EE C227C Convex Optimization and Approximation	Lecture 14 - 3/3/2022

Lecture 14: History of IPM and Analysis of Newton's Method

Lecturer: Jiantao Jiao *Scribe: Zhe Fu, Tianhao Wu*

In this lecture, we learned some history of Interior-point method (IPM) and the analysis of Newton's method. We showed that the analysis of Newton's method might be misleading, but it provided some intuitions for later works and it is one of the key reasons to enable IPM's success.

1 Some History

We can define a convex optimization problem as

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, 1 \leq i \leq m \\ & x \in G, G \subset \mathbb{R}^n \end{aligned} \tag{1}$$

Ellipsoid Method 1976 One of the major invention in convex optimization is the Ellipsoid method. Ellipsoid method is used for solving the generic convex optimization problems. People in 1976s show that if we have a first-order oracle $(f(x), \nabla f(x))$ and the constraint is $\|x\|_2 \leq R$, the number of queries that we need to achieve an ϵ accuracy result is:

$$O(n^2 \log(\frac{\max_{\|x\|_2 \leq R} f(x) - \min_{\|x\|_2 \leq R} f(x)}{\epsilon} + 1))$$

iterations, and each iteration's arithmetic cost is $O(n^2)$.
This is a major advance in optimization theory since the two steps are both computational efficient: the number of iterations is polynomial and for each step we only need $O(n^2)$, so overall we only need $O(n^4)$.

Leonid Khachiyan 1978 [1] Inspired by Ellipsoid method, Khachiyan used the method to show LP is polynomial time solvable. And the complexity is $O(n^4)$ as we have discussed in Ellipsoid method. To be more specific, we usually define n as the dimension of x and m as the number of constraints, the computation time is of $O(n^3(n+m))$.
Both Ellipsoid method and Khachiyan's work didn't make any impact at that time since people were focusing on simplex method.

Major breakthrough: Karmarkar 1984 [2] Karmarkar came up with an amazing algorithm that was fundamentally different from Ellipsoid method. He showed that LP can be solved in $O(n^{3.5})$, more precisely, is $O(m^{\frac{3}{2}} n^2)$.
People started to use Karmarkar's algorithm in practice to build solvers and found that it could work so much better than previous methods.

James Renegar & Clovis Gonzaga 1986 [3] [4] The two authors published two independent papers, and they both improved the computation time to $O(n^3)$. However, the major advance of their algorithm is not of the improvement of the complexity, but is that their algorithm looks much more traditional and very similar to the idea of Path Following in 1960s. Moreover, their proof is much simpler and easier to understand.
Their work excited the academia and the algorithm is an important fundamental theory for Interior-point method.

Yurri Nesterov 1987 Nesterov figured out the general approach to develop the Interior-point method (IPM) for convex programming.

Nesterov et al. 1994 [5] [6] Nesterov and Nemirovski summarized the state of art IPM and generalized it to any domain. They extended the scope of polynomial-time complexity results to a wide family of convex optimization problems.
Nesterov and Todd provides a theoretical foundation for efficient interior-point algorithms for convex programming problems expressed in conic form, when the cone and its associated barrier are self-scaled. They algorithms can take steps that go typically a large fraction of the way to the boundary of the feasible region.

2010s People started to use IPM to solve some classical combinatorial optimization problems, and it turned out that IPM was the most state of the art and fastest algorithm for them.

One of the key reasons to enable IPM's success is the so-called "correct" analysis of Newton's method.

2 Newton's method

Newton's method has existed for many centuries and it is a numerical root-finding algorithm which produces successively better approximations to the roots of a real-valued function.

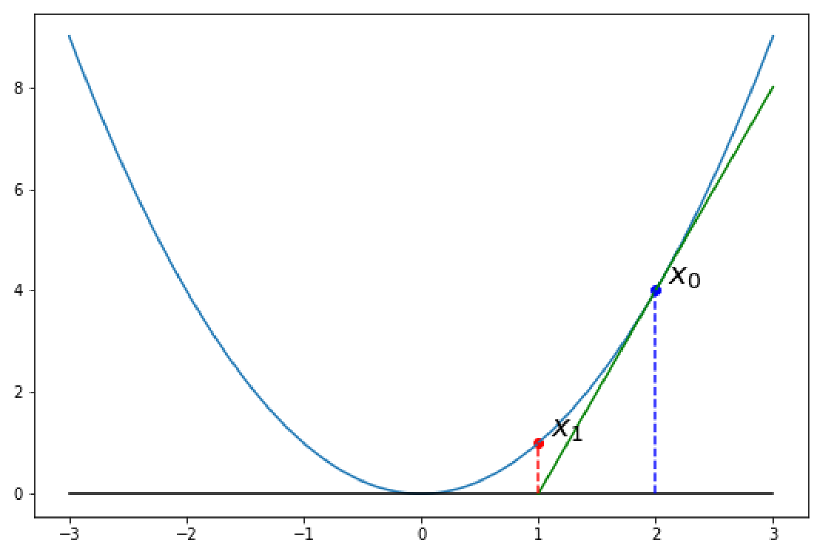

Figure 1: Example of how x_0 steps to x_1 by Newton's method.

Here we raise a specific example to show how this algorithm works: We want to find the root of $f(x)$ (shown in blue solid line in Figure 1) represents by equation

$$f(x) = g(x_0) + g^{'}(x_0)(x - x_0) = 0$$

$g(x_0)$ is the tangent line that contains point $(x_0, f(x_0))$ (shown in green solid line in Figure 1). The update rule is:

$$x_1 = x_0 - \frac{g(x_0)}{g^{'}(x_0)}$$

Example 1. Suppose we have a function $g(x) = a - \frac{1}{x}$ and its corresponding derivative $g^{'}(x) = \frac{1}{x^2}$. Then the equation to get x_{t+1} from x_t should be:

$$x_{t+1} = x_t - \frac{g(x_t)}{g^{'}(x_t)} = 2x_t - ax_t^2$$

Let $e_t = 1 - ax_t$, then we can get $e_{t+1} = e_t^2$:

$$\begin{cases} |e_0| < 1 \Rightarrow e_t \to 0 \\ |e_0| = 1 \Rightarrow e_t = 1 \\ |e_0| > 1 \Rightarrow e_t \to \infty \end{cases}$$

From this example we can see that Newton's method generally doesn't converge. This is the first issue of this method. The second issue comes from quadratic convergence. Linear convergence usually has the form like $|e_t| \leq \frac{1}{2}|e_{t-1}|$, and it requires $\log(\frac{1}{\epsilon})$ iterations to converge, while the quadratic convergence will need $\log\log(\frac{1}{\epsilon})$ iterations, which is so much faster.

3 Newton's method for minimization

Consider the case we want to minimize a function $f(x)$. Suppose we start from certain point x_0, we use a second order expansion $\tilde{f}$ to approximate f locally around x_0,

$$\tilde{f}(x) = f(x_0) + \langle \nabla f(x_0), x - x_0 \rangle + \frac{1}{2}(x - x_0)^T \nabla^2 f(x_0)(x - x_0).$$

Therefore we can set $x_1 = \arg\min_{x \in \mathbb{R}} \tilde{f}(x)$. Note that this require $\nabla^2 f(x_0) \succeq 0$. However, even if $\nabla^2 f(x_0)$ is not positive definite, we can also write down the first order optimal condition,

$$\nabla \tilde{f}(x_1) = 0,$$

which leads to the following Newton update rule,

$$x_1 = x_0 - [\nabla^2 f(x_0)]^{-1} \nabla f(x_0).$$

4 Analysis of Newton's method (old)

Following we will present the analysis of Newton's method (However, might be misleading, but provide some intuitions for later works). To start, we state the following theorem,

Theorem 2. *Suppose x^* is a local minimum of f. Assume that the function f satisfies the following two assumptions:*

1. *f have curvature around x^*, i.e., $\exists \mu > 0$ such that*

$$\nabla^2 f(x^*) \succeq \mu \mathbb{I}_n$$

2. *The Hessian matrices are Lipschitz, i.e., $\exists M < \infty$ such that for any x, y,*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_{op} \leq M \|x - y\|_2$$

We further assume that $\|x_0 - x^*\| \leq \frac{2\mu}{3M} = \bar{r}$*, then the Newton's method guarantees that for any* k*,*

$$\|x_k - x^*\|_2 \leq \bar{r},$$

and

$$\|x_{k+1} - x^*\|_2 \leq \frac{M}{2(\mu - M\|x_k - x^*\|_2)}\|x_k - x^*\|_2^2.$$

Proof For simplicity, denote $r_k = \|x_k - x^*\|_2$. We use a induction argument, suppose $r_k \leq \bar{r}$ (which is automatically satisfied when $k = 0$),

$$\begin{aligned}
x_{k+1} - x^* =& x_k - x^* - [\nabla^2 f(x_k)]^{-1}\nabla f(x_k) \\
=& x_k - x^* - [\nabla^2 f(x_k)]^{-1}\int_0^1 \nabla^2 f(x^* + \tau(x_k - x^*))(x_k - x^*)d\tau \\
=& [\nabla^2 f(x_k)]^{-1} G_k(x_k - x^*),
\end{aligned}$$

where

$$G_k = \int_0^1 [\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))]d\tau.$$

The first equality follows from the update rule, and the second equality is due to the Newton–Leibniz formula. Next, we upper bound G_k using assumption 2. Since operator norm is convex, by Jensen's inequality the following holds,

$$\begin{aligned}
\|G_k\|_{op} \leq& \int_0^1 \|\nabla^2 f(x_k) - \nabla^2 f(x^* + \tau(x_k - x^*))\|_{op} d\tau \\
\leq& \int_0^1 M(1-\tau)r_k d\tau \\
=& \frac{r_k M}{2}.
\end{aligned}$$

By assumption 1, we can lower bound $\nabla^2 f(x_k)$ by

$$\begin{aligned}
\nabla^2 f(x_k) \succeq& \nabla^2 f(x^*) - M r_k \mathbb{I}_n \\
\succeq& (\mu - M r_k)\mathbb{I}_n.
\end{aligned}$$

Therefore use the assumption that $r_k \leq \bar{r}$, we know that $\nabla^2 f(x_k)$ is positive definite, therefore, combining the two inequalities, we have

$$r_{k+1} \leq \frac{M}{2(\mu - M r_k)} r_k^2.$$

Moreover, using the fact that $r_k \leq \bar{r}$,

$$r_{k+1} \leq \frac{M}{2(\mu - M\bar{r})}\bar{r}^2 = \bar{r}.$$

Hence we finished the proof. □

5 Classical path following

We introduce the Classical path following algorithm ([7]) in this section. Consider the scenario we want to solve the following optimization problem,

$$\min_{x\in\mathcal{X}} c^T x$$

Suppose we have a penalty function $F(\cdot)$ which is convex and third order continuous differentiable, i.e., the third order differential are continuous. We further assume that the domain of F is the interior of $\mathcal{X}$, and satisfies the property that for $x \to \partial\mathcal{X}$,

$$F(x) \to \infty$$

Therefore we can turn to the new optimization problem,

$$\min_{x\in\text{int}(\mathcal{X})} F_t(x) = tc^T x + F(x)$$

We denote the minimizer as

$$x(t) = \arg\min_{x\in\text{int}(\mathcal{X})} F_t(x)$$

And we can see that when $t \to \infty$, $x(t) \to x^*$. We use the following algorithmic framework: Suppose we have an approximation of $x(t_{s-1})$ which we denote as $\tilde{x}(t_{s-1})$, then we update $t_{s-1} \to t_s$, and minimize $F_{t_s}(x)$ using Newton's method at point $\tilde{x}(t_{s-1})$ to get $\tilde{x}(t_s)$. We have the following theorem which followed the above framework and solve the case when the problem is a LP.

Theorem 3 (Renegar, Gonzaga)**.** *Suppose* $\mathcal{X} = \{x : a_i^T x \le b_i, i \in [m]\}$. *Set* $F(x) = \sum_{i=1}^m \log(\frac{1}{b_i - a_i^T x})$, $t_s = (1 + \frac{0.1}{\sqrt{m}})t_{s-1}$. *Suppose that* $F_{t_s}(x_s) - \min_x F_{t_s}(x) \le 0.1$, *the algorithm guarantees that*

$$c^T x_s - \min_{x\in\mathcal{X}} c^T x \le 2me^{\frac{-s}{10\sqrt{m}}}(c^T x_0 - \min_{x\in\mathcal{X}} c^T x).$$

The algorithm roughly says that we need to update $O(\sqrt{m})$ steps to achieve constant optimal.

References

[1] L. G. Khachiyan, "A polynomial algorithm in linear programming," in *Doklady Akademii Nauk*, vol. 244, no. 5. Russian Academy of Sciences, 1979, pp. 1093–1096.

[2] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, 1984, pp. 302–311.

[3] J. Renegar, "A polynomial-time algorithm, based on newton's method, for linear programming," *Mathematical programming*, vol. 40, no. 1, pp. 59–93, 1988.

[4] C. C. Gonzaga, "An algorithm for solving linear programming problems in o (n 3 l) operations," in *Progress in mathematical programming.* Springer, 1989, pp. 1–28.

[5] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming.* SIAM, 1994.

[6] Y. E. Nesterov and M. J. Todd, "Self-scaled barriers and interior-point methods for convex programming," *Mathematics of Operations research*, vol. 22, no. 1, pp. 1–42, 1997.

[7] A. V. Fiacco and G. P. McCormick, "The sequential unconstrained minimization technique (sumt) without parameters," *Operations Research*, vol. 15, no. 5, pp. 820–827, 1967.