ECE 490: Introduction to Optimization **Spring 2022**

Lecture 28

Advanced Topics and Class Review

Lecturer: Bin Hu, Date:05/03/2022

In this lecture, we briefly discuss two advanced topics (acceleration and ADMM), and then give a review for the contents covered in this course.

28.1 Acceleration

Suppose that we consider an unconstrained optimization problem where the objective function f is m-strongly convex and L-smooth. Denote the condition number $\kappa := \frac{L}{m}$. We know that we can use the gradient descent method to achieve an iteration complexity $O\left(\kappa \log(\frac{1}{\varepsilon})\right)$ (recall that the convergence rate is $1 - \frac{1}{\kappa}$). This means that the required iteration T for the gradient descent method to achieve ε-optimality scales with the condition number κ. For larger κ, more iterations are required. This is consistent with our intuition since larger κ means the problem is ill-conditioned and more difficult to solve.

There are more sophisticated algorithms which can decrease the iteration complexity from $O\left(\kappa \log(\frac{1}{\varepsilon})\right)$ to $O\left(\sqrt{\kappa} \log(\frac{1}{\varepsilon})\right)$. This improvement is significant. Just consider $\kappa = 10000$. Then $\sqrt{\kappa} = 100$. This states that Nesterov's method is roughly 100 times faster than the gradient method in this case. Now we briefly discuss how to achieve such acceleration via adding momentum terms.

Momentum methods use the gradient information and the one-step memory x_{k-1}. One such example is the Heavy-ball method that iterates as

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}) \tag{28.1}$$

The extra term $\beta(x_k - x_{k-1})$ is the so-called "momentum term." One needs to choose the stepsize α and the momentum β, and also initialize the method at x_0 and x_{-1}. Then based on this iteration, one can compute x_1, x_2, ...

With well-chosen α and β, the Heavy-ball method achieves faster convergence rate than the gradient method for a positive definite quadratic minimization problem. However, the same choice of α and β may not work for other smooth strongly-convex functions. On the other hand, Nesterov's method is proved to have an improved iteration complexity $O(\sqrt{\kappa} \log(\frac{1}{\varepsilon}))$ for all the functions being L-smooth and m-strongly convex.

Nesterov's accelerated method has the form

$$y_k = x_k + \beta(x_k - x_{k-1}) \tag{28.2}$$

$$x_{k+1} = y_k - \alpha \nabla f(y_k) \tag{28.3}$$

We can simply rewrite Nesterov's method as

$$x_{k+1} = x_k - \alpha \nabla f((1+\beta)x_k - \beta x_{k-1}) + \beta(x_k - x_{k-1}) \tag{28.4}$$

This looks very similar to Heavy-ball method. The difference is that Nesterov's accelerated method uses a gradient evaluated at $(1+\beta)x_k - \beta x_{k-1}$ while Heavy-ball method uses a gradient evaluated at x_k.

It is worth mentioning that both Heavy-ball method and Nesterov's method only use the first-order derivative (gradient) and do not require evaluating the second-order derivative (Hessian). Hence they belong to "first-order optimization methods." The convergence rate proofs for Nesterov's method are tricky. There are several ways to do this (e.g. estimated sequence, Lyapunov functions, ODE methods). We will skip these theoretical developments.

28.2 ADMM

The key idea of the alternating direction method of multipliers (ADMM) is to accelerate the optimization process by breaking convex constrained problems into smaller pieces. Consider the following problem

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By = c \end{aligned} \tag{28.5}$$

We formulate the augmented Lagrangian:

$$L_\rho(x, y, \lambda) = f(x) + g(y) + \lambda^\mathsf{T}(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|^2$$

ADMM alternates the minimization over x and y. Specifically, ADMM iterates as

$$x_{k+1} = \arg\min_x L_\rho(x, y_k, \lambda_k) \tag{28.6}$$

$$y_{k+1} = \arg\min_y L_\rho(x_{k+1}, y, \lambda_k) \tag{28.7}$$

$$\lambda_{k+1} = \lambda_k + \rho(Ax_{k+1} + By_{k+1} - c) \tag{28.8}$$

If we apply the method of multipliers to (28.5), the following iteration is adopted:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} &= \arg\min_{x,y} L_\rho(x, y, \lambda_k) \\ \lambda_{k+1} &= \lambda_k + \rho(Ax_{k+1} + By_{k+1} - c) \end{aligned}$$

Compared with the method of multipliers, the main advantage of ADMM is that sometimes the computation of $\arg\min_x L_\rho(x, y_k, \lambda_k)$ and $\arg\min_y L_\rho(x_{k+1}, y, \lambda_k)$ can still be parallelized even when the computation of $\arg\min_{x,y} L_\rho(x, y, \lambda_k)$ cannot be parallelized.

28.3 Final Review

We talked about several main types of problems in this course:

- Unconstrained optimization:

$$\min_{x\in\mathbb{R}^p} \; f(x) \tag{28.9}$$

- Optimization with equality constraints:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0, \; i = 1, \ldots, m \end{aligned} \tag{28.10}$$

- General constrained optimization

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0, \; i = 1, \ldots, m \\ & g_j(x) \leq 0, \; j = 1, \ldots, l \end{aligned} \tag{28.11}$$

We have talked about optimality conditions and also related algorithms for all the above problems. We give a brief summary as below. Our summary is brief, and does not mention everything. To prepare the final exam, it is recommended to go over the lecture notes carefully.

1. **Optimality conditions**: For(28.9), the necessary condition for a local min is simple, i.e. $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*) \succeq 0$. We have also talked about the sufficient condition $\nabla^2 f(x^*) \succ 0$. Similarly, we talked the optimality conditions (basically the Lagrange theorem) for (28.10). For (28.11), the optimality conditions are the so-called KKT conditions. Sometimes we also use the general sufficiency conditions. Many times we need to use the Weierstrass theorem (and the corollary) to prove the existence of global min. It is important to know how to apply these conditions.

2. **Convexity:** Convexity plays an important role in optimization. What is a convex set? What is a convex function? How can we prove a function is convex? What are the optimality conditions for convex optimization problems? All these are important.

3. **Algorithms for unconstrained optimization**: For (28.9), we have introduced various algorithms including the gradient descent method, Newton's method, and the subgradient method. It is important to know the update formulas for all these methods. If f is L-smooth and m-strongly convex, we know the gradient descent method achieves linear convergence to global min. If f is only convex and smooth, then the gradient method converges to global minimum at a rate $O(1/k)$, which is sublinear. If f is only smooth, then the gradient method converges to a stationary point at the rate

$O(1/k)$. If f is only convex but not smooth, then the subgradient method converges to the global minimum at the rate $O(\log(k)/\sqrt{k})$. It is important to know the difference between sublinear convergence, linear convergence, and superlinear convergence. In addition, the stepsize rule is also important. How to calculate gradient for neural network problems is another related topic.

4. **Projection operator:** If the feasible set is convex and closed, the projected gradient method can be applied. It is important to understand the concept of projection, the related optimality conditions, and the convergence proofs.

5. **Methods for constrained optimization**: We have talked about the barrier method and the penalty method. Their convergence properties have also been discussed.

6. **Duality theory:** It is important to understand how to formulate dual problems for constrained optimization problems. Such dual problems may lead to important algorithm developments (e.g. method of multipliers).

7. **Subgradient:** When the functions are not differentiable, we need to use subgradient. We have talked about how to characterize the subgradient and several convergence results for the subgradient method.