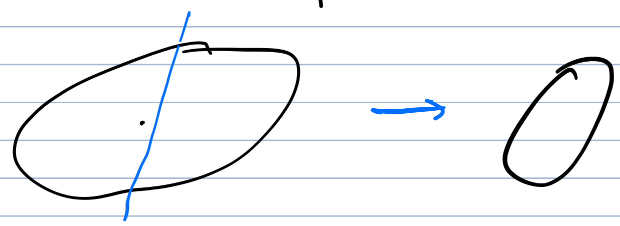


Idea: Maintain an ellipsoid of possible solutions



query the center of the ellipsoid.

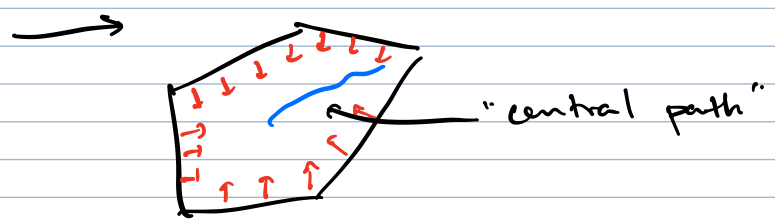
- if feasible, done
- if infeasible, use separating hyperplane to draw smaller ellipsoid



Can show that volume decreases, so process cannot go on forever.

Interior point

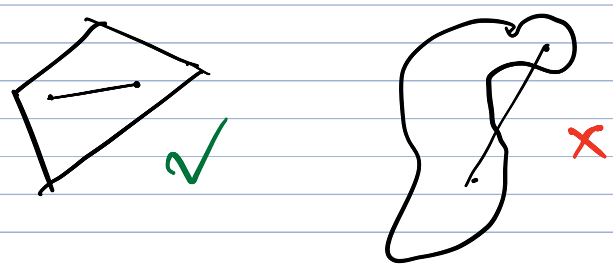
$\downarrow$: barrier function.



Convex programming

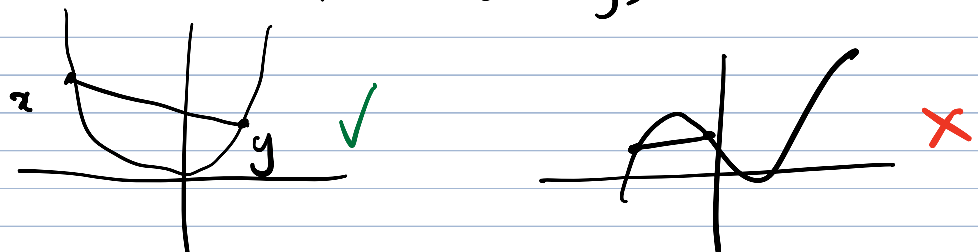
a further generalization of LPs.

<u>Def:</u> A set C is convex if $\forall x, y \in C$, $tx + (1-t)y \in C$, $\forall t$.



<u>Def:</u> A function $f: \mathbb{R}^n \to \mathbb{R}$ is convex if

$$f(tx + (1-t)y) \le t f(x) + (1-t) f(y), \ \forall x, y \in \mathbb{R}^n, \ t \in [0,1]$$



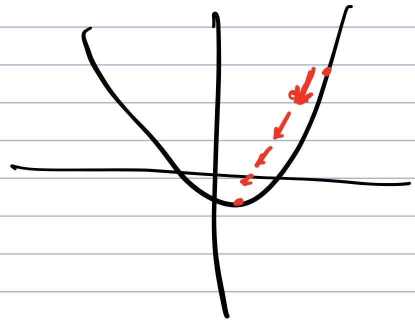
convex optimization:

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

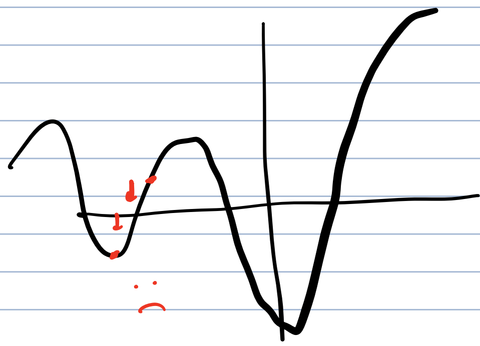
f is convex, C is convex.

given "good" access to f, C, this can be solved in poly-time.

convex function $\Longrightarrow$ local improvement implies global convergence.



gradient descent

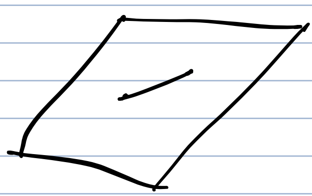


e.g. Any linear function is convex.

$$f(x) = a^T x.$$

$$f(tx + (1-t)x) = tf(x) + (1-t)f(x) \checkmark$$

e.g. Any polytope is convex.

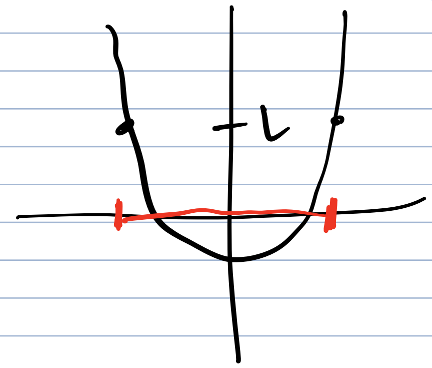


e.g. Any norm is convex (e.g. $\|\cdot\|_1$)
(b.c. of triangle inequality).

e.g. If f is a convex function, then $\forall L$,

$$G = \{ x : f(x) \le L \}$$ is convex

always minimization!



$\{ x : f(x) \geq L \}$ is **not** convex.

e.g. Let M_n be the set of $n \times n$ symmetric matrices.

Recall a symmetric matrix is positive semidefinite $\Leftrightarrow$ all eigenvalues are non-negative

linear constraints on M.

$$\Leftrightarrow x^T M x \geq 0 \quad \forall x \in \mathbb{R}^n.$$

Let $M_n^{\geq} : \{ M \in M_n : M \text{ is PSD} \}$

then $M_n^{\geq}$ is convex.

Semi-definite programming (SDP):

$$\min \langle C, X \rangle$$
$$\text{s.t.} \quad \langle A_i, X \rangle \geq b_i \quad , \forall i$$
$$X \quad \text{PSD}$$

can be solved efficiently! (since it's convex).

---

matrix completion.

given a low rank matrix that's missing data, can we find the missing entries?

Say unknown matrix is $M \in \mathbb{R}^{n \times m}$, and revealed entries are in set $S \subseteq [n] \times [m]$

previously: SVD.

not convex!

Now:

$$\min \ \mathrm{rank}(\hat{M})$$
$$\text{s.t.} \quad \hat{M}_{ij} = M_{ij}, \ \forall i,j \in S.$$

rank $\leftrightarrow$ matrices sparsity $\leftrightarrow$ vectors

Is thee a convex relaxation of rank?

Recall: SVD

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

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & 0 & \\ & & & & \ddots \end{pmatrix}$$

Σ is diagonal matrix of singular values.

M has rank r

$\iff$ M has r nonzero singular values.

so $\mathrm{rank}(M) = \|\Sigma\|_0$

natural idea: $\|\Sigma\|_1 \leftarrow$ is a valid norm on M

"nuclear norm"

$$\min \ \|\Sigma(\hat{M})\|_1$$
$$\text{s.t.} \quad \hat{M}_{ij} = M_{ij} \ \forall i,j \in S.$$

nuclear norm minimization.

uw!

Theorem [Candes-Recht, Candes-Tao, Keshevan-Montanari-Oh, Recht '09]

Suppose that the entries in S are chosen at random, that M is "spread out", and that

$$|S| \geq \tilde{\Omega}(r(m+n)).$$

Then NNM succeeds in recovering M.