Math 484: Nonlinear Programming[1] Mikhail Lavrov

Chapter 6, Lecture 1: The Penalty Method

April 3, 2019 *University of Illinois at Urbana-Champaign*

1 The absolute value penalty function

In this chapter we also deal with solving constrained optimization problems of the form

$$(P) \qquad \begin{cases} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{g}(\mathbf{x}) \le \mathbf{0} \end{cases}$$

but the theory of the penalty method is much less elaborate.

As an easy-to-think-about example, we'll take a 1-dimensional problem: to minimize x^2, subject to the constraint $x \ge 1$ (that is, $1 - x \le 0$).

The basic premise is this: we modify the objective function so that, whenever a constraint is violated, we pay a penalty for it. If the penalty is harsh enough, then the unconstrained minimum of the modified function will satisfy all the constraints anyway. At least, that's the hope.

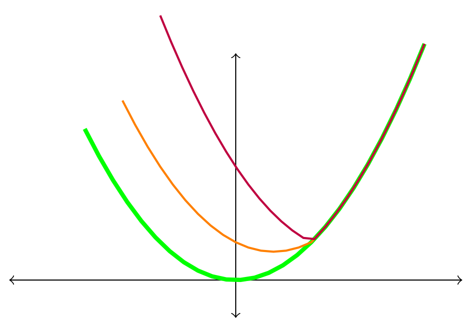
For our first attempt at this, given a function $g(\mathbf{x})$, define

$$g^+(\mathbf{x}) = \max\{0, g(\mathbf{x})\} = \begin{cases} 0 & \text{if } g(\mathbf{x}) \le 0, \\ g(\mathbf{x}) & \text{if } g(\mathbf{x}) \ge 0. \end{cases}$$

Instead of solving the constrained minimization problem P, we solve the unconstrained problem of minimizing

$$F_k(\mathbf{x}) = f(\mathbf{x}) + k \cdot \left[g_1^+(\mathbf{x}) + g_2^+(\mathbf{x}) + \cdots + g_m^+(\mathbf{x})\right],$$

where k is some very large number. In the example, this gives us the following curves ($k = 0$ in green, $k = 1$ in orange, $k = 3$ in purple):



The minimum of $x^2 + k(1 - x)^+$ is achieved at the desired point 1, provided $k \ge 2$.

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

So far, so good. But there's one problem with this approach. How do we minimize an unconstrained function? We set the derivative equal to 0. If $F_k(x) = x^2 + k(1-x)^+$, then

$$F_k'(x) = \begin{cases} 2x & x > 1, \\ 2x - k & x < 1, \\ \text{undefined} & x = 1. \end{cases}$$

The first case is 0 when $x = 0$, which never satisfies $x > 1$. The second case is 0 when $x = \frac{k}{2}$, which satisfies $x < 1$ only for small k. The third case is undefined, so we have to check it anyway.

In this 1-dimensional example, we only have one undefined point, $x = 1$. But if we generalize this method to actually relevant problems, things get worse: in the n-dimensional case, the gradient $\nabla F_k(\mathbf{x})$ is undefined whenever $g_i(\mathbf{x}) = 0$ for any i. So we end up checking the entire boundary of the feasible region of the problem as a special case.

A lot of the time, the optimal point will be on the boundary of the feasible region. So this method won't really help us there.

2 The Courant–Beltrami penalty function

The Courant–Beltrami penalty function modifies the method by paying a different penalty for violating the constraints: a penalty that makes sure that (provided we start with differentiable functions) the resulting function F_k will always be differentiable. Here, we take

$$F_k(\mathbf{x}) = f(\mathbf{x}) + k\left[(g_1^+(\mathbf{x}))^2 + (g_2^+(\mathbf{x}))^2 + \cdots + (g_m^+(\mathbf{x}))^2\right].$$

This has a continuous gradient because $\max\{0, x^2\}$ has a continuous first derivative: at 0, the derivative is 0 from both sides. In general, when we take the gradient of $(g_+(\mathbf{x}))^2$, we get

$$\nabla(g_+(\mathbf{x}))^2 = \begin{cases} \nabla(g(\mathbf{x}))^2 = 2g(\mathbf{x})\nabla g(\mathbf{x}) & g(\mathbf{x}) \geq 0, \\ \mathbf{0} & g(\mathbf{x}) \leq 0. \end{cases}$$

We can write this in one line as $\nabla(g_+(\mathbf{x}))^2 = 2g^+(\mathbf{x})\nabla g(\mathbf{x})$.

So in our easy example, we set

$$F_k(x) = x^2 + k[(1-x)^+]^2 = \begin{cases} x^2 & x \geq 1, \\ x^2 + k(1-x)^2 & x \leq 1. \end{cases}$$

Taking the derivative, we get

$$F_k'(x) = 2x - 2k(1-x)^+ = \begin{cases} 2x & x \geq 1, \\ 2x - 2k(1-x) & x \leq 1. \end{cases}$$

Just to reassure ourselves, we can check that $F_k'(1) = 2$ by using either case of the definition, so $F_k'(x)$ exists and is continuous for all x.

Now we solve $F_k'(x) = 0$ for x.

- In the case $x \geq 1$, we set $2x = 0$, and get $x = 0$ as the answer. This does not satisfy $x \geq 1$, so we throw it out. No critical points arise in this case.
- In the case $x \leq 1$, we set $2x - 2k(1-x) = 0$, or $(2+2k)x = 2k$, and get $x = \frac{2k}{2+2k} = \frac{k}{k+1}$ as the answer. For all $k \geq 0$, this satisfies $x \leq 1$, so we do get a critical point.

So the only critical point of $F_k(x)$ is at $x = \frac{k}{k+1}$. In fact, $x = \frac{k}{k+1}$ is a global minimizer of $F_k(x)$.

Note that $\frac{k}{k+1}$ is not actually feasible for the original problem: it does not satisfy the constraint $x \geq 1$. This is typical when using the Courant–Beltrami penalty function. However, $\frac{k}{k+1}$ is still a useful answer, because as $k \to \infty$, $\frac{k}{k+1} \to 1$, and $x = 1$ is the optimal solution to the original problem.

We are going to see that this will happen for sufficiently nice nonlinear programs. We might encounter several difficulties along the way:

- If the objective function $f(\mathbf{x})$ can approach $-\infty$ somewhere outside the feasible region, then the penalty method becomes a race between how quickly f decreases and how quickly the penalty term increases to make up for it.

  We can't guarantee anything about who wins that race, so we will assume that f is bounded from below.

- Under mild assumptions, we'll be able to show that, provided that the optimal solution to the penalty problem *exists* and converges to *something* as $k \to \infty$, then it converges to an optimal solution to the original problem.

  So then we will want to have some reason to expect this convergence to happen.

- Sometimes, the original problem has multiple optimal solutions. This is not really a problem for applying the method, as we'll see; it is a problem for having nice statements for our theorems, because the optimal solution to the penalty problem can't converge to multiple things at the same time.

  This will make our theorems a bit more complicated; we'll have to talk about "convergent subsequences" and the like.