EE C227C Convex Optimization and Approximation Lecture 21 - 04/10/2022

Lecture 21: Second Order Methods for General Function

Lecturer: Jiantao Jiao *Scribe: Justin Kang, Weijia Zeng*

1 Motivation

[1] We have already discussed second order methods in the context of the optimization of convex functions. These include some of the very oldest methods for convex optimization, namely, Newton methods. In contrast, the study of second order methods for general optimization remains an active area of research with many open problems. One advantage of second order methods is that very few assumptions are needed to formulate them (just the existence of the Hessian), as opposed to interior point methods which require many more assumptions. For general functions, we would say an algorithm has a good output if:

- For convex functions, we approach a global minimizer.
- For non-convex functions, we approach a local minimizer if one exists.

We begin our study by reviewing the deficiency of the canonical Newton's Method. Our goal is to solve the following unconstrained problem:

$$\min_{x\in\mathbb{R}^n} f(x). \tag{1}$$

The standard Newton's Method does the following:

$$x_{k+1} = x_k - \left[\nabla^2 f\left(x_k\right)\right]^{-1} \nabla f(x_k). \tag{2}$$

As previously noted, one issue with the above algorithm is that an poor initialization can cause it to diverge. Furthermore, without the assumption of (strong) convexity, it is possible that the Hessian is degenerate. In what follows, we discuss four methods that have been proposed in the literature try to solve these problems, however, only the final one provides us with"good" global convergence guarantees.

2 Levenberg-Marqurdt Regularization

The first approach is Levenberg-Marqurdt Regularization. It can be viewed as a heuristic for dealing with a degenerate Hessian matrix via regularization. The update rule is as follows:

$$G_k = \nabla^2 f(x_k) + \gamma I_n \tag{3}$$

$$x_{k+1} = x_k - G_k^{-1}\nabla f(x_k), \tag{4}$$

where γ can be chosen to ensure that $G_k > 0$. This algorithm can also be viewed as a combination of Newton's Method and Gradient Descent. When $\gamma = 0$, we recover standard Newton's Method, and when $\nabla^2 f\left(x_k\right) = 0$, the algorithm is equivalent to gradient descent.

This method is widely implemented in software packages such as matlab, but does not have global convergence guarantees.

3 Line Search

For the Line Search algorithm, we consider the intuition we developed previously about Newton's Method being too aggressive. Thus, in this algorithm, we consider doing a line search in the descent direction $-\left[\nabla^2 f\left(x_k\right)\right]^{-1} \nabla f(x_k)$. Formally, we write the update as:

$$x_{k+1} = x_k - h_k \left[\nabla^2 f\left(x_k\right)\right]^{-1} \nabla f(x_k), \tag{5}$$

where h_k is chosen to generate a monotone sequence of function values $f(x_{k+1}) < f(x_k)$, making this method a true descent method. This method is somewhat analogous to damped Newton's Method. Note that this method does not address the potential degeneracy issues of the hessian. In some cases, this algorithm may perform well, but in general, there are no convergence guarantees.

4 Trust Region Method

This method is another heuristic that has been used a lot in deep learning. For example the trust region policy optimization in deep reinforcement learning is an instance of this method.

Newton's Method is based on a local quadratic expansion of the function, which becomes inaccurate as we get further away from x_k, thus one heuristic is to find the point that minimizes this quadratic expansion in some norm-ball $\Delta(x_k) = \left\{x \mid \|x - x_k\|^2 \leq \epsilon\right\}$. Formally we write the update as:

$$x_{k+1} = \underset{x \in \Delta(x_k)}{\arg\min} \left[\langle \nabla f(x_k), x - x_k\rangle + \frac{1}{2}\left\langle \nabla^2 f(x_k)(x - x_k), x - x_k\right\rangle\right] \tag{6}$$

Similarly to Line Search, this algorithm can work in practice in many cases, but there are no global convergence guarantees.

5 Cubic Regularization

Cubic Regularization is the final method for second order general function optimization that we will discuss, and is the only approach with theoretical guarantees. We begin by looking at the analogy with how gradient methods are derived. First we find a global upper bound for our function based on a local linear expansion and a Lipshitz assumption:

$$f(x) < f(x_k) + \langle \nabla f(x_k), x - x_k\rangle + \frac{L}{2}\|x - x_k\|^2. \tag{7}$$

Then, we define our update by considering a point gives us a strong lower bound.

In analogy to the above case, we make an assumption that the Hessian is Lipschitz continuous:

$$\left\|\nabla^2 f(x) \nabla^2 f(y)\right\|^2 \leq L\|x - y\|^2, \tag{8}$$

where the norm on the left hand side is the operator norm, and the norm on the right hand side is the Euclidean norm. Given the above assumption, we can derive the following cubic global upper bound:

Lemma 1. *(Lemma 4.1.1 in [2])*

$$f(y) - f(x) + \langle \nabla f(x), y - x\rangle - \frac{1}{2}\left\langle \nabla^2 f(x)(y - x), y - x\right\rangle \leq \frac{L}{6}\|y - x\|^3. \tag{9}$$

Notice here for general function approximation, we don't assume that f is convex.

Naturally, the next approach is to choose y such that we have a strong global bound. Note that even if we only have an upper bound $M > L$, [2] we can augment this approach. In general, we wish to solve the following problem:

$$T_M(x) = \arg\min_y \left[\langle \nabla f(x), y - x \rangle + \frac{1}{2} \langle \nabla^2 f(x)(y-x), y-x \rangle + \frac{M}{6} \|y - x\|^3 \right]. \tag{10}$$

We drop the term f(x) because the right hand side of equation (10) is a function of y. Note that the above problem is always well-posed, though not in general convex. For example, it is possible that $\nabla^2 f(x)$ is not PSD, however, since the third term grows cubically, eventually it will dominate, and thus there is no direction in which the objective decreases to $-\infty$. Thus, assuming L is known, we can update cubic regularization as follows:

$$x_0 \in \mathbb{R}_n \tag{11}$$

$$x_{k+1} = T_L(x_k). \tag{12}$$

Thus, we can see that this procedure boils down to efficiently computing $T_L(x)$. We will return to this shortly, after a discussion of error metrics.

5.1 Error Metrics

In general, our goal is to find a point x^* that satisfies two conditions:

1. $\nabla f(x^*) = 0$
2. $\nabla^2 f(x^*) \geq 0$

The error metric, which encompasses both of these conditions, and is used in this analysis is:

$$\mu_M(x) = \max \left\{ \sqrt{\frac{2}{L+M} \|\nabla f(x)\|}, -\frac{2}{2L+M} \lambda_{min} \left(\nabla^2 f(x) \right) \right\}. \tag{13}$$

We can see that the first term in the maximization as the first condition is satisfied, while the second term becomes small as the second condition is satisfied. As expected. the error metric is always positive. Note that the above metric cannot be computed exactly if L is not exactly known. If we use the update, we can bound the error $\mu_L(x)$.

Theorem 2. *(theorem 4.1.1 in [2])*
Assume $f(x) \geqslant f^*$ for all x, Then,

$$\min_{i \leq i \leq k} \mu_L(x_i) \leq \frac{8}{3} \left(\frac{3(f(x_0) - f^*)}{2kL} \right)^{\frac{1}{3}} \tag{14}$$

Since we known L and $f(x_0)$, f^* are all constants, this theorem states that the minimum of the error metric for $x_1, ..., x_k$ is decreased in order $k^{\frac{1}{3}}$, which implies the $\nabla f(x)$ decreases in order $k^{\frac{1}{6}}$. The most important takeaway of this theorem is that it gives a global convergence guarantee.

6 Lagrangian Duality

Lagrangian duality was a very important part of the development of optimization theory in the pre-computer period. It solves constrained optimization problems by including the constraints in the objective. In contrast, modern interior point methods put complexity in the constraints and leave only a linear objective.

We begin with the following problem:

$$\min_{x \in Q} f_0(x), \quad Q \text{ closed}, \tag{15}$$

$$\text{s.t. } f_j(x) \leq 0 \;\; j \in [m]. \tag{16}$$

Lagrange's idea of solving the problem comes from the minimax principle.

Theorem 3. *(Minimax Principle: Theorem 1.3.1 in [3])*
Let the function $F(x, \lambda)$ be defined for $x \in Q_1 \mathbb{R}_n$ and $\lambda \in Q_2 \subseteq \mathbb{R}_m$, where both Q_1 and Q_2 are nonempty. Then,

$$\sup_{\lambda \in Q_2} \inf_{x \in Q_1} F(X, \lambda) \leq \inf_{x \in Q_1} \sup_{\lambda \in Q_2} F(x, \lambda) \tag{17}$$

This result is also know as weak duality. Notice that we don't make any assumptions on f(x), Q_1, and Q_2; the weak duality always holds.
We can now construct the Lagrangian:

$$\mathcal{L}\{x, \lambda\} = f_0(x) + \langle \lambda, f(x) \rangle, \tag{18}$$

where $f(x) = [f_1(x) \; f_2(x) \; \cdots f_m(x)]^T$ and $\lambda \in \mathbb{R}_+^m = \{\lambda \in \mathbb{R} \mid \lambda \geq 0\}$. Then we can write the the solution to (15) as:

$$f^* = \inf_{x \in Q} \{f_0(x) \mid f_j(x) \leq 0 \;\; j \in [m]\} = \inf_{x \in Q} \left\{ \sup_{\lambda \in \mathbb{R}_+^m} \mathcal{L}\{x, \lambda\} \right\}. \tag{19}$$

The dual of this function can be written as:

$$\Psi(\lambda) = \inf_{x \in Q} \mathcal{L}\{x, \lambda\}. \tag{20}$$

Note that when we consider the domain of Ψ, because we will eventually maximize this function, we can throw away points λ where the function is $-\infty$, so we write:

$$\text{dom}(\Psi) = \{\lambda \in \mathbb{R}^m \mid \Psi(\lambda) - \infty\}. \tag{21}$$

$$X^*(\lambda) \triangleq \arg\min_{x \in Q} \mathcal{L}(x, \lambda) \tag{22}$$

$$\sup_{\lambda} \left\{ \Psi(\lambda) \mid \lambda \in \mathbb{R}_+^m, \lambda \in \text{dom}(\Psi) \right\} \leq \inf_{x \in Q_1} \sup_{\lambda \in R_+^m} \mathcal{L}\{x, \lambda\} \tag{23}$$

Equation (23) is called Lagrangian weak duality in optimization. The left hand side of (23) is the largest lower bound of the optimal value f^* in (19), which is usually harder to compute than an upper bound of f^*. When the equality in (23) holds, it's the so called strong duality. Unfortunately, strong duality never holds in non-convex optimization. Slater's condition is a special case where strong duality holds.

Theorem 4. *(Slater's Condition)*
$f_0(x)$ and $f_j(x) j \in [m]$ are all convex functions, and there exists a strictly feasible point $\bar{x}$ such that $f_j(\bar{x}) < 0 \;\; j \in [m]$, then strong duality holds.

For linear constraints, strict feasibility is not required for strong duality. Note that Slater's Condition applies only to convex programs, which is not suitable for enabling us to solve for $T_M(x)$ efficiently. Fortunately, we can show that this problem satisfies another sufficient condition for strong duality.

Theorem 5. *(Theorem 1.3.2 in [3])*
Let λ_* be a solution of $\sup_\lambda \left\{ \Psi(\lambda) \mid \lambda \in \mathbb{R}_+^m, \lambda \in \text{dom}(\Psi) \right\}$. Assume for some $\epsilon > 0$ we have:

$$\Delta_\epsilon^+(\lambda_*) \triangleq \left\{ \lambda \in \mathbb{R}_+^m \mid \|\lambda - \lambda_*\| \leq \epsilon \right\}. \tag{24}$$

Let $X(\lambda) \in X^*(\lambda), \lambda \neq \lambda_*$ be uniquely defined and the limit exists. Then

$$X^* = \lim_{\lambda \to \lambda_*} X(\lambda), \;\; \lambda \in \Delta_\epsilon^+(\lambda_*) \tag{25}$$

If $X^* \in X^*(\lambda_*)$ then strong duality holds. We can use this to show that cubic regularization can be efficiently solved.

References

[1] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, Mar. 1964. [Online]. Available: http://dx.doi.org/10.1214/aoms/1177703732

[2] Y. Nesterov, *Second-Order Methods*. Cham: Springer International Publishing, 2018, pp. 241–322. [Online]. Available: https://doi.org/10.1007/978-3-319-91578-4_4

[3] ——, *Nonlinear Optimization*. Cham: Springer International Publishing, 2018, pp. 3–58. [Online]. Available: https://doi.org/10.1007/978-3-319-91578-4_1