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 24, 2018

23 Lecture 23: Newton's Method

Up until now, we have only considered first order methods to optimize functions. Now, we will utilize second order information to achieve a faster rate of convergence.

As always, our objective is to minimize a function $f : \mathbb{R}^d \to \mathbb{R}$. The basic idea of Newton's Method is to set the first-order Taylor expansion of the gradient to zero: $F(x) = \nabla f(x) = 0$. This leads to an iterative update step that will (under certain conditions) lead to a significantly faster convergence rate than gradient descent methods.

To illustrate the point, consider a single variable function $\varphi : \mathbb{R} \to \mathbb{R}$. Our goal is to solve the non-linear equation $\varphi(x) = 0$. From Taylor's theorem, we can express the first-order form of $\varphi(x)$ as

$$\varphi(x) = \varphi(x_0) + \varphi'(x) \cdot (x - x_0) + o(|x - x_0|)$$

given $\delta = x - x_0$ we equivalently have that

$$\varphi(x_0 + \delta) = \varphi(x_0) + \varphi'(x) \cdot \delta + o(|\delta|)$$

Disregarding the $o(|\delta|)$ term, we solve (over δ) the following objective:

$$\varphi(x_0) + \varphi'(x_0)\delta = 0$$

Then, $\delta = -\frac{\varphi(x_0)}{\varphi'(x_0)}$, leading to the iteration $x_{t+1} = x_t - \frac{\varphi(x_t)}{\varphi'(x_t)}$.

We cam similarly make an argument for a multi variable function $F : \mathbb{R}^d \to \mathbb{R}$. Our goal is to solve $F(x) = 0$. Again, from Taylor's theorem we have that

$$F(x + \Delta) = F(x) + J_F(x)\Delta + o(\|\Delta\|)$$

where J_F is the Jacobian. This gives us $\Delta = -J_F^{-1}(x)F(x)$, and the iteration

$$x_{t+1} = x_t - J_F^{-1}(x_t)F(x_t)$$

Given $f : \mathbb{R} \to \mathbb{R}$, Newton's Method applies this update to $F(x) = \nabla f(x) = 0$. It uses the update rule

$$x_{t+1} = x_t - \nabla^2 f(x_t)^{-1} \nabla f(x_t)$$

Equivalently to setting the first order Taylor expansion of the gradient to 0, a Newton step minimizes the second order Taylor approximation

$$f(x) \approx f(x_t) + \nabla f(x_t)^T (x - x_t) + \frac{1}{2}(x - x_t)^T \nabla^2 f(x_t)(x - x_t)$$

Now, we will show that Newton's Method converges to a local minimum, given a starting point that is within a neighborhood of that point.

Theorem 23.1. *Given $f : \mathbb{R}^d \to \mathbb{R}$ and assuming that*

1. *f is twice continuously differentiable*
2. *$\nabla^2 f(x)$ is Lipschitz: $\|\nabla^2 f(x) - \nabla^2 f(x')\| \leqslant \|x - x'\|$*
3. *$\exists x^*$ s.t. $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq \alpha I$ and $\|x^0 - x^*\| \leqslant \frac{\alpha}{2}$*

Then, $\|x_{t+1} - x^*\| \leqslant \frac{1}{\alpha}\|x_t - x^*\|^2$

Proof. Given that $\nabla f(x^*) = 0$, we have that

$$\begin{aligned} x_{t+1} - x^* &= x_t - x^* - \nabla^2 f(x_t)^{-1} \nabla f(x_t) \\ &= \nabla^2 f(x_t)^{-1} [\nabla^2 f(x_t)(x_t - x^*) - (\nabla f(x_t) - \nabla f(x^*))] \end{aligned}$$

This implies that

$$\|x_{t+1} - x^*\| \leqslant \underbrace{\|\nabla^2 f(x_t)^{-1}\|}_{\text{Goal: show} \leqslant \frac{2}{\alpha}} \cdot \underbrace{\|\nabla^2 f(x_t)(x_t - x^*) - (\nabla f(x_t) - \nabla f(x^*))\|}_{\text{Goal: show} \leqslant \frac{1}{2}\|x^t - x^*\|^2}$$

First, let's bound the second term $\|\nabla^2 f(x_t)(x_t - x^*) - (\nabla f(x_t) - \nabla f(x^*))\|$. Applying the integral remainder form of Taylor's theorem to $\nabla f(x_t)$ we have that

$$\nabla f(x_t) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x_t + \gamma(x^* - x_t)) \cdot (x_t - x^*) d\gamma$$

We therefore have that

$$\begin{aligned}
&\|\nabla^2 f(x_t)(x_t - x^*) - (\nabla f(x_t) - \nabla f(x^*))\| \\
&= \| \int_0^1 [\nabla^2 f(x_t) - \nabla^2 f(x_t + \gamma(x^* - x_t))](x_t - x^*) d\gamma \| \\
&\leqslant \int_0^1 \|\nabla^2 f(x_t) - \nabla^2 f(x_t + \gamma(x^* - x_t))\| \cdot \|x_t - x^*\| d\gamma \\
&\leqslant \left(\int_0^1 \gamma d\gamma \right) \|x_t - x^*\|^2 && (\nabla^2 f(x_t) \text{ is Lipschitz}) \\
&= \frac{1}{2} \|x_t - x^*\|^2
\end{aligned}$$

Now, let's bound the first term $\|\nabla^2 f(x_t)^{-1}\|$. By the Wielandt-Hoffman Theorem,

$$\begin{aligned}
|\lambda_{min}(\nabla^2 f(x_t)) - \lambda_{min}(\nabla^2 f(x^*))| &\leqslant \|\nabla^2 f(x_t) - \nabla^2 f(x^*)\| \\
&\leqslant \|x_t - x^*\| && (\nabla^2 f(x_t) \text{ is Lipschitz})
\end{aligned}$$

Thus, for $\|x_t - x^*\| \leqslant \frac{\alpha}{2}$ and given that $\nabla^2 f(x^*) \succeq \alpha I$, this implies that $\lambda_{min}(\nabla^2 f(x_t)) \geqslant \frac{\alpha}{2}$. Hence, $\|\nabla^2 f(x_t)^{-1}\| \leqslant \frac{2}{\alpha}$.

Putting the two bounds together, we have that

$$\|x_{t+1} - x^*\| \leqslant \frac{2}{\alpha} \cdot \frac{1}{2} \|x_t - x^*\|^2 = \frac{1}{\alpha} \|x_t - x^*\|^2$$

■

Note that we did not need convexity in the proof. Given that we are within a neighborhood of the local minimum x^*, then we can achieve ϵ error in just $O(\log \log \frac{1}{\epsilon})$ iterations. (this is called *quadratic convergence*.)

23.1 Damped Update

In general, Newton's method can be quite unpredictable.

Example: Consider $f(x) = \sqrt{x^2+1}$ – this looks like a smoothed version of $|x|$.Clearly $x^* = 0$. Calculating the necessary parameters as in newtons method we find

$$f'(x) = \frac{x}{\sqrt{x^2+1}}$$
$$f''(x) = (1+x^2)^{-3/2}$$

Note that $f(x)$ is strongly convex since its second derivative strictly positive and 1–smooth ($|f'(x)| < 1$. We have that the newton step for minimizing $f(x)$ is

$$\begin{aligned} x_{t+1} &= x_t - \frac{f'(x_t)}{f''(x)} \\ &= -x_t^3 \end{aligned}$$

The behavior of this algorithm depends on the magnitude of x_t. In particular we have the following three regimes

$$\begin{cases} |x_t| < 1 & \text{Algorithm converges } \textit{cubically} \\ |x_t| = 1 & \text{Algorithm oscillates between } -1 \text{ and } 1 \\ |x_t| > 1 & \text{Algorithm diverges} \end{cases}$$

Note the similarity between underdamped, critically damped, and overdamped dynamical systems. This example shows that even for strongly convex functions with Lipschitz gradients that Newton's method is only guaranteed to converge locally. To avoid divergence, a popular technique is to use a *damped* step–size:

$$x_{t+1} = x_t - \eta_t \nabla^2 f(x_t)^{-1} \nabla f(x_t)$$

η_t can be chosed by backtracking line search. Usually though $\eta = 1$ is a good first choice since, if you are in a region of convergence, you are guaranteed quadratic convergence.

23.2 Quasi–Newton Methods

Comparing Gradient Descent and Newton's method side by side:

1. Gradient Descent

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

2. Newton's Method

$$x_{t+1} = x_t - \nabla^2 f(x_t)^{-1} \nabla f(x_t)$$

we see that gradient descent approximates $\nabla^2 f(x_t)^1$ as a scaled version of the identity. That is, gradient descent is equivalent to Newton's method when $\nabla^2 f(x_t)^{-1} = \eta_t I$ where I is the identity matrix. Quasi–newton methods take the analogy a step further by approximating the Hessian by some other matrix. The idea in doing so is to avoid an expensive matrix inversion at each step. What we want is an approximation

$$\widehat{f}_{B_t}(x) \approx f(x_t) + \nabla f(x_t)^\top (x - x_t) + \frac{1}{2}(x - x_t) B_t^{-1} (x - x_t)$$

such that

1. $\nabla \widehat{f}_{B_t}(x_t) = \nabla f(x_t)$.
   It seems reasonable that our approximation should be the same up to first order.
2. $\nabla \widehat{f}_{B_t}(x_{t-1}) = \nabla f(x_{t-1})$
   This condition states that the gradient should still be correct at the previous iterate.

If the two last gradients are correct, we can expect our Hessian approximation to be reasonable along the direction $x_t - x_{t-1}$. This is called a *Secant Approximation* which can be written as

$$\nabla \widehat{f}_{B_t}(x_{t+1}) = \nabla f(x_t) - B_t^{-1}(x_{t+1} - x_t)$$

If we let

$$\begin{aligned} s_t &= x_{t+1} - x_t \\ y_t &= \nabla \widehat{f}_{B_t}(x_{t+1}) - \nabla f(x_t) \end{aligned}$$

Then we arrive at the *Secant Equation*

$$s_t = B_t y_t$$

There could be multiple B_t that satisfy this condition. We can enforce other constraints to help narrow down on a particular choice. Some popular requirements are requiring B_t to be positive definite, making sure B_t is as close to B_{t-1} as possible for some appropriate metric, or requiring B_t to be a low–rank update of previous iterates where the update can be done via the Sherman–Morrison formula. One of the most successful implementations of this is called BFGS named after Broyden, Fletcher, Goldfarb, Shanno and its limited–memory counterpart, L–BFGS.

References