Math 484: Nonlinear Programming[1] Mikhail Lavrov

Chapter 3, Lecture 1: Newton's Method

April 15, 2019 *University of Illinois at Urbana-Champaign*

1 Approximate methods and asumptions

The final topic covered in this class is iterative methods for optimization. These are meant to help us find approximate solutions to problems in cases where finding exact solutions would be too hard.

There are two things we've taken for granted before which might actually be too hard to do exactly:

1. Evaluating derivatives of a function f (e.g., ∇f or Hf) at a given point.
2. Solving an equation or system of equations.

Before, we've assumed (1) and (2) are both easy. Now we're going to figure out what to do when (2) and possibly (1) is hard. For example:

- For a polynomial function of high degree, derivatives are straightforward to compute, but it's impossible to solve equations exactly (even in one variable).
- For a function we have no formula for (such as the value function $\text{MP}(\mathbf{z})$ from Chapter 5, for instance) we don't have a good way of computing derivatives, and they might not even exist.

2 The classical Newton's method

Eventually we'll get to optimization problems. But we'll begin with Newton's method in its basic form: an algorithm for approximately finding zeroes of a function $f : \mathbb{R} \to \mathbb{R}$. This is an iterative algorithm: starting with an initial guess x_0, it makes a better guess x_1, then uses it to make an even better guess x_2, and so on. We hope that eventually these approach a solution.

From a point x_k, Newton's method does the following:

1. Compute $f(x_k)$ and $f'(x_k)$.
2. Approximate $f(x)$ by the linear function $f(x_k) + (x - x_k)f'(x_k)$.
3. Let x_{k+1} be the point where the linear approximation is zero:

$$f(x_k) + (x_{k+1} - x_k)f'(x_k) = 0 \iff x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

If we start with a good enough x_0, and compute $x_1, x_2, x_3, \dots$, then with some luck, as $k \to \infty$, $x_k \to x^*$, which satisfies $f(x^*) = 0$.

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

2.1 Examples

When Newton's method works, it usually works very well. For example, if we try to solve $4x^3 + x - 1 = 0$ with an initial guess of $x_0 = 1$, we'll get:

$$\begin{aligned} x_0 &= 1.\ 00000\ 00000 \\ x_1 &= 0.\ 69230\ 76923 \\ x_2 &= 0.\ 54129\ 30628 \\ x_3 &= 0.\ 50239\ 01750 \\ x_4 &= 0.\ 50000\ 85354 \\ x_5 &= 0.\ 50000\ 00001 \end{aligned}$$

and it's easy to see that we are getting closer and closer to $x^* = 0.5$ which really is a solution of the equation.

This example is an ideal case, where the convergence is quadratic: once x_k is sufficiently close to x^*, $|x^* - x_{k+1}| \le C|x^* - x_k|^2$. This means that the number of correct digits roughly doubles at each step. You should think of this type of convergence as "fast" convergence.

Occasionally, we get worse behavior. For example, if we try to solve $4x^3 - 3x + 1 = 0$ with an initial guess of $x_0 = 0$, we get

$$\begin{aligned} x_0 &= 1.\ 00000\ 00000 \\ x_1 &= 0.\ 33333\ 33333 \\ x_2 &= 0.\ 42222\ 22222 \\ &\vdots \\ x_{10} &= 0.\ 49971\ 21712 \\ &\vdots \\ x_{20} &= 0.\ 49999\ 97190 \end{aligned}$$

which would maybe have looked impressive if you hadn't seen the previous example first. In this example, we get linear convergence: even when x_k is close to the limit x^*, we only have $|x^* - x_{k+1}| \le C|x^* - x_k|$ for some $C < 1$. You should think of this type of convergence as "slow convergence".

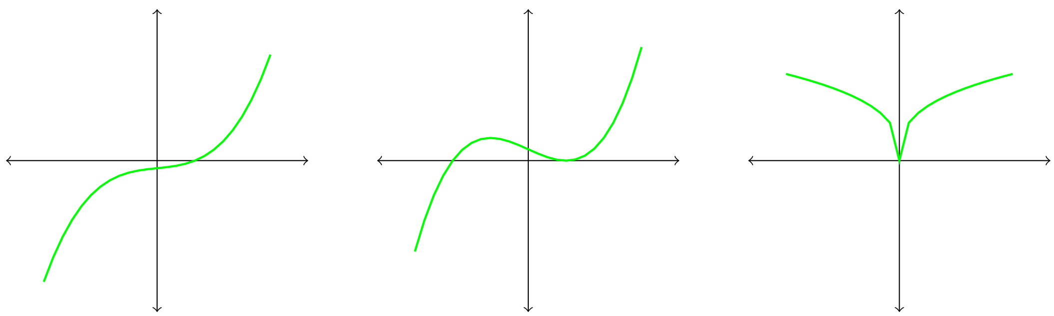
You might ask: can't we also encounter convergence rates where $|x^* - x_{k+1}| \le C|x^* - x_k|^\alpha$ for some α other than 1 or 2? There are methods that do this—for example, there are variants of Newton's method which can sometimes achieve cubic convergence. However, all values of $\alpha > 1$ are very similar to each other, and very different from $\alpha = 1$. When $\alpha = 1$, then depending on C, we get a fixed number of new correct digits at each step. When $\alpha > 1$, the number of correct digits grows exponentially. The base of the exponent varies with α, but exponential growth is very quick no matter what the base of the exponent is.

Going back to Newton's method: it's also possible for it to never converge at all. For example, suppose we take $f(x) = |x|^{1/3}$, for which the only root is $x^* = 0$. Starting with $x_0 = 1$, we get

$x_1 = -2$, $x_2 = 4$, $x_3 = -8$, $x_4 = 16$, and in general $x_k = (-2)^k$. We are getting further and further away from the desired value of x^* as we go.

There are other ways in which we can fail to converge; for example, we might get into a loop where, after a few iterations, we're back where we started. We might also end up at a point x_k where $f'(x_k) = 0$, in which case Newton's method doesn't give us a way to proceed at all.

We get an intuitive explanation for what's going on in these three examples once we graph all three functions. From left to right, the graphs of $4x^3 + x - 1$, $4x^3 - 3x + 1$, $|x|^{1/3}$ are:



In the first case, we got fast convergence because, near the root, we have a straightforward linear approximation to the function and everything is nice.

In the second case, at the root x^*, we have not just $f(x^*) = 0$ but also $f'(x^*) = 0$. Points where $f'(x)$ is close to 0 are normally very bad for Newton's method, because then the linear approximation is very nearly horizontal. So we have to fight this problem all along the way, and get slow convergence as a result.

In the third case, at the root x^*, the derivative $f'(x^*)$ is undefined. It's not surprising that when the root is as poorly behaved as this one, Newton's method has trouble converging to it.

Assuming we're not in the third case where $f'(x^*)$ is undefined, Newton's method is guaranteed to converge, provided we have a sufficiently good initial guess. We won't prove this, but intuitively speaking, the local behavior of any nice function near the root is going to be more or less the same as the local behavior of one of our first two examples, so it's going to converge because they do.

But if we start out very far from x^*, it takes some luck to get close enough to x^* for Newton's method to start actually being helpful.

3 Variants of Newton's method

3.1 Newton's method for minimization

Suppose we have a function $f : \mathbb{R} \to \mathbb{R}$. But now, we want to minimize (or maximize) f. Newton's method can be adapted to this problem; we can think of how to do this in two ways.

First, we can try find a critical point of f: a place where $f'(x) = 0$. This is something that ordinary

Newton's method can do. Starting at some guess x_0, we follow the iteration

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

which is Newton's method applied to the function f'.

Second, we can imagine working with f by approximating it by a quadratic function, minimizing that, and having that be our iterative procedure at each step. (Why a quadratic? Because that's the smallest degree at which we could, in principle, have a minimizer or maximizer.) From the point x_k, we approximate

$$f(\mathbf{x}) \approx f(x_k) + f'(x_k)(x - x_k) + f''(x_k)\frac{(x - x_k)^2}{2}.$$

A quadratic equation $y = ax^2 + bx + c$ has its vertex at x-coordinate $-\frac{b}{2a}$. In this case, thinking of this parabola in terms of $x - x_k$, we get $-\frac{b}{2a} = -\frac{f'(x_k}{2 \cdot \frac{1}{2} f''(x_k)}$. So if our next iterative point x_{k+1} is the vertex of the parabola, it must satisfy

$$x_{k+1} - x_k = -\frac{f'(x_k)}{f''(x_k)}$$

which gives us the same recurrence as before.

Any way we look at it, this method is not very intelligent: it only finds critical points of f, so it cannot distinguish minimizers from maximizers. So when we use this method, we will want to use the second derivative test to classify the limiting point as a local minimizer or local maximizer, and if we got the wrong one, we can start over from some other point far away and try again.

3.2 The secant method

The secant method is a replacement for ordinary Newton's method when we can't compute derivatives. In this case, we replace the derivative $f'(x_k)$ by the approximation $\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ using the values of f at the *two* most recent points.

This gives us the iterative step

$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1})$$

which involves no derivatives. That's the secant method.

As we approach a solution x^*, x_k and x_{k-1} become very close to each other, and therefore $\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$ is a very good approximation of $f'(x_k)$ (or of $f'(x_{k-1})$). So we expect this method to perform almost as well as Newton's method (in cases where Newton's method does perform well).

We lose some efficiency, of course, because the approximation is not perfect. But the advantage is that we don't have to compute derivatives, so we can deal with a wider class of problems.