Math 484: Nonlinear Programming[1] Mikhail Lavrov

Chapter 3, Lecture 5: Using Descent Methods

April 24, 2019 *University of Illinois at Urbana-Champaign*

1 Picking step sizes

Last time, we discussed descent methods: methods with iterative step

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t_k \mathbf{p}^{(k)}$$

where $\mathbf{p}^{(k)}$ is a direction, and t_k is a step size. Both of these have to be picked somehow at every step.

We had several criteria for a good descent method. Phrased in terms of $\phi_k(t) = f(\mathbf{x}^{(k)} + t\mathbf{p}^{(k)})$, they say:

1. $\phi_k(t_k) < \phi_k(0)$.
2. $\phi_k'(0) < 0$.
3. $\phi_k'(t_k) \geq \beta \cdot \phi_k'(0)$.
4. $\phi_k(t_k) \leq \phi_k(0) + (\alpha \cdot \phi_k'(0))t_k$.

Wolfe's theorem, which we proved in the previous lecture, says that:

Theorem 1.1 (Wolfe)**.** *Let α, β be real numbers satisfying $0 < \alpha < \beta < 1$. Assume $f : \mathbb{R}^n \to \mathbb{R}$ has continuous ∇f and is bounded from below.*

Starting from a point $\mathbf{x}^{(k)}$, whenever $\nabla f(\mathbf{x}^{(k)}) \cdot \mathbf{p}^{(k)} < 0$, then there is a range $[a_k, b_k]$ (with $0 < a_k < b_k$) such that all criteria for a good descent method are satisfied by picking direction $\mathbf{p}^{(k)}$ and step size $t_k \in [a_k, b_k]$.

It's important that there is a range $[a_k, b_k]$ and not just a single value that works, because a range is easier to find.

The textbook has a very interesting sentence about picking t_k: "Finding a numerically efficient, reliable method for setting t_k in practice involves a substantial amount of numerical art as well as science." In other words, "We don't really know what we're doing, but some people know how to make it work."

What we *can* do is determine, for a given value of t_k, whether it satisfies criteria 3 and 4, and if not, whether we should try to make it bigger or smaller.

We know that Criterion 4, $\phi_k(t_k) \leq \phi_k(0) + (\alpha \cdot \phi_k'(0))t_k$,

- is satisfied for all sufficiently small $t_k > 0$, because in that case, we have $\phi_k(t_k) \approx \phi_k(0) + \phi_k'(0) \cdot t_k$, and $\phi_k'(0) < \alpha \cdot \phi_k'(0)$.

[1] This document comes from the Math 484 course webpage: https://faculty.math.illinois.edu/~mlavrov/courses/484-spring-2019.html

- fails for all sufficiently large t_k, because $\phi_k(t)$ is bounded below, and the linear function $\phi_k(0) + (\alpha \cdot \phi_k'(0))t$ is not.

So if Criterion 4 is not satisfied, we should try to make t_k smaller.

On the other hand, suppose t_k is a point where Criterion 4 is satisfied but Criterion 3 is not: ϕ_k is decreasing too quickly at t_k. Looking ahead, we know that the graph of $\phi_k(t)$ will meet the line $\phi_k(0) + (\alpha \cdot \phi_k'(0))t$ at some $t > t_k$. To do that, its average rate of decrease has to be slower than $\alpha \cdot \phi_k'(0)$, so eventually, this rate of decrease has to slow down enough to satisfy Criterion 3. We should try to make t_k larger.

This gives us a "bigger or smaller?" type game to play with t_k until we get it right. The textbook has some suggestions about how to play this game, but they're all heuristics: they all rely on some assumptions about what typical functions look like.

Eventually, once we find a value of t_k that's too small, and a value of t_k that's too large, we have the target range bracketed, and can just keep cutting it in half. Since there's an entire range $[a_k, b_k]$ of values that work, we don't have to find an exact value: we just have to get within $\frac{1}{2}|a_k - b_k|$ of the center of this range.

2 Picking descent directions

The other half of the problem is picking the descent direction $\mathbf{p}^{(k)}$. Wolfe's theorem doesn't have much to say about this: all we've asked for so far is that

$$\nabla f(\mathbf{x}^{(k)}) \cdot \mathbf{p}^{(k)} < 0$$

to satisfy Criterion 2. This is impossible when $\mathbf{x}^{(k)}$ is a critical point (in that case, the dot product is always zero, because $\nabla f(\mathbf{x}^{(k)}) = \mathbf{0}$) but otherwise half of all vectors in $\mathbb{R}^n$ will work.

One obvious choice is $\mathbf{p}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$, since that's what the method of steepest descent does. We don't necessarily want to do that, because somehow that didn't work out very well.

A more general strategy: for any positive definite matrix Q, setting $\mathbf{p}^{(k)} = -Q\nabla f(\mathbf{x}^{(k)})$ will work. Recall that Q is positive definite if $\mathbf{x}^\mathsf{T} Q\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$. Therefore

$$\nabla f(\mathbf{x}^{(k)}) \cdot \mathbf{p}^{(k)} = -\nabla f(\mathbf{x}^{(k)})^\mathsf{T} Q \nabla f(\mathbf{x}^{(k)}) < 0,$$

which is what we want. Equivalently, because Q is positive definite if and only if Q^{-1} is positive definite, we could set $\mathbf{p}^{(k)} = -Q^{-1}\nabla f(\mathbf{x}^{(k)})$ for any positive definite matrix Q.

With that in mind, let's look back at another source of inspiration: Newton's method for minimization. Here, our iterative step is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - Hf(\mathbf{x}^{(k)})^{-1}\nabla f(\mathbf{x}^{(k)})$$

which is almost like a descent method: it's using $t_k = 1$ and $\mathbf{p}^{(k)} = -Q^{-1}\nabla f(\mathbf{x}^{(k)})$ where $Q = Hf(\mathbf{x}^{(k)})$.

This does not always produce a descent direction, because $Hf(\mathbf{x}^{(k)})$ is not always positive definite. It would work when f is a strictly convex function.

To solve this problem, when $Hf(\mathbf{x}^{(k)})$ is not positive definite, we can try to fix this. Suppose that the eigenvalues of $Hf(\mathbf{x}^{(k)})$ are $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. Then the eigenvalues of $Hf(\mathbf{x}^{(k)}) + \mu I$ are

$$\lambda_1 + \mu \leq \lambda_2 + \mu \leq \cdots \leq \lambda_n + \mu.$$

A symmetric matrix is positive definite if and only if all its eigenvalues are positive. So if we pick any $\mu > -\lambda_1$, then $Hf(\mathbf{x}^{(k)}) + \mu I$ will be positive definite, and we can set $\mathbf{p}^{(k)} = -Q^{-1}\nabla f(\mathbf{x}^{(k)})$, where $Q = Hf(\mathbf{x}^{(k)}) + \mu I$.

This lets us compromise between a "Newton-like" descent direction and a "steepest descent-like" descent direction. When μ is small (or even 0, in cases where $Hf(\mathbf{x}^{(k)})$ was already positive definite) the we are going in almost the same direction as Newton's method would go. When μ is large, then $Hf(\mathbf{x}^{(k)}) + \mu I$ is mostly affected by the "μI" part, and taking $Q = \mu I$ would make $\mathbf{p}^{(k)}$ parallel to $-\nabla f(\mathbf{x}^{(k)})$.

3 Summary

Altogether, here is the way an iterative step of a descent method might go.

1. Compute $Hf(\mathbf{x}^{(k)})$, and find some μ_k such that $Hf(\mathbf{x}^{(k)}) + \mu_k I \succ 0$.
2. Choose a descent direction $\mathbf{p}^{(k)}$ by solving

   $$(Hf(\mathbf{x}^{(k)}) + \mu_k I)\mathbf{p}^{(k)} = -\nabla f(\mathbf{x}^{(k)}).$$

   (We avoid computing $(Hf(\mathbf{x}^{(k)}) + \mu_k I)^{-1}$ for the usual reasons.)
3. Choose a step size t_k by trying $t_k = 1$ and then increasing or decreasing t_k until criteria 3 and 4 are both satisfied.
4. Set the next point to $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t_k\mathbf{p}^{(k)}$.