10-725/36-725: Convex Optimization **Spring 2015**

Lecture 16: Primal-dual interior point methods I

Lecturer: Ryan Tibshirani *Scribes: Matt Barnes, Hanzhang Hu, Nam Doan*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

Last time, we have discussed about *barrier method*. Barrier method is used to solve problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0,\ i = 1, \ldots, m. \\ & Ax = b \end{aligned} \tag{16.1}$$

By introducing variable t to eliminate the inequality constraints, we have

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & tf(x) + \phi(x) \\ \text{subject to} \quad & Ax = b \end{aligned} \tag{16.2}$$

where $\phi(x)$ is the log-barrier function $\phi(x) = -\sum_{i=1}^{m} \log(-h_i(x))$.

16.1 Linear programming and its duality

The standard form of linear programming or **primal** problem is

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned} \tag{16.3}$$

where $A \in \mathbb{R}^{m \text{ x } n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. One note is that any linear program could be written in the standard form. Moreover, we also assume that A is full row-rank (this condition could be viewed that there are no linear dependence inside matrix A, otherwise we could remove some rows of A because of redundancy)

The dual of the above problem is

$$\begin{aligned} \max_{y} \quad & b^T y \\ \text{subject to} \quad & A^T y \leq c \end{aligned} \tag{16.4}$$

Adding slack variable s to the dual, we have a new form

$$\begin{aligned} \max_{y,s} \quad & b^T y \\ \text{subject to} \quad & A^T y + s = c \\ & s \geq 0 \end{aligned} \tag{16.5}$$

Theorem(Weak duality): *Assume x is primal feasible and y is dual feasible then*

$$b^T y \leq c^T x \tag{16.6}$$

In other word, the objective value of dual is the lower bound of objective value of primal.

Proof: Assume that x is primal feasible and y is dual feasible, we have

$$c^T x - b^T y = (A^T y + x)^T x - (Ax)^T y = s^T x \geq 0 \tag{16.7}$$

where $s \geq 0$ is the dual feasibility condition and $x \geq 0$ is the primal feasibility condition.

Theorem(Strong duality): *Assume primal LP is feasible then it is bounded if and only if the dual is feasible. In that case, their optimal values are the same and they are attained.*

This is the important property of linear programming because this case leads to the equality of objective function of both primal and dual problems. In this case, $c^T x - b^T y = s^T x = 0$ (from weak duality) but $x, s \geq 0$ so $s_i x_i = 0 \ \forall i$.

Since the strong and weak duality, the point x^* and (y^*, s^*) are respectively primal and dual optimal solutions if and only if x^* is a feasible point of primal problem and (y^*, s^*) is a feasible point of dual problem. In other words, (x^*, y^*, s^*) is the solution of

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ x_j s_j &= 0, j = 1, \cdots, n \\ x, s &\geq 0 \end{aligned} \tag{16.8}$$

The first two conditions and the last one guarantee (x^*, y^*, s^*) is feasible for both the primal and dual problems. The third condition is derived from weak duality and it is called complementary condition. Moreover, these conditions are exactly the KKT conditions.

There are two main classes of algorithms for linear programming

- Simplex method: It was developed by Dantzig around 1940 and is still one of the most popular algorithms for linear programming. Its idea is to maintain first three condition and aim for the fourth one.
- Interior-point method: Unlike the *Simplex method*, it tries to maintain the first two and the fourth conditions while aiming for the third condition.

For history of *simplex* and *interior-point* methods, please read slide 9 of this lecture.

We can apply the barrier method from Equation 16.2 to eliminate inequality constraint of primal problem in Equation 16.3

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c^T x - \tau \sum_{i=1}^{n} \log x_i \\ \text{subject to} \quad & Ax = b \end{aligned} \tag{16.9}$$

Similarly to the dual in Equation 16.5, we have

$$\begin{aligned} \max_{y,s} \quad & b^T y + \tau \sum_{i=1}^{n} \log s_i \\ \text{subject to} \quad & A^T x + s = c \end{aligned} \tag{16.10}$$

Note: $\tau = \frac{1}{t} > 0$. Moreover, in dual problem, we maximize the objective so we need to flip the sign of barrier function from negative to positive.

Take the dual of problem in 16.9, we have

$$L(x, y) = c^T x - \tau \sum \log x_i + y^T (b - Ax) = (c - A^T y)^T x - \tau \sum \log x_i + b^T y \tag{16.11}$$

So the dual objective is $\min_x L(x, y)$. If $(c - A^T y)^T$ has one single negative component, it will make the dual objective infinite. So, to make dual objective finite, we require $c - A^T y > 0$

Let $s = (c - A^T y)$, then $s_i > 0$ for all i. We also have:

$$L(x, y) = \sum_i (s_i x_i - \tau \log x_i) + b^T y$$

Thus, we can minimize component wise w.r.t. x, and have:

$$\min_{x_i} s_i x_i - \tau \log x_i = \tau - \tau \log(\tau / s_i),$$

since the optimal $x_i = \tau / s_i$. Thus

$$\min_x L(x, y) = n\tau - \tau \sum \log \frac{\tau}{s_i} + b^T y$$

So the dual problem, modulo a constant $(n\tau)$, is the problem 16.10.

16.2 Primal-dual Central Path and Key Idea of Primal-dual Interior Point Method

The KKT condition of the pair of barrier problem says the optimal solutions satisfies:

$$\begin{aligned} & Ax = b \\ & A^T y + s = c \\ & x_j s_j = \tau \\ & x, s > 0 \end{aligned} \tag{16.12}$$

which we can call a perturbed KKT condition of the original problem (before adding the barrier functions). We define the **primal-dual central path** as the set of $\{x(\tau), y(\tau), s(\tau) : \tau > 0\}$, where for each $\tau > 0$,

$x(\tau)$ and $(y(\tau), s(\tau))$ are solutions to the pair of barrier problems. Note that for each τ, there is at most one triple (x, y, s), since both primal and dual have strictly convex objectives due to the log barrier function.

The key idea of the algorithm is then to generate (x^k, y^k, s^k) at each step k, in order to approximate $(x(\tau^k), y(\tau^k), s(\tau^k))$, where $\tau^k > 0$ is a decreasing sequence. There are three details in implementing this idea: (1) measurement of proximity to the central path, (2) behavior of τ^k, and (3) update rule of (x^k, y^k, s^k).

We define the strictly feasible set as

$$\mathcal{F}^0 := \{(x, y, s) : Ax = b, A^T y + s = c, x, s > 0\}. \tag{16.13}$$

For x, s in $\mathbb{R}^n$, we define $X := \text{diag}(x)$, and $S := \text{diag}(s)$, and the vector $(x_1 s_1, x_2 s_2, ..., x_n s_n)$ can be written as $XS\mathbf{1}$. For x, s in $\mathbb{R}^n_+$ we define $\mu(x, s) := \frac{x^T s}{n}$.

A couple observations of the above definitions: (1) Recall that $x_i s_i \geq 0$, so $XS\mathbf{1} \in \mathbb{R}^n_{++}$, and such mapping from $(x, y, s) \in \mathcal{F}^0$ to $\mathbb{R}^n_{++}$ is a bijection (not proved in class). Hence we can study the central path in the space of $\mathbb{R}^n_{++}$ using $XS\mathbf{1}$. (2) Recall that the duality gap of the original probelm $c^T x - b^T y = x^T s \geq 0$, so intuitively $\mu(x, s)$ measures how close the solution is to the optimality.

Now we can define two types of neighborhoods of the central path: For $\theta \in (0, 1)$, the two-norm neighborhood of the central path is:

$$\mathcal{N}_2(\theta) := \{(x, y, s) \in \mathcal{F}^0 : \|XS\mathbf{1} - \mu(x, s)\mathbf{1}\|_2 \leq \theta_\mu(x, s)\}. \tag{16.14}$$

For $\gamma \in (0, 1)$, the one-sided infinity-norm neighborhood of the central path is:

$$\mathcal{N}_{-\infty}(\gamma) := \{(x, y, s) \in \mathcal{F}^0 : x_i s_i \geq \gamma\mu(x, s), i = 1, ..., n\}. \tag{16.15}$$

To understand $\mathcal{N}_2(\theta)$, we note that when (x, y, s) is on the central path iff there is some $\tau > 0$, such that for all i, $x_i s_i = \tau$, i.e., the vector $XS\mathbf{1} \in \mathbb{R}^n_{++}$ is on the ray $\mathbf{0} + \tau[1, 1, ..., 1]^T$. Hence in the space of $XS\mathbf{1}$, $\mathcal{N}_2(\theta)$ defines a cone sandwiching the $[1, 1, ...1]$ ray, which is the central path after the $XS\mathbf{1}$ mapping. The larger θ is, the wider the cone.

Similarly, $\mathcal{N}_{-\infty}(\gamma)$ also defines a cone in the space of $XS\mathbf{1}$. However, the cone can be much wider than its two-norm counterpat. In particular, as γ approaches 0, $\mathcal{N}_{-\infty}(\gamma)$ approaches $\mathcal{F}^0$. Note that this doesn't happen to $\mathcal{N}_2(\theta)$ when θ approaches 1.

We will see later in the algorithm how τ is updated. To update (x^k, y^k, s^k), we will use the following Newton's method for solving equations: recall that $(x(\tau), y(\tau), s(\tau))$ solves

$$\begin{bmatrix} A^T y + s - c \\ Ax - b \\ XS\mathbf{1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau\mathbf{1} \end{bmatrix}, x, s > 0 \tag{16.16}$$

Then for each Newton step, we update with $(\Delta x, \Delta y, \Delta s)$ satisfying the following:

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \tau\mathbf{1} - XS\mathbf{1} \end{bmatrix}. \tag{16.17}$$

Recall that if we were to solve $f(x) = 0$, Newton's step update Δx satisfies that $\nabla f(x)^T \Delta x = -f(x)$, and $x^+ = x + \Delta x$.

16.3 Short-step path following algorithm

Let $\theta, \delta \in (0, 1)$ be such that $\frac{\theta^2 + \delta^2}{2^{3/2}(1-\theta} \leq \left(1 - \frac{delta}{\sqrt{n}}\right)\theta$

Let $(x^0, y^0, s^0) \in \mathcal{N}_2(\theta)$
For $k = 0, 1, \ldots$
Compute Newton step for
$(x, y, s) = (x^k, y^k, s^k), \tau = \left(1 - \frac{\delta}{\sqrt{n}}\right) \mu(x, s)$
Set $(x^{k+1}, y^{k+1}, s^{k+1}) := (x^k, y^k, s^k) + (\Delta x, \Delta y, \Delta s)$

For example, one could choose $\theta = \delta = 0.4$ to satisfy the first inequality constraint in the SPF algorithm. The algorithm begins with a point in the neighborhood of the central path (i.e. $(x^0, y^0, s^0) \in \mathcal{N}_2(\theta)$). Then at each step, we compute the Newton step in the direction a little further along the central path (i.e. a slightly smaller value of τ). This can be thought of as 'chasing' the central path.

Notice the difference between this dual-primal interior point method and the log-barrier method from Lecture 15. There is no need for an inner and an outer loop! At each iteration, we are only taking a single unit Newton step and guarantee the new solution is still in the neighborhood of the central path ($\in \mathcal{N}_2(\theta)$). Further, the new solution is guaranteed to decrease $\mu(x^k, s^k)$, i.e. it moves in the correct direction.

Theorem 16.1 *The sequence generated by Algorithm SPF satisfies*

$$(x^k, y^k, s^k) \in \mathcal{N}_2(\theta) \tag{16.18}$$

and

$$\mu(x^{k+1}, s^{k+1}) = \left(1 - \frac{\delta}{\sqrt{n}}\right) \mu(x^k, s^k) \tag{16.19}$$

Corollary 16.2 *In* $\mathcal{O}\left(\sqrt{n} \log\left(\frac{n\mu(x^0, s^0)}{\epsilon}\right)\right)$ *the algorithm yields* $(x^k, y^k, s^k) \in \mathcal{F}^0$ *such that*

$$c^T x_k - b^T y_k \leq \epsilon \tag{16.20}$$

Proof: Only a sketch for the proof of Theorem 16.1 was covered in class. Here, we want to iteratively show $x^+ = x^{k+1} = x + \Delta x$, $y^+ = y^{k+1} = y + \Delta y$ and $s^+ = s^{k+1} = s + \Delta s$ all stay in the neighborhood of the central path $\mathcal{N}_2$.

$$\begin{aligned} Ax^+ &= Ax + A\Delta x && (16.21)\\ &= b + 0 && (16.22)\\ &= b && (16.23) \end{aligned}$$

because x was in the neighborhood of the central path (thus $x \in \mathcal{F}^0$, so $Ax = b$) and $A\Delta x = 0$ from the Newton step equations.

Likewise,

$$\begin{aligned} A^T y^+ + s^+ &= A^T y + s + A^T \Delta y + \Delta s && (16.24)\\ &= c + 0 && (16.25)\\ &= c && (16.26) \end{aligned}$$

because $A^T y + s = c$ when y and s were both in the neighborhood of the central path and $A^T \Delta y + \Delta s = 0$ from the Newton step. Thus, the new solution (x^+, y^+, s^+) is feasible (i.e. $\in \mathcal{F}^0$).

The difficult part of the proof is to show x^+ and s^+ are positive and the inequality $||XS\mathbb{1} - \mu(x,s)\mathbb{1}||_2 \leq \theta\mu(x,s)$ holds. We do not consider these issues here, and instead proceed to the second part of Theorem 16.1 (Eq 16.19) and Corollary 16.2. If you look at

$$\begin{aligned} X^+S^+\mathbb{1} &= (X+\Delta X)(S+\Delta S)\mathbb{1} && (16.27)\\ &= X\Delta S + S\Delta X + \Delta X\Delta S\mathbb{1} + XS\mathbb{1} && (16.28)\\ &= \tau\mathbb{1} - XS\mathbb{1} + \Delta X\Delta S\mathbb{1} + XS\mathbb{1} && (16.29)\\ &= \tau\mathbb{1} + \Delta X\Delta S\mathbb{1} && (16.30)\end{aligned}$$

where $X\Delta S + S\Delta X = \tau\mathbb{1} - XS\mathbb{1}$ from the Newton steps.

$$\begin{aligned} \mu(x^+,s^+) &= \frac{1}{n}(x^+)^T s^+ && (16.31)\\ &= \frac{1}{n}(n\tau + \Delta x^T\Delta s) && (16.32)\\ &= \tau + \frac{1}{n}\Delta x^T\Delta s && (16.33)\\ &= \left(1-\frac{\delta}{\sqrt{n}}\right)\mu(x,s) + \frac{1}{n}\Delta x^T\Delta s && (16.34)\\ &= \left(1-\frac{\delta}{\sqrt{n}}\right)\mu(x,s) && (16.35)\end{aligned}$$

where $\tau = \left(1-\frac{\delta}{\sqrt{n}}\right)\mu(x,s)$ by the definition of the SPF algorithm and $\Delta x^T\Delta s = 0$ by using two of the Newton step equations:

$$\begin{aligned} &\begin{cases} A^T\Delta y + \Delta s = 0 \\ A\Delta x = 0 \end{cases} && (16.36)\\ &\Delta x^T A^T\Delta y + \Delta x^T\Delta s = 0 && (16.37)\\ &0^T\Delta y + \Delta x^T\Delta s = 0 && (16.38)\\ &\Delta x^T\Delta s = 0 && (16.39)\end{aligned}$$

After k iterations, Eq 16.35 becomes

$$\mu(x^k,s^k) = \left(1-\frac{\delta}{\sqrt{n}}\right)^k \mu(x^0,s^0) \tag{16.40}$$

Lastly, the corollary follows from the previous equation:

$$\begin{aligned} c^Tx - b^Ty = n\mu(x,s) &\leq \epsilon && (16.41)\\ n\left(1-\frac{\delta}{\sqrt{n}}\right)^k \mu(x^0,s^0) &\leq \epsilon && (16.42)\\ \left(1-\frac{\delta}{\sqrt{n}}\right)^k &\leq \frac{\epsilon}{n\mu(x^0,s^0)} && (16.43)\\ k\log\left(1-\frac{\delta}{\sqrt{n}}\right) &\leq \log\left(\frac{\epsilon}{n\mu(x^0,s^0)}\right) && (16.44)\\ k &\geq \mathcal{O}\sqrt{n}\log\left(\frac{n\mu(x^0,s^0)}{\epsilon}\right) && (16.45)\end{aligned}$$

(this part was not explicitely derived in class, I just did it here). ■

This is called the 'short-step' path following algorithm because the $\mathcal{N}_2$ neighborhood is so narrow the steps must be very small.

16.4 Long-step path following algorithm

The idea of long-step path following is very similar, except using the larger neighborhood of $\mathcal{N}_{-\infty}$. The only major difference is long-step path following may require the use of line search to stay in the neighborhood (ie. the choice of α_k).

Let $\gamma \in (0,1)$ and $0 < \sigma_{min} < \sigma_{max} < 1$
Let $(x^0, y^0, s^0) \in \mathcal{N}_{-\infty}(\gamma)$
For $k = 0, 1, \ldots$
  Choose $\sigma \in [\sigma_{min}, \sigma_{max}]$
  Compute Newton step for
    $(x, y, s) = (x^k, y^k, s^k), \tau = \sigma\mu(x^k, s^k)$
  Choose α_k as the largest $\alpha \in [0, 1]$ such that
    $(x^k, y^k, s^k) + \alpha(\Delta x, \Delta y, \Delta s) \in \mathcal{N}_{-\infty}(\gamma)$
  Set $(x^{k+1}, y^{k+1}, s^{k+1}) := (x^k, y^k, s^k) + \alpha_k(\Delta x, \Delta y, \Delta s)$

Theorem 16.3 *The sequence generated by Algorithm LPF satisfies*

$$(x^k, y^k, s^k) \in \mathcal{N}_{-\infty}(\theta) \tag{16.46}$$

and

$$\mu(x^{k+1}, s^{k+1}) \leq \left(1 - \frac{\delta}{n}\right) \mu(x^k, s^k) \tag{16.47}$$

for some constant δ that depends on $\gamma, \sigma_{min}, \sigma_{max}$ but not on n

Corollary 16.4 *In $\mathcal{O}\left(n \log\left(\frac{n\mu(x^0,s^0)}{\epsilon}\right)\right)$ the algorithm yields $(x^k, y^k, s^k) \in \mathcal{F}^0$ such that*

$$c^T x_k - b^T y_k \leq \epsilon \tag{16.48}$$

Theoretically, the bound on the long-step algorithm requires more iterations than the bound on the short-step algorithm. However, the long-step usually performs better in practice.

16.5 Infeasible interior-point algorithms

Both SPF and LPF require finding an initial point $(x^0, y^0, s^0) \in \mathcal{F}^0$. The IPF algorithm only requires $x^0, s^0 > 0$ but does not guarantee the solutions x^k, s^k, y^k stay in some neighborhood. However, μ values will converge to 0 linearly, as do the residuals.

We did not have time to cover the actual IPF algorithm, nor other topics on 'IPF for more general convex optimization' and the 'Primal-Dual Algorithm.'