ORIE 6300 Mathematical Programming I November 13, 2014

Lecture 23

Lecturer: David P. Williamson *Scribe: Mukadder Sevi Baltaoglu*

1 Interior Point Methods

Consider the standard primal and dual linear programs:

$$\begin{array}{rl} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad\qquad \begin{array}{rl} \max & b^T y \\ \text{s.t.} & A^T y + s = c \\ & s \geq 0 \end{array}$$

and define

$$\mathcal{F}^o(P) = \{x \in \Re^n : Ax = b, x > 0\}$$
$$\mathcal{F}^o(D) = \{(y, s), y \in \Re^m, s \in \Re^n : A^T y + s = c, s > 0\}$$

Last time, we looked at $B_\mu(x) = c^T x + \mu \sum_{j=1}^n ln(x_j)$ for $x \in \mathcal{F}^o(P)$, $\mu > 0$ and proved the following theorem:

Theorem 1 *If $\mathcal{F}^o(P)$ and $\mathcal{F}^o(D)$ are nonempty, then a necessary and sufficient condition for x to be the unique minimizer of $B_\mu(x)$ on $\mathcal{F}^o(P)$ is that there exists $(y, s) \in \mathcal{F}^o(D)$ such that*

$$A^T y + s = c$$
$$Ax = b$$
$$XSe = \mu e \Leftrightarrow x_j s_j = \mu \quad \forall j$$
$$where\ X = diag(x), S = diag(s)$$

Let's denote the problem of finding $(y, s) \in \mathcal{F}^o(D)$ that satisfies the conditions of Theorem 1 as $(*)$ and let's denote the solution of the problem as $x(\mu)$, $y(\mu)$, $s(\mu)$ for some fixed $\mu > 0$. Then, $\{x(\mu), y(\mu), s(\mu) : \mu > 0\}$ is called the primal-dual central path. Now, we will proceed to find the solution of $(*)$.

1.1 Newton Direction

We can solve the problem $(*)$ with a method based on Newton's method for finding a zero of a function. In the one dimensional, unconstrained case, in order to find a zero of a function $f(x)$, we start with an initial point x_0 and then repeatedly update the point until the value $f(x_k)$ of the function is sufficiently close to zero as follows:

$$f'(x_k)\Delta x + f(x_k) = 0 \text{ where } \Delta x = x_{k+1} - x_k$$

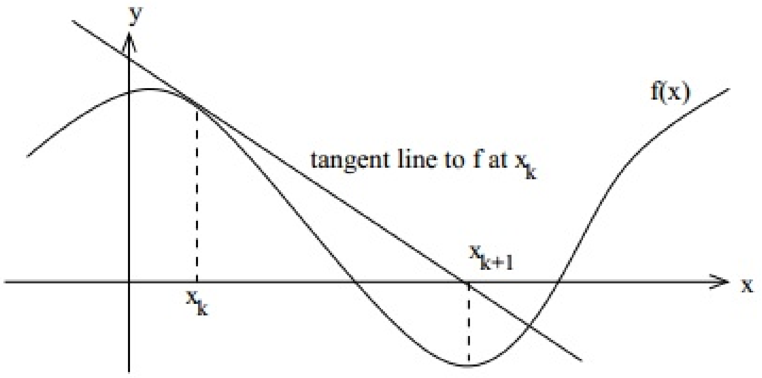

Figure 1: An update by Newton's Method in the one dimensional, unconstrained case

Interior point methods apply the same approach to find the solution of (*) as follows. We want to find a zero of the function

$$F(x,y,s) = \begin{pmatrix} A^Ty+s-c \\ Ax-b \\ XSe-\mu e \end{pmatrix}$$

with $(x,s) \geq 0$.

In order to do this, we make use of the Jacobian J, which is a matrix of partial derivatives of F.

$$J(x,y,s) = \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix}$$

Then, a Newton direction $(\Delta x, \Delta y, \Delta s)$ is a solution to the following equation:

$$J(x,y,s) \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} + F(x,y,s) = 0$$

Now the question is for what values of μ should we try to solve (*). We now make a nasty change in notation, by redefining $\mu \equiv \frac{1}{n}\sum_{i=1}^{n} x_i s_i = \frac{1}{n}x^T s$, that is, given our current solution, solving the system for this value of μ makes the $x_i s_i$ equal for all i. To balance the movement towards the central path against the movement toward optimal solutions, we maintain a centering parameter $\sigma \in [0,1]$. If $\sigma = 1$, then our update will move towards the center of the feasible region. On the other hand, if $\sigma = 0$, then our update step is in the direction of optimal solution to the LP. A step with $\sigma = 1$ is referred to as a centering step and a step with $\sigma = 0$ is referred to as an affine-scaling step. The choice of the centering parameter σ provides us with a trade-off between moving towards the central path and moving towards optimal solution.

Now, we want to find the solution to the following system of equations:

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = - \begin{pmatrix} A^Ty+s-c \\ Ax-b \\ XSe-\sigma\mu e \end{pmatrix}$$

Since (x, y, s) is already feasible, we can write this equation as;

$$\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -XSe + \sigma\mu e \end{pmatrix}$$

for (x, y, s) feasible.

Some interior point methods examine only the feasible region of the primal LP and as such are known as pure primal interior point methods. The interior point algorithm that we presented here examines the feasible regions of both the primal and the dual LPs and so it is referred to as primal-dual interior point algorithm.

Primal-Dual Interior-Point

```
$(x^0, y^0, s^0) \leftarrow$ initial feasible point $(x^0, s^0 > 0)$
$\mu^0 \leftarrow \frac{1}{n}(x^0)^T s^0$
$k \leftarrow 0$
While $\mu^k > \epsilon$
    Solve $\begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta y^k \\ \Delta s^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k S^k e + \sigma^k \mu^k e \end{pmatrix}$
    $(x^{k+1}, y^{k+1}, s^{k+1}) \leftarrow (x^k, y^k, s^k) + \alpha^k(\Delta x^k, \Delta y^k, \Delta s^k)$
      where $\alpha^k$ is such that $x^{k+1}, s^{k+1} > 0$
    $\mu^{k+1} \leftarrow \frac{1}{n}(x^{k+1})^T s^{k+1}$
    $k \leftarrow k + 1$
```

where $\mu = \frac{x^T s}{n}$ and $\sigma \in [0, 1]$ is a centering parameter.

This general framework omits several details that must be addressed in any implementation of an interior-point algorithm for LP. In particular, we have not specified how the centering parameter σ^k is chosen, as different interior-point algorithms use different methods to select σ^k. Furthermore, we have not considered how to find an initial solution to the LP, when we terminate, how we know that we are making progress, or how to choose the scaling parameter α^k.

During the interior-point method, we keep away from the boundary by staying in some *neighborhood* of central path. There are several common types of neighborhoods used by interior point algorithms. For example, for a parameter θ, several versions of the algorithm use a neighborhood that uses the L_2 norm to measure the distance is defined as follows:

$$N_2(\theta) = \{x \in \mathcal{F}^o(P), (y, s) \in \mathcal{F}^o(D) : \|XSe - \mu e\|_2 \leq \Theta\mu\}.$$

1.2 Path Following Methods

Path-following algorithms use update steps that follow the central path. The extent to which a path-following algorithm follows the central path is determined by the centering parameter σ. The method of choosing σ distinguishes different path-following algorithms.

- **Long-Step**: Long step algorithms pick the centering parameter σ far from 1, as a result, the solutions are farther from the central path. This algorithm requires at most $O(n \ln(1/\epsilon))$ iterations to achieve $\mu < \epsilon$. These algorithms work well in practice.

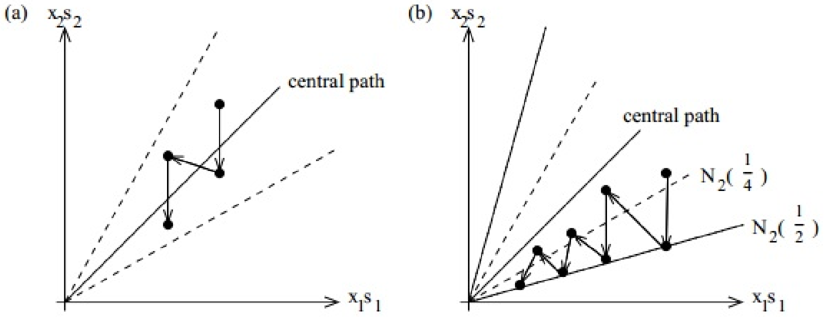

Figure 2: (a) A neighborhood of the central path in the case of n=2 variables (b) The predictor-corrector algorithm alternates between predictor steps, in which it moves as far as possible while remaining in $N_2(1/2)$, and corrector steps, in which it takes a full step ($\alpha = 1$), returning to $N_2(1/4)$

- **Short-Step**: Short step algorithm picks the centering parameter σ close to 1, so that the solutions stay near the central path. This algorithm requires at most $O(\sqrt{n}\ln(1/\epsilon))$ iterations to achieve $\mu < \epsilon$. Despite the better theoretical guarantee, these algorithms typically perform worse in practice than long-step algorithms.

- **Predictor-Corrector**: This algorithm strikes a balance between moving toward the optimal solution and following the central path by alternating between steps with $\sigma = 0$ (predictor), and $\sigma = 1$ (corrector). This algorithm requires at most $O(\sqrt{n}\ln(1/\epsilon))$ iterations to achieve $\mu < \epsilon$. Commercial codes typically implement this variant of the algorithm.

1.3 Short-Step Path Following Algorithm in Detail

To give some of the flavor of the analysis of the algorithm, we now consider a short step path following algorithm, characterized by the parameters α^k and σ^k. We choose $\alpha^k = 1$ so that we take a full update step in the Newton direction each iteration. We choose the centering parameter to be $\sigma^k = 1 - \frac{0.4}{\sqrt{n}}$ at each iteration. Note that the centering parameter is close to unity so that the steps taken by the algorithm are indeed short. We wish to establish a complexity bound for the short step algorithm.

We first claim the following lemma, which states that as long as the initial points are within some neighborhood of the central path, feasibility is guaranteed, and we stay away from the boundary.

Lemma 2 *If $(x^0, y^0, s^0) \in N_2(0.4)$ then $(x^k, y^k, s^k) \in N_2(0.4)$ for all k.*

Lemma 3 $\mu^{k+1} = \sigma^k \mu^k = \left(1 - \frac{0.4}{\sqrt{n}}\right)\mu^k$ *for all k.*

Theorem 4 *If $(x^0, y^0, s^0) \in N_2(0.4)$ and $\mu^0 \leq C$, then after $K = O(\sqrt{n}\ln(\frac{C}{\epsilon}))$ iterations $\mu^k \leq \epsilon$.*

Proof:

$$\mu^K \leq \left(1 - \frac{0.4}{\sqrt{n}}\right)^K \mu^0 \leq \left(1 - \frac{0.4}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{0.4}\ln(\frac{C}{\epsilon})} C.$$

By using $1-x \leq e^{-x}$, we get that

$$\mu^K \leq (e^{-\frac{0.4}{\sqrt{n}}})^{\frac{\sqrt{n}}{0.4}\ln(\frac{C}{\epsilon})} C = e^{-\ln(C/\epsilon)} C = \epsilon.$$

□

Recall that $\mu = \frac{1}{n}x^T s = \frac{1}{n}x^T(c - A^T y) = \frac{1}{n}(x^T c - (Ax)^T y) = \frac{1}{n}(x^T c - b^T y)$. Since $\mu^K \leq \epsilon$, $x^T c - b^T y \leq n\epsilon$. So, primal and dual solutions are within $n\epsilon$ of optimal.

We can now prove Lemma 3 in more general form.

Lemma 5 $(\Delta x^k)^T \Delta s^k = 0$ *and* $\mu^{k+1} = (1 - \alpha^k(1-\sigma^k))\mu^k$*.*

Observe that for $\alpha^k = 1$, $\sigma^k = 1 - \frac{0.4}{\sqrt{n}}$ then $\mu^{k+1} = (1 - \frac{0.4}{\sqrt{n}})\mu^k$ as desired.

Proof: We know $A^\Delta y^k + \Delta s^k = 0$ and $A\Delta x^k = 0$. Then,

$$(\Delta x^k)^T \Delta s^k = (\Delta x^k)^T(-A^T \Delta y^k) = -(A\Delta x^k)^T \Delta y^k = 0.$$

Also,

$$\begin{gathered} S^k \Delta x^k + X^k \Delta s^k = -X^k S^k e + \sigma^k \mu^k e \\ \Rightarrow s_i^k \Delta x_i^k + x_i^k \Delta s_i^k = -x_i^k s_i^k + \sigma^k \mu^k \quad \forall i \\ \Rightarrow (s^k)^T \Delta x^k + (x^k)^T \Delta s^k = -(x^k)^T s^k + n\sigma^k \mu^k. \end{gathered}$$

Hence,

$$\begin{aligned} \mu^{k+1} = \frac{1}{n}(x^{k+1})^T(s^{k+1}) &= \frac{1}{n}(x^k + \alpha^k \Delta x^k)^T(s^k + \alpha^k \Delta s^k) \\ &= \frac{1}{n}[(x^k)^T s^k + \alpha^k((\Delta x^k)^T s^k + (x^k)^T \Delta s^k) + (\alpha^k)^2 (\Delta x^k)^T \Delta s^k]. \end{aligned}$$

Since $(\Delta x^k)^T \Delta s^k = 0$,

$$\mu^{k+1} = \mu^k + \frac{\alpha^k}{n}[-(x^k)^T s^k + n\sigma^k \mu^k] = \mu^k + \frac{\alpha^k}{n}[-n\mu^k + n\sigma^k \mu^k] = (1 - \alpha^k(1-\sigma^k))\mu^k.$$

□