# Course Notes for EE227C (Spring 2018): Convex Optimization and Approximation

Instructor: Moritz Hardt

Email: hardt+ee227c@berkeley.edu

Graduate Instructor: Max Simchowitz

Email: msimchow+ee227c@berkeley.edu

April 26, 2018

## 25 Enter interior point methods

In the last lecture, we discussed Newton's method. Although it enjoys a fast local convergence guarantee, global convergence of Newton's method is not guaranteed. In this lecture, we'll introduce interior point methods, which can be thought of as an extension of Newton's method to ensure global convergence. We will first introduce the main idea of *barrier methods* at great generality, before we specialize our analysis to linear programming.

### 25.1 Barrier methods

Barrier methods replace inequality constraints with a so-called *barrier* function that is added to objective function of the optimization problem. Consider the following optimization problem:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega, \\ & g_j(x) \leqslant 0\,, \qquad j = 1, 2, \cdots, r\,, \end{aligned}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g_j : \mathbb{R}^n \to \mathbb{R}$ are given functions. The function $f$ is continuous, and $\Omega$ is a closed set. For the rest of the lecture, we assume convex $g_j$ and $\Omega = \mathbb{R}^n$. And we denote $x^*$ as the optimal solution of the problem.

**Definition 25.1** (Interior of the constraint region)**.** The interior (relative to $\Omega$) of the constraint region is defined as $S = \{x \in \Omega : g_j(x) < 0,\ j = 1, 2, \cdots, r\}$.

Assuming nonempty and convex $S$, we define a so-called barrier function $B(x)$ defined on $S$, such that $B(x)$ is continuous the function blows up as we approach the boundary of the constraint region. More formally, $\lim_{g_j(x) \to 0\_} B(x) = \infty$. Two most common examples are logarithmic barrier function and inverse barrier function:

$$\text{Logarithmic:} \qquad B(x) = -\sum_{j=1}^{r} \ln\{-g_j(x)\} \tag{1}$$

$$\text{Inverse:} \qquad B(x) = -\sum_{j=1}^{r} \frac{1}{g_j(x)}. \tag{2}$$

Both of them are convex if all $g_j(x)$ are convex.

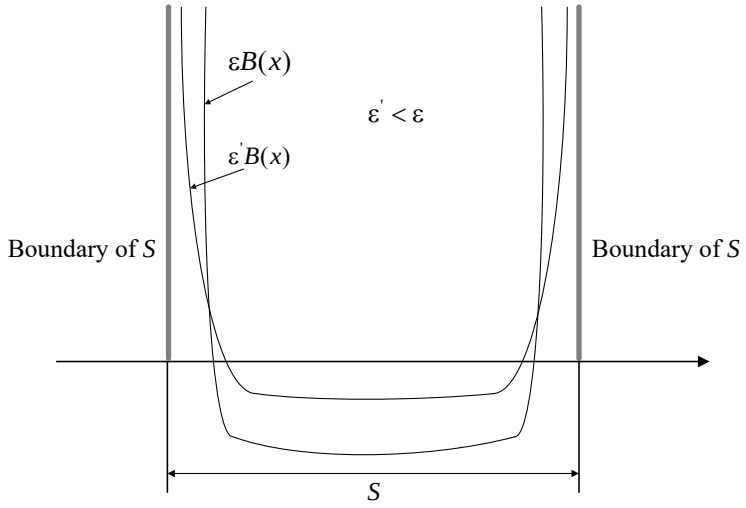


Figure 1: Form of a barrier term

Given a barrier function $B(x)$, define a new cost function $f_\epsilon(x) = f(x) + \epsilon B(x)$, where $\epsilon$ is a positive real number. Then we can eliminate the inequality constraints in the original problem and obtain the following problem:

$$\begin{aligned} \min_{x} \quad & f_\epsilon(x) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \tag{3}$$

The form of the barrier term $\epsilon B(x)$ is illustrated in Figure 1.

The barrier method is defined by introducing a sequence $\{\epsilon_t\}$ such that $0 < \epsilon_{t+1} < \epsilon_t$ for $t = 0, 1, 2, ...$ and $\epsilon_t \to 0$. Then we find a sequence $\{x_t\}$ such that $x_t \in \arg\min_{x \in S} f_{\epsilon_t}(x)$. Note that the barrier term $\epsilon_t B(x)$ goes to zero for all interior points $x \in S$ as $\epsilon_t \to 0$, allowing $x_t$ to get increasingly closer to the boundary. Therefore, intuitively, $x_t$ should approach $x^*$ no matter $x^*$ is in the interior or on the boundary of $S$. Its convergence is formalized in the following proposition.

**Proposition 25.2.** *Every limit point of a sequence $\{x_t\}$ generated by a barrier method is a global minimum of the original constrained problem.*

*Proof.* See Proposition 5.1.1 of [Ber16]. ■

The previous proposition shows that the global optima of our barrier problems converge to the global constrained optimum. But how do we solve this sequence of optimization problems. The key intuition is this. An initial interior point can often be obtained easily for some large enough $\epsilon_0$. Then in each iteration, we can use $x_t$ as an initialization to find $x_{t+1}$ by Newton's method. If $\epsilon_t$ is close to $\epsilon_{t+1}$, we expect that $x_t$ is also close to $x_{t+1}$. Therefore, we have reason to hope that $x_t$ is in the local convergence region for Newton's method. In this manner we can extend the local convergence guarantee of Newton to a more global property.

## 25.2 Linear programming

After sketching the basic idea in full generality, we will now tailor the logarithmic barrier method to the linear programming (LP) problem defined as follows:

$$\text{LP}: \quad \min_x \quad c^\top x \quad \text{s.t.} \quad Ax \geqslant b \tag{4}$$

Here, $A \in \mathbb{R}^{m \times n}$ with $m \geqslant n$ and $\text{rank}(A) = n$. Denote $x^*$ as the optimal point.

First, we write out the augmented cost function by the logarithmic barrier method, i.e.,

$$f_\epsilon(x) = c^\top x - \epsilon \sum_{j=1}^{m} \ln\left(A_j^\top x - b\right). \tag{5}$$

where $A_j^\top$ is the $j$-th row of $A$. Define $x_\epsilon^* = \text{argmin}_x f_\epsilon(x)$.

**Fact 25.3.** *The optimal point $x_\epsilon^*$ exists and is unique for any $\epsilon > 0$.*

*Proof.* We can easily check that $f_\epsilon(x)$ is convex (as a sum of two convex functions). Therefore, the minimizer $x_\epsilon^*$ must exist and is unique.

To show the convexity of $f_\epsilon$, we can check the second-order derivative, which is positive definite as shown in (7) later. ■

### 25.2.1 Central path

The central path of the LP problem in 4 is depicted by the set of $\{x_\epsilon^* | \epsilon > 0\}$, as shown in Figure 2.

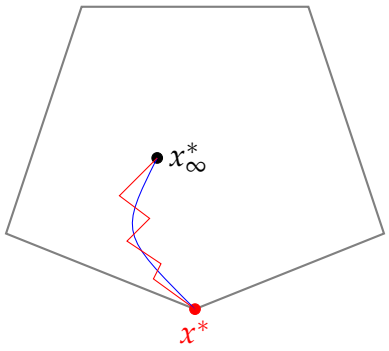


Figure 2: The central path

Our goal is to design an algorithm that will approximately follow the central path. Assume that we already have a "good enough" initial point, then at every step, we apply one step of Newton's method. To guarantee that the algorithm converges, we need to answer the following two questions:

- Under what conditions does the single-step Newton method work?
- How should we update $\epsilon$?

### 25.2.2 Newton decrement

To apply Newton's method, first we need to find out the first-order and second-order derivatives of $f_\epsilon$. Note that

$$\nabla f_\epsilon(x) = c - \epsilon \sum_{j=1}^{m} \frac{A_j}{A_j^\top x - b} \triangleq c - \epsilon A^\top S^{-1} \mathbb{1} \tag{6}$$

$$\nabla^2 f_\epsilon(x) = \epsilon A^\top S^{-2} A = \epsilon \sum_{j=1}^{m} \frac{A_j A_j^\top}{s_j^2} \tag{7}$$

where $\mathbb{1} = [1, 1, \cdots, 1]^\top \in \mathbb{R}^{m \times 1}$, and $S = \text{diag}\{s_1, \ldots, s_m\}$ is the diagonal matrix of slack quantities $s_j = A_j^\top x - b$.

Recall the Newton update

$$\bar{x} = x - [\nabla^2 f_\epsilon(x)]^{-1} \nabla f_\epsilon(x) = x - [\epsilon A^\top S^{-2} A]^{-1} \left( c - \epsilon A^\top S^{-1} \mathbb{1} \right) . \tag{8}$$

Recall that Newton's method finds the solution by making the first-order condition zero. To measure how much the Newton update will decrease the first-order approximation, we introduce the concept of Newton decrement.

Define the *Newton decrement* $q(x, \epsilon)$ as

$$q^2(x, \epsilon) = \nabla f_\epsilon(x)^\top [\nabla^2 f_\epsilon(x)]^{-1} \nabla f_\epsilon(x)\,. \tag{9}$$

Equivalently,

$$\begin{aligned} q(x, \epsilon) &= \left\| [\nabla^2 f_\epsilon(x)]^{-1/2} \nabla f_\epsilon(x) \right\|_2 \\ &= \left\| \nabla^2 f_\epsilon(x)^{-1} \nabla f_\epsilon \right\|_{\nabla^2 f_\epsilon(x)}, \end{aligned}$$

where $\|x\|_H = \sqrt{x^\top H x}$. The last identity reveals that we can think of the Newton decrement as the magnitude of the Newton step measured in the *local norm* of the Hessian.

Note that the Newton decrement also relates to the difference between $f_\epsilon(x)$ and the minimum of its second-order approximation:

$$\begin{aligned} & f_\epsilon(x) - \min_{\bar{x}} \left( f_\epsilon(x) + \nabla f_\epsilon(x)^\top (\bar{x} - x) + (\bar{x} - x)^\top \nabla^2 f_\epsilon(x)(\bar{x} - x) \right) \\ = \;& f_\epsilon(x) - \left( f_\epsilon(x) - \frac{1}{2} \nabla f_\epsilon(x)^\top [\nabla^2 f_\epsilon(x)]^{-1} \nabla f_\epsilon(x) \right) \\ = \;& \frac{1}{2} \nabla f_\epsilon(x)^\top [\nabla^2 f_\epsilon(x)]^{-1} \nabla f_\epsilon(x) \triangleq \frac{1}{2} q^2(x, \epsilon). \end{aligned} \tag{10}$$

We'll use the Newton decrement to find out the conditions for the convergence guarantee of the algorithm.

### 25.2.3 An update rule and its convergence guarantee

We'll now come up with an update rule that can guarantee convergence if some initial conditions are satisfied. To develop the update rule, we first introduce the following propositions.

**Proposition 25.4.** *Assume $Ax > b$ and $q(x, \epsilon) < 1$, then we have*

$$c^\top x - c^\top x^* \leqslant 2\epsilon n. \tag{11}$$

In particular, if we maintain that $x_t$ is interior point satisfying $Ax_t > b$, and $q(x_t, \epsilon_t) < 1$, then $c^\top x_t$ converges to $c^\top x^*$ as $\epsilon_t$ goes to 0, i.e., $x_t$ converges to global optimum. However, the condition $q(x_t, \epsilon_t) < 1$ is not trivial.

**Proposition 25.5.** *If $Ax > b$, and $q(x, \epsilon) < 1$, then the pure Newton iterate step $\bar{x}$ satisfies,*

$$q(\bar{x}, \epsilon) \leqslant q(x, \epsilon)^2 \tag{12}$$

It ensures that $q(\bar{x}, \epsilon) < 1$ given $q(x, \epsilon) < 1$ and $x$ is interior point. But we also want that $q(\bar{x}, \bar{\epsilon}) < 1$ for some $\bar{\epsilon} < \epsilon$.

**Proposition 25.6.** *Assume $q(x, \epsilon) \leqslant \frac{1}{2}$, interior point $Ax > b$, put*

$$\bar{\epsilon} = \left(1 - \frac{1}{6\sqrt{n}}\right)\epsilon, \tag{13}$$

*then we have*

$$q(\bar{x}, \bar{\epsilon}) \leqslant \frac{1}{2} \tag{14}$$

These propositions suggest the following update rule,

$$x_{t+1} = x_t - \nabla^2 f_{\epsilon_t}(x)^{-1} \nabla f_{\epsilon_t}(x_t) \tag{15}$$

$$\epsilon_t = \left(1 - \frac{1}{6\sqrt{n}}\right)\epsilon \tag{16}$$

**Theorem 25.7.** *Suppose $(x_0, \epsilon_0)$ satisfies $Ax_0 > b$ and $q(x_0, \epsilon_0) \leqslant \frac{1}{2}$, then the algorithm converges in $\mathcal{O}(\sqrt{n}\log(n/\eta))$ iterations to $\eta$ error, i.e., we have $c^\top x_t \leqslant c^\top x^* + \eta$ after $\mathcal{O}(\sqrt{n}\log(n/\eta))$ iterations.*

*Proof.* As Newton step maintains $x_{t+1}$ in the interior, by using the three propositions above, we have

$$\begin{aligned} c^\top x_t &\leqslant c^\top x^* + 2\epsilon_t n \\ &= c^\top x^* + 2\left(1 - \frac{1}{6\sqrt{n}}\right)^t \epsilon_0 \\ &\leqslant c^\top x^* + 2\exp\left(-\frac{t}{6\sqrt{n}}\right)\epsilon_0 \end{aligned} \tag{17}$$

Therefore, to have a error of $\eta$, $t \geqslant \frac{6\sqrt{n}}{\epsilon_0}\log\frac{2n}{\eta}$. We can then conclude that the algorithm converges in $\mathcal{O}(\sqrt{n}\log(n/\eta))$ iterations to $\eta$ error. ■

The algorithm stated above is the so-called short-step method. Although theoretical convergence rate is guaranteed, the combination of small decrease in $\epsilon$ and a single Newton step is slow in practice. Instead, a more practical method is the so-called long-step method, where $\epsilon$ is reduced in faster rate and several Newton steps are taken per iteration.

# References

[Ber16] D.P. Bertsekas. *Nonlinear Programming*. Athena scientific optimization and computation series. Athena Scientific, 2016.