Lecture 11

Last time

- Recap
- Initial point
- Optimality
- Pivoting

Today

- Complexity of Simplex
- Intro to interior point methods
- Remembering Newton.

Complexity of Simplex

With what we learned we can write a complete simplex method

Simplex method

- Start with a BFS $x(B_0)$ associated to B_0 (Use Phase 1)
- Loop for $k = 0, 1, \ldots$
  - Compute dual solution
    $$y(B_k) = A_{B_k}^{-T} c_{B_k}.$$
  - If $y(B_k)$ is feasible $(\bar{c} \geq 0)$ return $x(B_k)$ and $y(B_k)$.

- Else
  - Pick any j with $\bar{c}_j < 0$ and compute
    $$d = \begin{pmatrix} -A_{B_k}^{-1} A_j \\ 1 \\ 0 \end{pmatrix}.$$
  - If $d \geq 0$
    Return "unbounded LP."
  - Else set
    $B_{k+1} = B_k \cup \{j\} \setminus \{i\}$
    with $i \in \operatorname{argmin}\left\{ -\frac{x(B_k)_i}{d_i} \,\middle|\, d_i < 0 \right\}$.

Q: How do we know that the loop stops?

In order to guarantee that we need to use Bland's rule:

- Pick $j \in B^c$ with $\bar{c}_j < 0$ to be the smallest such index.
- Pick i the smallest index in $\operatorname{argmin}\left\{ -\frac{x(B_k)_i}{d_i} \,\middle|\, d_i < 0 \right\}$.

Using this rule, simplex always finishes. We will not prove that in this class.

Q. How fast is simplex?

- In the worst case it can take exponential time in the dimension (Klee and Minty '72).
- For random problems it takes $O(n+m)$ iterations on average (Borgwardt '87, Smale '83)
- For randomly perturbed problems, it finishes after poly(n, m) many iterations. (Smoothed Analysis, Spielman & Teng '04)

Intro to interior point methods

The Simplex has to fundamental

drawbacks:

1. It only applies to linear programming.
2. It might take exponential time.

Researchers in the '80s and '90s aimed to design a method to tackle 2 and unadvertidly found a method to tackle 1 as well.

History interlude

- In '79 Khatchiyan proved that the Elipsoid method converges in polynomial time for LP's.

This algorithm doesn't work well in practice (way worst than Simplex). BUT Khatchiyan's paper was extremly influential (even appearing in the New York times) and got many mathematicians interested in developing practical provably efficient methods for LPs.

- In '84 Karmarkar developed an Interior Point Method (IPM) for LPs that was theoretically on par with the Ellipsoid Method and had good practical performance (Also appeared on the New York Times).
- Later work by Renegar, Nesterov, and others led to IPM for other conic optimization problems (SOCP and SDP).
- It took some years and more practical insights, but

eventually IPM implementations were competitive with simplex and today these two are the basis of more LP commercial solvers.

Key insight

We describe the methods for LPs. Assume I wanted to solve

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \le b. \end{aligned} \qquad (☺)$$

Instead of directly solving this problem, we could consider an unconstrained problem

$$\min \; \eta c^T x + B(x) \qquad (L_\eta)$$

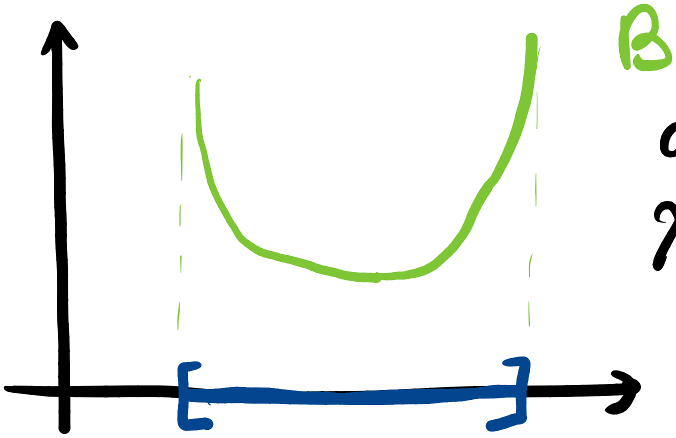
where B satisfies

1. $\operatorname{dom}(B) = \operatorname{int} \overbrace{\{x \mid Ax \le b\}}^{P}$
2. $\forall q \in \operatorname{bd} B$ we have

$$\lim_{x \to q} B(x) = \infty.$$

3. B is strictly convex

Intuition

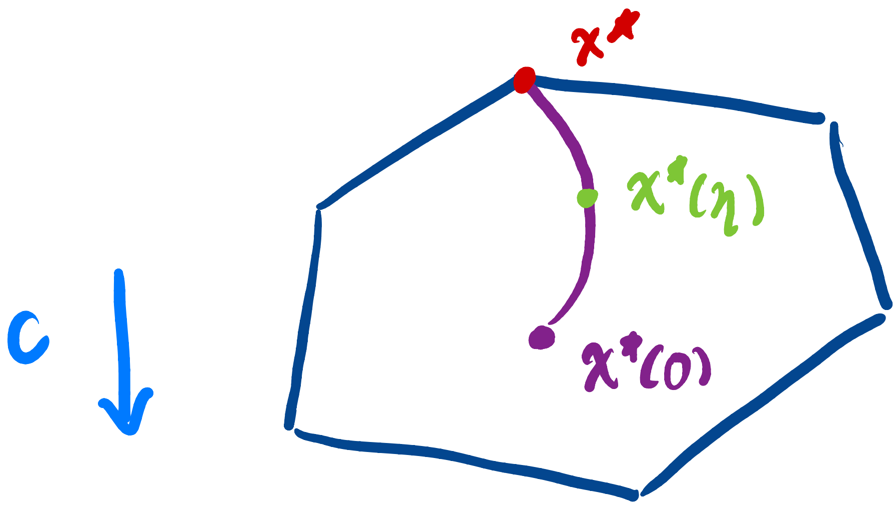


By 3, (L_η) has a unique minimizer x^*_η that lies in int P

As $\eta \uparrow \infty$ that minimizer approaches a solution x^* to (☺).

This defines a path of interior solutions

called the central path



If we started at $x^*(0)$, we could aim to find numerically approximate $x^*(\eta)$ for small η using an iterative method initialized at $x^*(0)$.

IPM (informal)

- Pick x_0 sufficiently close to $x^*(0)$ and pick $\eta_0 > 0$ small.
- Loop $k = 0, 1, \ldots, T$:
  - Find an approximate solution x_{k+1} to (L_{η_k}) using x_k for initialization.
  - Increase $\eta_{k+1} = q\,\eta_k$ for $q > 1$.
- Return x_T

Once more, we arrive to a lot of questions:

Q0: What B function to use?

Q1: How to find x_0? What

is sufficiently close?

Q2: What method to use to find x_{k+1}? How well do we need to approximate $x^*(\eta_k)$?

Q3: How to show that the method finds an approximate solution in polynomial time?

Remembering Newton

In Nonlinear 1 we covered Newton's method, which given a problem

$f \in C^2$

$$\min_{x \in \mathbb{R}^d} f(x)$$

updates

$$x_{k+1} \leftarrow x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k).$$

This method turned up

to be extremly fast near a minimizer. Let $x^* \in \arg\min f$.

Theorem (*): Suppose that f is such that $\forall x$ near x^*

$$\alpha I \preceq \nabla f^2(x) \preceq \beta I$$

α-strongly convex β-smooth

Then, for any point x_0 sufficiently close to $x^* = \arg\min f$, we have

$$\|x_1 - x^*\|_2 \le \frac{\beta}{2\alpha} \|x_0 - x^*\|_2^2 .$$

⊣

In particular when

$$\|x_0 - x^*\| < \frac{\alpha}{4\beta},$$

we have quadratic convergence. This seems like a natural candidate for answering Q2. But note that the number of steps to achieve good accuracy depends on the constants α and β, which depend on B,

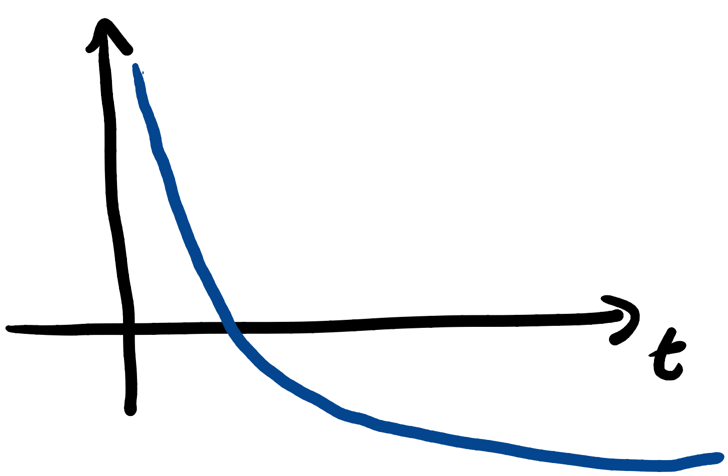
since

$$\nabla^2 f_n(x) = D^2 B(x).$$

Let's answer Q_0 to see how does this Hessian look like. For LPs it is reasonable (and in fact a good idea) to use:

$$B(x) = -\sum_{i=0}^{m} \log(b_i - a_i^T x).$$

Recall $-\log(t)$ looks like



So B satisfies the properties we want and

$$\nabla B(x) = -\sum \frac{a_i}{(b_i - a_i^T x)}$$

$$\nabla^2 B(x) = +\sum \frac{a_i a_i^T}{(b_i - a_i^T x)^2}$$

Then, our convergence rate will depend on $\lambda_{min}(\nabla^2 B(x^*(\eta_k)))$ and $\lambda_{max}(\nabla^2 B(x^*(\eta_k)))$ and so the convergence will depend on both

a) The conditioning of $A^T A$

b) How close we are to the boundary of P.

Both are terrible because we can have a badly conditioned representation of the problem (A, b, c) that leads to arbitrarily slow convergence. Further, we don't want our complexity blowing up as we approach the boundary of B.

We cannot use Theorem (∗).

The solution will turn out to be the affine invariance of Newton's method.