Chapter 5: Ellipsoid Method and Interior Point Method

June 20, 2022

Scribe: Eric Chen

1 Ellipsoid Method

We can think of Ellipsoid Method as bisection method in high dimension. The goal of this algorithm is to decide whether a given polyhedron

$$P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$$

is nonempty.

Definition 1.1 (Separation Oracle). Given $x \in \mathbb{R}^n$ and a convex set K, the Separation Oracle SEP_K either: (1) asserts that $x \in K$ or (2) returns a separating hyperplane between K and x such that

$$h^T z \leq h^T x, \quad \forall z \in K.$$

Definition 1.2 (Löwner-John Ellipsoid). The **Löwner-John Ellipsoid** is the unique ellipsoid E of minimum volumn satisfying $K \subseteq E$. Moreover, if we shrink E by a factor $1/n$ around its center to get E_{new}, then we have $E_{new} \subseteq K$.

Theorem 1.3. *Let $E = E(z, D)$ be an ellipsoid in $\mathbb{R}^n$ and let nonzero vector $a \in \mathbb{R}^n$. Consider the halfspace $H = \{x \in \mathbb{R}^n \mid a^T x \geq a^T z\}$ and let*

$$\bar{z} = z + \frac{1}{n+1}\frac{Da}{\sqrt{a^T Da}},$$
$$\bar{D} = \frac{n^2}{n^2-1}\left(D - \frac{2}{n+1}\frac{Daa^T D}{a^T Da}\right).$$

So we get a new ellipsoid $E' = E(\bar{z}, \bar{D})$. Moreover, we have

(a) $E \cap H \subset E'$,

(b) $Vol(E') < e^{\frac{1}{2(n+1)}} Vol(E)$.

Then the ellipsoid method for determining whether a given polyhedron is empty.

Algorithm 1 The Ellipsoid Method

Require:

(a) $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ that define $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$.

(b) A ball $E_0 = E(x_0, r^2 I)$ such that $P \subset E_0$.

for $t = 0, 1, \ldots$ **do**
  Run $SEP_P(x_t)$
  if $h \neq 0$ **then** we have $H = E_t \cap \{x \mid h^T x \leq h^T x_t\}$, $E_{t+1} = E(x_{t+1}, Q_{t+1})$.
  else $x_t \in P$, **return** x_t.
  end if
end for

2 Primal-Dual Methods

Consider the problem

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

we have its dual problem

$$\begin{aligned} \min \quad & b^T \lambda \\ \text{s.t.} \quad & A^T \lambda + s = c \\ & s \geq 0 \end{aligned}$$

To obtain optimal solution, we write down the KKT conditions:

$$\begin{aligned} A^T \lambda + s &= c \\ Ax &= b \\ x_i s_i &= 0, \quad i = 1, 2, \ldots, n \\ (x, s) &\geq 0. \end{aligned}$$

To derive primal-dual interior-point methods we restate the optimality conditions in a slightly different form:

$$F(x, \lambda, s) = \begin{bmatrix} A^T \lambda + s - c \\ Ax - b \\ XSe \end{bmatrix} = 0$$
$$(x, s) \geq 0$$

where

$$X = \text{diag}(x_1, x_2, \ldots, x_n), \quad S = \text{diag}(s_1, s_2, \ldots, s_n).$$

To find solutions, we obtain the search direction $(\Delta x, \Delta\lambda, \Delta s)$ by solving the following linear system:

$$J(x, \lambda, s) \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = -F(x, \lambda, s)$$

where J is the Jacobian of F, we denote

$$r_b = Ax - b, \quad r_c = A^T\lambda + s - c,$$

then we can write the Newton equations as follows:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}.$$

As this step would usually violate the bound $(x, s) \geq 0$, so we perform a line search along the Newton direction and define the new iterate as

$$(x, \lambda, s) + \alpha(\Delta x, \Delta\lambda, \Delta s),$$

for some line search parameter $\alpha \in (0, 1]$, and this is known as the **affine scaling direction**. The restriction of this direction is that we can make limited progress towards the optimal solution. Another alternative method is that we take a Newton step toward the point for which $x_i s_i = \sigma\mu$, where

$$\mu = \frac{1}{n} x^T s$$

is known as *duality measure*, and $\sigma \in [0, 1]$ is known as the *centering parameter*, the modified step is then

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe + \sigma\mu e \end{bmatrix}.$$

An intuition behind this is as $\sigma > 0$, it is usually possible to take a longer step α along the direction before violating the bounds $(x, s) \geq 0$. Below is the general framework of primal-dual path-following algorithm.

Algorithm 2 Framework of Primal-Dual Path-Following

Require: (x^0, λ^0, s^0) with $(x^0, s^0) > 0$;
  for $k = 0, 1, 2, \ldots$ **do**
    Choose $\sigma_k \in [0, 1]$ and solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -r_c^k \\ -r_b^k \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix},$$

    where $\mu_k = (x^k)^T s^k / n$;
    Set

$$(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha_k (\Delta x^k, \Delta \lambda^k, \Delta s^k),$$

    choosing α_k such that $(x^{k+1}, s^{k+1}) > 0$.
  end for

Note that the choices of centering parameter σ_k and step length α_k are crucial to the performance of the method.

2.1 Central Path

Definition 2.1. We define the primal-dual *feasible set* $\mathcal{F}$ and *strictly feasible set* $\mathcal{F}^o$ as follows:

$$\mathcal{F} = \left\{(x, \lambda, s) \mid Ax = b, A^T\lambda + s = c, (x, s) \geq 0\right\},$$
$$\mathcal{F}^o = \left\{(x, \lambda, s) \mid Ax = b, A^T\lambda + s = c, (x, s) > 0\right\}.$$

Definition 2.2. We define *central path* as

$$\mathcal{C} = \{(x_\tau, \lambda_\tau, s_\tau) \mid \tau > 0\}.$$

We say that point $(x_\tau, \lambda_\tau, s_\tau) \in \mathcal{C}$ if it satisfies:

$$\begin{aligned} A^T\lambda + s &= c, \\ Ax &= b, \\ x_i s_i &= \tau, \quad i = 1, 2, \ldots, n \\ (x, s) &> 0. \end{aligned}$$

Remark. *It can be shown that $(x_\tau, \lambda_\tau, s_\tau)$ is defined uniquely for each $\tau > 0$ if and only if $\mathcal{F}^o$ is nonempty.*

2.2 Central Path Neighborhoods and Path-Following Methods

Path-following algorithms explicitly restrict the iterates to a neighborhood of the central path $\mathcal{C}$ and follow $\mathcal{C}$ to a solution of the linear program. The two most interesting neighborhoods of $\mathcal{C}$ are

$$\mathcal{N}_2(\theta) = \{(x, \lambda, s) \in \mathcal{F}^o \mid \|XSe - \mu e\|_2 \leq \theta\mu\},$$

for some $\theta \in [0, 1)$, and

$$\mathcal{N}_{-\infty}(\gamma) = \{(x, \lambda, s) \in \mathcal{F}^o \mid x_i s_i \geq \gamma\mu \quad \text{all } i = 1, 2, \ldots, n\},$$

for some $\gamma \in (0, 1]$. We can actually notice that $\mathcal{N}_2(\theta)$ is more restrictive while $\mathcal{N}_{-\infty}(\gamma)$ is quite modest.

By using the above neighborhoods, we specify the *long-step path-following* algorithm. This algorithm can make rapid progress because of use of the wide neighborhood $\mathcal{N}_{-\infty}(\gamma)$ for γ close to zero, and we choose the step length α_k to be as large as possible.

In the algorithm and analysis, we use the notation:

$$\left(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)\right) \stackrel{\text{def}}{=} \left(x^k, \lambda^k, s^k\right) + \alpha\left(\Delta x^k, \Delta\lambda^k, \Delta s^k\right),$$
$$\mu_k(\alpha) \stackrel{\text{def}}{=} x^k(\alpha)^T s^k(\alpha)/n.$$

Algorithm 3 Long-Step Path-Following

Require: $\gamma, \sigma_{\min}, \sigma_{\max}$ with $\gamma \in (0, 1), 0 < \sigma_{\min} \leq \sigma_{\max} < 1$, and $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$;

for $k = 0, 1, 2, \ldots$ **do**

Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$;

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta\lambda^k \\ \Delta s^k \end{bmatrix} = \begin{bmatrix} -r_c^k \\ -r_b^k \\ -X^k S^k e + \sigma_k \mu_k e \end{bmatrix},$$

to obtain $(\Delta x^k, \Delta\lambda^k, \Delta s^k)$;

Choose largest α_k such that

$$\left(x^k(\alpha), \lambda^k(\alpha), s^k(\alpha)\right) \in \mathcal{N}_{-\infty}(\gamma);$$

Set

$$\left(x^{k+1}, \lambda^{k+1}, s^{k+1}\right) = \left(x^k\left(\alpha_k\right), \lambda^k\left(\alpha_k\right), s^k\left(\alpha_k\right)\right);$$

end for

2.3 Long-Step Path-Following Analysis

The main goal of this analysis is to show that given some small tolerance $\epsilon > 0$, the algorithm requires $O(n|\log\epsilon|)$ iterations to reduce the duality measure by a factor of ϵ, that is to identify a point (x^k, λ^k, s^k) for which $\mu_k \leq \epsilon\mu_0$.

Lemma 2.3. *Let u and v be any two vectors in $\mathbb{R}^n$ with $u^T v \geq 0$. Then*

$$\|UVe\|_2 \leq 2^{-3/2}\|u + v\|_2^2,$$

where

$$U = \mathit{diag}(u_1, u_2, \ldots, u_n), \quad V = \mathit{diag}(v_1, v_2, \ldots, v_n).$$

Lemma 2.4. *If* $(x, \lambda, s) \in \mathcal{N}_{-\infty}(\gamma)$*, then*

$$\|\Delta X \Delta S e\| \leq 2^{-3/2}(1+\gamma)n\mu.$$

Theorem 2.5. *Given the parameters* $\gamma, \sigma_{\min}, \sigma_{\max}$*, then there is a constant* δ *independent of* n *such that*

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{n}\right)\mu_k,$$

for all $k \geq 0$*.*

Theorem 2.6. *Given* $\sigma \in (0,1)$ *and* $\gamma \in (0,1)$*, suppose the starting point* $(x^0, \lambda^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$*. Then there is an index* K *with* $K = O(n \log 1/\epsilon)$ *such that*

$$\mu_k \leq \epsilon \mu_0, \quad \forall k \geq K.$$

Proof. As

$$\mu_{k+1} \leq \left(1 - \frac{\delta}{n}\right)\mu_k,$$

we obtain

$$\log \mu_{k+1} \leq \log\left(1 - \frac{\delta}{n}\right) + \log \mu_k.$$

By repeating this step, we have

$$\log \mu_k \leq k \log\left(1 - \frac{\delta}{n}\right) + \log \mu_0.$$

As $\log(1+x) \leq x$, we have

$$\log(\mu_k/\mu_0) \leq k\left(-\frac{\delta}{n}\right).$$

Therefore, the condition $\mu_k/\mu_0 \leq \epsilon$ is satisfied if we have

$$k\left(-\frac{\delta}{n}\right) \leq \log \epsilon.$$

This inequality holds for all k that satisfy

$$k \geq K \stackrel{\text{def}}{=} \frac{n}{\delta} \log \frac{1}{\epsilon} = \frac{n}{\delta} |\log \epsilon|.$$

□

3 Practical Primal-Dual Algorithms

3.1 Corrector and Centering Steps

A key feature of practical algorithms is their use of corrector steps that compensate for the linearization error made by the Newton (affine-scaling) step in modeling the equation $x_i s_i = 0, i = 1, 2, \ldots, n$. Consider the affine-scaling direction $(\Delta x, \Delta\lambda, \Delta s)$ defined by

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{aff}} \\ \Delta\lambda^{\text{aff}} \\ \Delta s^{\text{aff}} \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix},$$

if we take a full step in this direction, we obtain

$$\begin{aligned} &\left(x_i + \Delta x_i^{\text{aff}}\right)\left(s_i + \Delta s_i^{\text{aff}}\right) \\ &= x_i s_i + x_i \Delta s_i^{\text{aff}} + s_i \Delta x_i^{\text{aff}} + \Delta x_i^{\text{aff}} \Delta s_i^{\text{aff}} = \Delta x_i^{\text{aff}} \Delta s_i^{\text{aff}}. \end{aligned}$$

We can see that the updated value $x_i s_i$ is $\Delta x_i^{\text{aff}} \Delta s_i^{\text{aff}}$ rather than the ideal value of 0. So we can solve the following system in order to obtain a step $(\Delta x^{\text{cor}}, \Delta\lambda^{\text{cor}}, \Delta s^{\text{cor}})$ that attempts to correct for this deviation from ideal:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{cor}} \\ \Delta\lambda^{\text{cor}} \\ \Delta s^{\text{cor}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Delta X^{\text{aff}} \Delta S^{\text{aff}} e \end{bmatrix}.$$

Generally speaking, the combined step $\left(\Delta x^{\text{aff}}, \Delta\lambda^{\text{aff}}, \Delta s^{\text{fff}}\right) + \left(\Delta x^{\text{cor}}, \Delta\lambda^{\text{cor}}, \Delta s^{\text{cor}}\right)$ does a better job of reducing the duality measure than does the affine-scaling step alone.

The affine-scaling step can be used as the basis of choosing σ_k. Roughly speaking, if the affine-scaling step reduces the duality measure significantly, there is not much for centering, so a smaller value of σ_k is appropriate. Conversely, if not much progress can be made along this direction before reaching the boundary of the nonnegative orthant, a larger value of σ_k will ensure that the next iterate is more centered, so a longer step will be possible from this next point. Specially, we define the affine-scaling direction as follows:

$$\begin{aligned} \alpha_{\text{aff}}^{\text{pri def}} &= \min\left(1, \min_{i:\Delta x_i^{\text{aff}}<0} -\frac{x_i}{\Delta x_i^{\text{aff}}}\right), \\ \alpha_{\text{aff}}^{\text{dual}} &\stackrel{\text{def}}{=} \min\left(1, \min_{i:\Delta s_i^{\text{aff}}<0} -\frac{s_i}{\Delta s_i^{\text{aff}}}\right), \end{aligned}$$

and then we define μ_{aff} as:

$$\mu_{\text{aff}} = \left(x + \alpha_{\text{aff}}^{\text{pri}} \Delta x^{\text{aff}}\right)^T \left(s + \alpha_{\text{aff}}^{\text{dual}} \Delta s^{\text{aff}}\right)/n.$$

Then the centering parameter is chosen as:

$$\sigma = \left(\frac{\mu_{\text{aff}}}{\mu}\right)^3.$$

Then we can obtain the search direction by solving:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe - \Delta X^{\text{aff}} \Delta S^{\text{aff}} e + \sigma \mu e \end{bmatrix}.$$

3.2 Step Lengths

We define $\alpha^{\text{pri}}_{k,\max}$ and $\alpha^{\text{pri}}_{k,\max}$ as follows:

$$\alpha^{\text{pri}}_{k,\max} \stackrel{\text{def}}{=} \min_{i:\Delta x^k_i<0} -\frac{x^k_i}{\Delta x^k_i}, \quad \alpha^{\text{dual}}_{k,\max} \stackrel{\text{def}}{=} \min_{i:\Delta s^k_i<0} -\frac{s^k_i}{\Delta s^k_i},$$

we can observe that they are the largest values of α for which $x^k + \alpha \Delta x^k \geq 0$ and $s^k + \alpha \Delta s^k \geq 0$ respectively. In practical implementations, we choose

$$\alpha^{\text{pri}}_k = \min\left(1, \eta_k \alpha^{\text{pri}}_{k,\max}\right), \quad \alpha^{\text{dual}}_k = \min\left(1, \eta_k \alpha^{\text{dual}}_{k,\max}\right),$$

where $\eta_k \in [0.9, 1.0)$ is chosen so that $\eta_k \to 1$ as iterates approach the primal-dual solution, to accelerate the asymptotic convergence.

3.3 Starting Point

What we provide here is a heuristic that finds a starting point that satisfies the equality constraints in the primal and dual problems reasonably well, while maintaining positivity of the x and s components and avoiding excessively large values of these components.
First, we find $(\tilde{x}, \tilde{\lambda}, \tilde{s})$ by solving the problems below:

$$\min_x \frac{1}{2} x^T x \text{ subject to } Ax = b,$$
$$\min_{(\lambda,s)} \frac{1}{2} s^T s \text{ subject to } A^T \lambda + s = c.$$

Then we can find solutions explicitly:

$$\tilde{x} = A^T \left(AA^T\right)^{-1} b, \quad \tilde{\lambda} = \left(AA^T\right)^{-1} Ac, \quad \tilde{s} = c - A^T \tilde{\lambda}.$$

Then to reduce nonnegative components, we define:

$$\delta_x = \max\left(-(3/2) \min_i \tilde{x}_i, 0\right), \quad \delta_s = \max\left(-(3/2) \min_i \tilde{s}_i, 0\right),$$

and adjust x and s as follows:

$$\hat{x} = \tilde{x} + \delta_x e, \quad \hat{s} = \tilde{s} + \delta_s e,$$

and obviously, we have $\hat{x} \geq 0, \hat{s} \geq 0$. Moreover, to ensure that components of x^0 and s^0 are not too close to zero and not too dissimilar, we add two more scalars defined as follows:

$$\hat{\delta}_x = \frac{1}{2}\frac{\hat{x}^T\hat{s}}{e^T\hat{s}}, \quad \hat{\delta}_s = \frac{1}{2}\frac{\hat{x}^T\hat{s}}{e^T\hat{x}}$$

Finally, we define the starting point as follows:

$$x^0 = \hat{x} + \hat{\delta}_x e, \quad \lambda^0 = \tilde{\lambda}, \quad s^0 = \hat{s} + \hat{\delta}_s e.$$

3.4 Solving Linear Systems

As the coefficient matrix in the linear systems are always large and sparse with special structures, so an alternative way is proposed to solve this linear system that fully exploits the special structure. Consider the equation below:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -r_{xs} \end{bmatrix}.$$

It is obvious that X and S are diagonal and nonsingular, so we can eliminate Δs to derive the following equation:

$$\begin{bmatrix} -D^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -r_c + X^{-1}r_{xs} \\ -r_b \end{bmatrix},$$
$$\Delta s = -X^{-1}r_{xs} - X^{-1}S\Delta x,$$

where $D = S^{-1/2}X^{1/2}$. By solving this *augmented system*, we obtain:

$$\begin{aligned} AD^2A^T\Delta\lambda &= -r_b - AXS^{-1}r_c + AS^{-1}r_{xs} \\ \Delta s &= -r_c - A^T\Delta\lambda, \\ \Delta x &= -S^{-1}r_{xs} - XS^{-1}\Delta s, \end{aligned}$$

thus the equation has been solved.
Notice that for AD^2A^T, this is theoretically positive definite, so we can apply direct sparse Cholesky factorization to it, then perform two triangular solves to get $\Delta\lambda$, while in practical, AD^2A^T is sometimes ill-conditioned so some modifications may be needed. The way to handle this problem is beyond the scope of this course.

3.5 Framework of Algorithm

Algorithm 4 Predictor-Corrector Algorithm

Require: (x^0, λ^0, s^0) described as above;

for $k = 0, 1, 2 \ldots$ **do**

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x^{\text{aff}} \\ \Delta \lambda^{\text{aff}} \\ \Delta s^{\text{aff}} \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe \end{bmatrix}$$

to get $\left(\Delta x^{\text{aff}}, \Delta \lambda^{\text{aff}}, \Delta s^{\text{aff}}\right)$;

Calculate $\alpha_{\text{aff}}^{\text{pri}}$, $\alpha_{\text{aff}}^{\text{dual}}$ and μ_{aff} as described above;

Set centering parameter

$$\sigma = \left(\frac{\mu_{\text{aff}}}{\mu}\right)^3;$$

Solve

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \\ \Delta s \end{bmatrix} = \begin{bmatrix} -r_c \\ -r_b \\ -XSe - \Delta X^{\text{aff}} \Delta S^{\text{aff}} e + \sigma \mu e \end{bmatrix}$$

to get $(\Delta x, \Delta \lambda, \Delta s)$;

Calculate α_k^{pri} and α_k^{dual}, and set

$$x^{k+1} = x^k + \alpha_k^{\text{pri}} \Delta x,$$
$$\left(\lambda^{k+1}, s^{k+1}\right) = \left(\lambda^k, s^k\right) + \alpha_k^{\text{dual}} (\Delta \lambda, \Delta s).$$

end for

Notice that no convergence theory is available for this algorithm, in fact, there are examples for which the algorithm diverges. When the linear programming problem is infeasible or unbounded, the algorithm above typically diverges, with infeasibilities r_b^k and r_c^k and/or the duality measure μ_k goint to ∞.