# Solving SDPs (quicker)

1. Iterative algorithms for solving SDPs
2. Speed up? Solving smaller problems $\to$ JL and dimension reduction
3. Sketching linear systems
4. Sketching SDPs.

[Laurent - Valentin, Chapter 4]

Conic program $\left[\begin{array}{c} \min c^T x \\ x \in K \\ Ax = b \end{array}\right]$ We will talk about general ideas for solving it and discuss the connection with $K$ being "nice"

Review: Newton's method

To start, assume that $f$ is so that its Hessian is PSD everywhere ($\nabla^2 f(u) \succ 0$).

As correctly noted in class, strict convexity alone does not imply $\nabla^2 f \succ 0$! (consider $x^4$)

$$q(x) = f(u) + \nabla f(u)^T x + \frac{1}{2} x^T \cdot \underbrace{\nabla^2 f(u)}_{\text{PSD}} \cdot x$$

$$0 = \nabla q(x_*) = \nabla f(u) + \nabla^2 f(u) x_*$$

$$x_* = -(\nabla^2 f(u))^{-1} \cdot \nabla f(u)$$

So, initializing $u \approx x_*$ and solve quadratic approximation

Iteration: $\boxed{\begin{array}{l} \tilde{x} = -(\nabla^2 f(x_k))^{-1} \cdot \nabla f(x_k) \\ x_{k+1} = x_k + \tilde{x} \end{array}}$

If $\begin{cases} \text{strictly convex} \\ x_0 \approx x_* \end{cases}$, converges quadratically

If convex but not strictly convex, example:

$f(z) = \frac{1}{4} z^4 - z$ argmin $x_* = 1$

$f'(z) = z^3 - 1$

$f''(z) = 3z^2$

But: $x_0 = 0$ means division by zero

$x_0 = \frac{1}{\sqrt[3]{2}} \to x_1 = 0$ (check!) and $\infty$ again

See wikipedia page "Newton fractal" for further discussion of this example and beautiful pictures :)

Other problem: $x_0 + \tilde{x} \notin \Omega$ ($\to$ take step size ensuring $x_k \in \Omega$)

More intelligent way for equality constraints

Consider Lagrangian

$$\mathcal{L}(x, \lambda_1, \dots \lambda_m) = f(x) + \sum \lambda_i a_i^T x$$

$$\nabla \mathcal{L}(y^*) = 0$$

$$f(x) \approx f(u) + \nabla f^T(u) x + \frac{1}{2} x^T \nabla^2 f(u) x$$

$$\to \begin{cases} \nabla f(u) + \nabla^2 f(u) x_* + \sum \lambda_i a_i = 0 \\ a_i^T (u + x_*) = b_i \quad i = 1 \dots m \end{cases}$$

Solve systems to find $x_*$ (next direction)

Now, we have additional constraint: $x \in K$

Idea: modify a function adding a barrier function: $\begin{bmatrix} \inf c^T x + \varphi(x) \\ a_i^T x = b_i \end{bmatrix}$ $\varphi(x) = \begin{cases} 0 & x \in K \\ \infty & \text{otherwise} \end{cases}$

Convexity?

Create more reasonable (strictly convex) functions:
$x \to \partial K$
$\varphi(x) \to \infty$

<u>Examples</u> $\cdot K = \mathbb{R}^n_{\geq 0}$ $\varphi(x) = -\ln(x_1 \dots x_n)$

$\cdot K = PSD(n)$ $\varphi(X) = -\ln \det X$ $\qquad \nabla(\ln \det X) = X^{-1}$

$\nabla \varphi(X) = -X^{-1}$

Solve $\min \begin{bmatrix} t(c^T x) + \varphi(x) \\ a_i^T x = b_i \end{bmatrix} \longrightarrow x(t)$ optimal solution — central path

- $t \to \infty$ $\quad x(t) \to x_*$
- large $t$ implies the point became far from optimum and Newton is slow
- ① compute $x(t)$ starting from $x_0$ (Newton)

② $x_0 := x(t)$
$t := \mu \cdot t$ (increase)

③ repeat until some stopping criterion

For SDPs:

<u>How to find a starting point?</u>

Phase 1: $\begin{bmatrix} \min\limits_{X,\lambda} \lambda \\ X + \lambda I \succeq 0 \\ \langle A_i, X \rangle = b_i \end{bmatrix} \longrightarrow$ find strictly feasible solution — apply interior method
— if we find $(X, \lambda)$ with $\lambda < 0$, we find feasible solution for original problem, otherwise it does not exist

<u>Central path $x(t)$</u>

(P) $\max \begin{bmatrix} \mathrm{Tr}(C^T X) \\ \mathrm{Tr}(A_i X) = b_i \\ X \succeq 0 \end{bmatrix} \longleftrightarrow$ (D) $\min \begin{bmatrix} \sum \lambda_i b_i \\ \sum \lambda_i A_i - C \succeq 0 \end{bmatrix}$ $\qquad \sum \lambda_i A_i - C =: Y$

Can be restated in the following way:
With $X_0$: $\mathrm{Tr}(A_i X_0) = b_i \; \forall i$ and $L := \{X \succeq 0. \; \mathrm{Tr}(A_i X) = b_i\}$

$$\max \begin{bmatrix} \mathrm{Tr}(C^T X) \\ X \succeq 0 \\ X \in X_0 + L \end{bmatrix} \longleftrightarrow \mathrm{Tr}(X_0 C) + \min \begin{bmatrix} \mathrm{Tr}(X_0 Y) \\ Y \succeq 0 \\ Y \in -C + L^\perp \end{bmatrix}$$

$$\sum \lambda_i A_i - C =: Y$$

$$\sum \lambda_i \mathrm{Tr}(A_i X_0) - \mathrm{Tr}(X_0 C) = \mathrm{Tr}(X_0 Y)$$

Consider path functions:

$P_t(X) := -t\,\mathrm{Tr}(CX) - \ln\det X \longrightarrow$ maximal over $X_0 + L$ at a point we call $X(t)$

$D_t(Y) := t\,\mathrm{Tr}(X_0 Y) - \ln\det Y \longrightarrow$ minimal over $-C+L^\perp$ at a point we call $Y(t)$

$(X(t), Y(t))$ is called primal-dual central path.

Then (a) $X(t)\cdot Y(t) = \frac{1}{t} I$

(b) $\underbrace{\mathrm{Tr}(X_0 Y(t)) + \mathrm{Tr}(C X_0)} - \mathrm{Tr}(X(t) C) = \frac{n}{t}$

Corollary: linear convergence in $t$ as $t \to \infty$, both $\to 0$

Why?

(a) $\Rightarrow$ (b) Let $\sum \lambda_i b_i = \overbrace{\mathrm{Tr}(CX_0) + \mathrm{Tr}(X_0 \cdot Y(t))}^{\lambda_i \text{ are so that}}$

$$\sum \lambda_i b_i - \mathrm{Tr}(CX(t)) = \sum \lambda_i \mathrm{Tr}(A_i, X(t)) - \mathrm{Tr}(C, X(t))$$

$$= \mathrm{Tr}(Y(t) X(t)) = n \cdot \mathrm{Tr}(\tfrac{1}{t} I)$$

(a) By the assumption about optimality of $X(t)$: $\quad -tC - X(t)^{-1} \in L^\perp$

$Y(t)$: $\quad tX_0 - Y^{-1}(t) \in L$

(Derivative $= 0$ on the feasible space)

Then,

$\frac{1}{t} X(t)^{-1} \in C + L^\perp$ – feasible for dual

$\frac{1}{t} Y^{-1}(t) \in X_0 + L$ – feasible for primal

So, $X(t) \succeq \frac{1}{t} Y^{-1}(t)$ optimal and feasible for primal

$Y(t) \preceq \frac{1}{t} X^{-1}(t)$ optimal and feasible for dual

$\frac{1}{t} I \preceq X(t) Y(t) \preceq \frac{1}{t} I$ ∎

---

This gives polynomial algorithms (but still not fast...)

Idea: reduce the size the problem.