Lecture 17: Interior point methods

Rajat Mittal

IIT Kanpur

The topic for this week will be the Interior point methods. Simplex, as we know, travels from vertex to vertex in search of the optimal solution. It is known that some examples of LP (like *Klee-Minty cube*) have to visit exponential number of vertices for most of the versions of simplex method. The interior point method, as the name suggests, starts with a solution in the inside of the feasible region and continues the progress in the interior to converge to optimal solution. We will provide some intuition about the method and skip the convergence analysis and other technical details. Please take a look at the references mentioned on the course web-page for more detail. The notes below are prepared with the help of these references.

1 Newton's method

Most of you might be familiar with the Newton's method to solve for the zero of a differentiable function on reals. Suppose $f : \mathbb{R} \to \mathbb{R}$ is a function on reals and we have a starting x_0 and $f(x_0)$. Newton's idea was to use the Taylor series expansion and ignore the higher derivatives. Taylor's series says that the function value in the neighborhood of x_0 can be written as,

$$f(x) = f(x_0) + \frac{f^{(1)}(x_0)}{1!}(x - x_0) + \frac{f^{(2)}(x_0)}{2!}(x - x_0)^2 + \cdots ,$$

where $f^{(n)}(x)$ is the n^{th} derivative of f. Hence, the linear approximation can be written as,

$$\bar{f}(x) := f(x_0) + \frac{f^{(1)}(x_0)}{1!}(x - x_0).$$

Newton's method (also known as Newton-Raphson's method) looks for successive x_i's using this approximation. That means, given a solution x_i, the next x_{i+1} is obtained by,

$$x_{i+1} := x_i - \frac{f(x_i)}{f^{(1)}(x_i)}.$$

We will call this a *Newton step.* The method in a nutshell starts with a guess for the zero, x_0 and successively applies the previous equation to improve the guess for the root of the function f.

For the case when the function is vector valued and multi-variable, $f : \mathbb{R}^n \to \mathbb{R}^m$, the next step is given by,

$$x_{i+1} := x_i - J(x_i)^+ f(x_i).$$

Here J is the Jacobian matrix of function f, whose i, j entry is the derivative of the i^{th} co-ordinate of f with respect to j^{th} variable. $J^+ = (J^T J)^{-1} J^T$ is the generalized inverse since the Jacobian might not be square. The term $J(x_i)^+ f(x_i)$ which is added to x_i to obtain a new solution is called the *Newton search direction.* In some cases the Newton search direction is multiplied by some α before being added to x_i. In general this method might not converge to the root of the function. But under some restrictions on the function, this method can converge very quickly to the root. The kind of functions we will encounter will happen to satisfy such restrictions.

Exercise 1. Show that this method is good for linear functions.

We would like to apply this method to solve linear programs.

2 Complementary slackness conditions

Suppose we look at the following linear program,

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

This is almost the standard form, just that minimum and maximum are reversed and the dual has slack variables now.

Exercise 2. Convert both the LP's into standard form.

The complementary slackness condition for these can be written as,

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= 0 \\ x &\geq 0 \\ s &\geq 0 \end{aligned}$$

Here X is the diagonal matrix with diagonal as x and similarly S. e is the all 1's vector. If we ignore the inequality constraints, we can use Newton's method on the first three equations. We start with a feasible solution x, y, s in the interior and apply the Newton step. The problem as expected would be that the x and s of new solution might not be positive. In this case we might go out of feasible region. To overcome this, a penalty in the objective can be applied so as to discourage the solutions from begin negative.

3 Barrier function

The trick is to put a barrier (penalty) function which discourages the solution to get to negative values. Suppose instead of the original linear program, we look at a slight variant,

$$\begin{aligned} \min \; & c^T x - \mu \textstyle\sum_{i=1}^{n} \ln x_j \\ \text{s.t.} \; & \qquad Ax = b \end{aligned}$$

Lets look at the changes in detail.

- μ is some constant and not a variable for this optimization program. As μ tends to zero, the optimization program gets closer to the original LP (assuming the solution is positive).
- The new optimization problem is not an LP.
- The objective function is changed such that whenever x_i goes close to zero, a huge penalty is imposed on the solution. Remember that $\ln x_i$ is a very big negative number for x_i close to zero.
- The constraint $x \geq 0$ is removed because of the new objective function.

The analog of complementary slackness conditions for the above optimization program is,

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ XSe &= \mu e \\ x &\geq 0 \\ s &\geq 0 \end{aligned}$$

This are called the *first order optimality conditions* for the *Lagrangian* of the above optimization program. For details about Lagrange multipliers and first order optimality conditions, please look at the references mentioned on the course web-page.

The basic idea of interior point method is to *approximately* solve these new conditions (the first three) using Newton's method, changing the μ (making it smaller) at every new step. For every $\mu \geq 0$ there is a unique solution $(x(\mu), y(\mu), s(\mu))$ for the above equations. The path of these solutions (for different μ) is called the *central path*. Notice that a bigger μ makes sure that x, s remain positive and our solution is close to central path. A smaller μ gets us closer to the optimal solution. In the interior point methods we choose the μ to be bigger in the beginning and take big Newton steps (stay comfortably inside the interior, follow the central path). To the end, μ is reduced substantially and we get close to optimality.

The Newton step $(\delta x, \delta y, \delta s)$ for the above equation can be written as,

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \\ \delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu e - XSe \end{pmatrix} \tag{1}$$

4 Algorithm

The following algorithm is called *primal dual interior point method* and the specific algorithm is taken from Robert Robere's notes on Interior point method and Linear Programming.

```
Choose $\rho \in (0,1), \mu_0 \geq 0$ ;
Start with a feasible solution of the primal and dual, $x_0, y_0, s_0$ ;
for $i = 1, \cdots$ do
    if Desired accuracy is not reached then
        $\mu_i := \rho\mu_{i-1}$ ;
        Compute the $\delta x, \delta y, \delta s$ from Eqn. 1;
        For a suitable $\alpha$, $(x_i, y_i, s_i) = (x_{i-1}, y_{i-1}, s_{i-1}) + \alpha(\delta x, \delta y, \delta s)$ ;
    else
        Break;
    end
end
```

Algorithm 1: Primal dual interior point method

This pseudocode has many choices.

- ρ: The choice of ρ affects the balance between going for optimal solution and solutions close to central path.
- *Initial solution*: This can be obtained as in the case of earlier algorithms.
- α: It affects the rate of convergence. One good choice is to use the α which maximizes the new objective function (with the logarithmic barrier, without constraints).

- *Desired accuracy*: It can be shown that the duality gap (difference between primal and dual is bounded by $n\mu$. So once the $n\mu$ becomes smaller than acceptable error, the algorithm can be stopped.

It can be shown that with a suitable choice of ρ and α, the algorithm is polynomial in the size of input. We will skip the convergence analysis. Again, for details look at the references mentioned in the course web-page.