ECE 490: Introduction to Optimization **Fall 2018**

Lecture 10

Unconstrained Optimization of Smooth Convex Functions, Part II

Lecturer: Bin Hu, Date:10/02/2018

In the last lecture, we started to talk about optimization of smooth strongly-convex functions. When the objective function is smooth and only convex (not strongly-convex), the convergence rate of the gradient method becomes $O(1/k)$. In this lecture, we will talk about Nesterov's method that achieves an accelerated rate $O(1/k^2)$ for smooth convex objective functions.

10.1 Nesterov's Method for Convex Functions

The gradient method achieves the rate $O(1/k)$ by adopting a constant stepsize. However, Nesterov's method relies on a time-varying momentum to achieve the rate $O(1/k^2)$. Specifically, given L-smooth convex f, Nesterov's method iterates as follows:

$$x_{k+1} = x_k - \alpha \nabla f((1+\beta_k)x_k - \beta x_{k-1}) + \beta_k(x_k - x_{k-1}) \tag{10.1}$$

where $\alpha = \frac{1}{L}$ and β_k is a prescribed sequence. One typical choice is setting $\beta_k = \frac{k-1}{k+2}$. Another popular choice is defining β_k recursively as follows.

$$\zeta_{-1} = 0, \quad \zeta_{k+1} = \frac{1+\sqrt{1+4\zeta_k^2}}{2}, \quad \beta_k = \frac{\zeta_{k-1}-1}{\zeta_k}.$$

The sequence $\{\zeta_k\}$ satisfies $\zeta_k^2 - \zeta_k = \zeta_{k-1}^2$. Due to the time-varying nature of β_k, we need to use the following time-varying model for (10.1):

$$\begin{aligned} \xi_{k+1} &= A_k \xi_k + B_k u_k \\ v_k &= C_k \xi_k \\ u_k &= \nabla f(v_k) \end{aligned} \tag{10.2}$$

We choose $A_k = \begin{bmatrix} (1+\beta_k)I & -\beta_k I \\ I & 0 \end{bmatrix}$, $B_k = \begin{bmatrix} -\alpha I \\ 0 \end{bmatrix}$, $C_k = \begin{bmatrix} (1+\beta_k)I & -\beta_k I \end{bmatrix}$, and $\xi_k = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}$. Then $v_k = C_k\xi_k = \begin{bmatrix} (1+\beta_k)I & -\beta_k I \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix} = (1+\beta_k)x_k - \beta_k x_{k-1}$, and $u_k = \nabla f(v_k) = \nabla f((1+\beta_k)x_k - \beta_k x_{k-1})$. We can see B_k actually does not depend on k. But if we let α depend on k, then we need B to depend on k. So (10.2) is general. We will modify the dissipation inequality to provide a sublinear rate analysis for (10.2).

10.2 How to prove the rate $O(1/k^2)$?

We can still use the dissipation inequality technique to prove such a rate. The dissipation inequality appeared in the previous lectures has the following form:

$$V(\xi_{k+1}) - \rho^2 V(\xi_k) \leq S(\xi_k, u_k)$$

We can use the dissipation inequality to prove various results:

1. If $S(\xi_k, u_k) \leq 0$, then the dissipation inequality becomes $V(\xi_{k+1}) - \rho^2 V(\xi_k) \leq 0$. This is a linear convergence in V. We have used this type of arguments to show the linear convergence of the gradient method.

2. If $S(\xi_k, u_k) \leq -(f(x_{k+1}) - f(x^*)) + \rho^2(f(x_k) - f(x^*))$, we have $V(\xi_{k+1}) + f(x_{k+1}) - f(x^*) \leq \rho^2(V(\xi_k) + f(x_k) - f(x^*))$. This is a linear convergence in $V(\xi_k) + f(x_k) - f(x^*)$. We have shown the linear convergence of Nesterov's method via this type of arguments.

3. Sometimes the algorithms are stochastic. Then the supply rate condition also has to take the randomness into accounts. If $\mathbb{E}S(\xi_k, u_k) \leq M$, then the dissipation inequality leads to a bound in the form of $\mathbb{E}V(\xi_k) \leq \rho^{2k}\mathbb{E}V(\xi_0) + \frac{M}{1-\rho^2}$. This means the algorithm goes to a small ball around the optimal solution at a linear rate. We have used this argument to show the behaviors of the stochastic gradient method with a constant stepsize.

4. If $S(\xi_k, u_k) \leq -(f(x_{k+1}) - f(x^*)) + \rho^2(f(x_k) - f(x^*))$ and $V = 0$, then the dissipation inequality leads to $f(x_{k+1}) - f(x^*) \leq \rho^2(f(x_k) - f(x^*))$. This is a linear convergence in $f(x_k) - f(x^*)$. The linear convergence of the gradient method can also be proved using such an argument.

5. If $S(\xi_k, u_k) \leq f(x^*) - f(x_k)$ and $\rho^2 = 1$, then the dissipation inequality leads to the inequality $V(\xi_{k+1}) - V(\xi_k) + f(x_k) - f(x^*) \leq 0$. Summing this inequality leads to $\sum_{t=0}^{k}(f(x_t) - f(x^*)) \leq V(\xi_0) - V(\xi_{k+1})$. We have used this argument to show that the gradient method is guaranteed to converge at the sublinear rate $O(1/k)$ when the objective function is smooth and convex.

For all these above cases, we construct the dissipation inequality by finding a P matrix satisfying the following condition:

$$\begin{bmatrix} A^\mathsf{T}PA - \rho^2 P & A^\mathsf{T}PB \\ B^\mathsf{T}PA & B^\mathsf{T}PB \end{bmatrix} - X \leq 0. \tag{10.3}$$

Now for the general model (10.2), (A, B, C) depend on k. Therefore, we need to modify the above condition as

$$\begin{bmatrix} A_k^\mathsf{T}P_{k+1}A_k - P_k & A_k^\mathsf{T}P_{k+1}B_k \\ B_k^\mathsf{T}P_{k+1}A_k & B_k^\mathsf{T}P_{k+1}B_k \end{bmatrix} - X_k \leq 0. \tag{10.4}$$

The key question is what X_k we should use. We can use the same arguments for Problem 1 in HW2 to show

$$\begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(y_k) \end{bmatrix}^{\mathsf{T}} M_k \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(y_k) \end{bmatrix} \leq f(x_k) - f(x_{k+1})$$
$$\begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(y_k) \end{bmatrix}^{\mathsf{T}} N_k \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \\ \nabla f(y_k) \end{bmatrix} \leq f(x^*) - f(x_{k+1})$$

where N_k and M_k directly depend on β_k. Now we define $f^* = f(x^*)$. If we choose $X_k := \mu_k M_k + (\mu_{k+1} - \mu_k) N_k$ for all k. then the supply rate S satisfies the condition

$$S(\xi_k, w_k) = \begin{bmatrix} \xi_k \\ w_k \end{bmatrix}^{\mathsf{T}} X_k \begin{bmatrix} \xi_k \\ w_k \end{bmatrix} \leq \mu_k (f(x_k) - f^*) - \mu_{k+1}(f(x_{k+1}) - f^*).$$

which can be used to show the rate $O(1/k^2)$ if μ_k is chosen properly. Specifically, if we can find positive semidefinite P_k and non-negative increasing sequence $\{\mu_k\}$ such that

$$\begin{bmatrix} A_k^{\mathsf{T}} P_{k+1} A_k - P_k & A_k^{\mathsf{T}} P_{k+1} B_k \\ B_k^{\mathsf{T}} P_{k+1} A_k & B_k^{\mathsf{T}} P_{k+1} B_k \end{bmatrix} - \mu_k M_k - (\mu_{k+1} - \mu_k) N_k \leq 0. \tag{10.5}$$

then we will be able to use our standard dissipation inequality arguments to show

$$\begin{aligned}(\xi_{k+1} - \xi^*)^{\mathsf{T}} P_{k+1} (\xi_{k+1} - \xi^*) - (\xi_k - \xi^*)^{\mathsf{T}} P_k (\xi_k - \xi^*) &\leq \begin{bmatrix} \xi_k \\ w_k \end{bmatrix}^{\mathsf{T}} X_k \begin{bmatrix} \xi_k \\ w_k \end{bmatrix} \\ &\leq \mu_k (f(x_k) - f^*) - \mu_{k+1}(f(x_{k+1}) - f^*).\end{aligned}$$

This is equivalent to

$$(\xi_{k+1} - \xi^*)^{\mathsf{T}} P_{k+1} (\xi_{k+1} - \xi^*) + \mu_{k+1}(f(x_{k+1}) - f^*) \leq (\xi_k - \xi^*)^{\mathsf{T}} P_k (\xi_k - \xi^*) + \mu_k (f(x_k) - f^*).$$

We can iterate the above inequality to show

$$\mu_k (f(x_k) - f^*). \leq (\xi_k - \xi^*)^{\mathsf{T}} P_k (\xi_k - \xi^*) + \mu_k (f(x_k) - f^*) \leq (\xi_0 - \xi^*)^{\mathsf{T}} P_0 (\xi_0 - \xi^*) + \mu_0 (f(x_0) - f^*).$$

If $\frac{1}{\mu_k} = O(1/k^2)$, then we immediately obtain the rate $O(1/k^2)$ for Nesterov's method.

Clearly, obtaining a rate $O(1/k^2)$ is more subtle than obtaining other rates due to the fact that now we need to find a sequence of P_k and μ_k. A specific choice for Neseterov's method is setting $\mu_k := (\zeta_{k-1})^2$ and $P_k := \frac{L}{2} \begin{bmatrix} \zeta_{k-1} \\ 1 - \zeta_{k-1} \end{bmatrix} \begin{bmatrix} \zeta_{k-1} & 1 - \zeta_{k-1} \end{bmatrix}$. We will not talk about $O(1/k^2)$ rate in future lectures.

10.3 Possible Generalizations

We have covered how to apply the dissipation inequality to obtain bounds in various forms. Many more algorithms (ADMM, stochastic variance reduction, distributed optimization methods, proximal algorithms, coordinate descent, etc) can be analyzed using the same framework. We will cover some of these variants in later lectures, but our main focus will be slowly shifted to other aspects of optimization. The iteration complexity theory only gives high level guidelines for choosing algorithms at the beginning. If the goal is just solving one particular instance of an optimization method, the iteration complexity theory may or may not be that useful. Nevertheless, if the goal is to develop algorithms with provable guarantees, the iteration complexity theory is much relevant. Finally, let's discuss some possibilities of tailoring dissipation inequality for your own research.

1. The most straightforward extension is to rewrite a new optimization method (that you are interested in) as our general optimization model with some (A, B, C) and applying our framework to analyze it. Here, once (A, B, C) are found, it is possible to follow our routine to obtain some rate guarantees.

2. A more challenging task is to figure out X when looking at a new class of objective functions. When f is not convex, the construction of X can be tricky and case-dependent. Depending on the particular problems you are working on, you may have to develop X by yourself. Some cutting edge research just focuses on developing such X for neural networks. The good thing is that once you have developed X for a particular class of f, other people can immediately use it and this X can be applied to analyze many algorithms with different (A, B, C).