2022-03-25

1 Nonlinear equations and optimization

If $f : \mathbb{R}^n \to \mathbb{R}^n$, then solving the system $f(x) = 0$ is equivalent to minimizing $\|f(x)\|^2$. Similarly, if $g : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, then any local minimizer x_* satisfies the nonlinear equations $\nabla g(x_*) = 0$. There is thus a close connection between nonlinear equation solving on the one hand and optimization on the other, and methods used for one problem can serve as the basis for methods for the other.

As with nonlinear equations, the one-dimensional case is the simplest, and may be the right place to start our discussion. As with the solution of nonlinear equations, our main strategy for dealing with multi-variable optimization problems will be to find a promising search direction and then solve (approximately) a one-dimensional line search problem.

2 Minimization via 1D Newton

Suppose $g : \mathbb{R} \to \mathbb{R}$ has at least two continuous derivatives. If we can compute g' and g'', then one of the simplest ways to find a local minimum is to use Newton iteration to find a stationary point:

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

Geometrically, this is equivalent to finding the maximum (or minimum) of a second-order Taylor expansion about x_k; that is, x_{k+1} is chosen to minimize (or maximize)

$$\hat{g}(x_{k+1}) = g(x_k) + g'(x_k)(x_{k+1} - x_k) + \frac{1}{2}g''(x_k)(x_{k+1} - x_k)^2.$$

The details are left as an exercise.

There are two gotchas in using Newton iteration in this way. We have already run into the first issue: Newton's method is only locally convergent. We can take care of that problem by combining Newton with bisection, or by scaling down the length of the Newton step. But there is another issue, too: saddle points and local maxima are also stationary points!

There is a simple precaution we can take to avoid converging to a maximum: insist that $g(x_{k+1}) < g(x_k)$. If $x_{k+1} = x_k - \alpha_k u$ for some $\alpha_k > 0$, then

$$g(x_{k+1}) - g(x_k) = -\alpha_k g'(x_k)u + O(\alpha_k^2).$$

So if $g'(x_k)u > 0$, then $-u$ is a *descent direction*, and thus $g(x_{k+1}) < g(x_k)$ provided α_k is small enough. Note that if x_k is not a stationary point, then $-u = -g'(x_k)/g''(x_k)$ is a descent direction iff $g'(x_k)u = g'(x_k)^2/g''(x_k) > 0$. That is, we will only head in the direction of a minimum if $g''(x_k)$ is positive. Of course, g'' will be positive and the Newton step will take us in the right direction if we are close enough to a strong local minimum.

3 Approximate bisection and golden sections

Assuming that we can compute first derivatives, minimizing in 1D reduces to solving a nonlinear equation, possibly with some guards to prevent the solver from wandering toward a solution that does not correspond to a minimum. We can solve the nonlinear equation using Newton iteration, secant iteration, bisection, or any combination thereof, depending how sanguine we are about computing second derivatives and how much we are concerned with global convergence. But what if we don't even want to compute first derivatives?

To make our life easier, let's suppose we know that g is twice continuously differentiable and that it has a unique minimum at some $x_* \in [a, b]$. We know that $g'(x) < 0$ for $a \leq x < x_*$ and $g'(x) > 0$ for $x_* < x \leq b$; but how can we get a handle on g' without evaluating it? The answer lies in the mean value theorem. Suppose we evaluate $g(a)$, $g(b)$, and $g(x)$ for some $x \in (a, b)$. What can happen?

1. If $g(a)$ is smallest ($g(a) < g(x) \leq g(b)$), then by the mean value theorem, g' must be positive somewhere in (a, x). Therefore, $x_* < x$.

2. If $g(b)$ is smallest, $x_* > x$.

3. If $g(x)$ is smallest, we only know $x_* \in [a, b]$.

Cases 1 and 2 are terrific, since they mean that we can improve our bounds on the location of x_*. But in case 3, we have no improvement. Still, this is promising. What could we get from evaluating g at *four* distinct points $a < x_1 < x_2 < b$? There are really two cases, both of which give us progress.

1. If $g(x_1) < g(x_2)$ (i.e. $g(a)$ or $g(x_1)$ is smallest) then $x_* \in [a, x_2]$.
2. If $g(x_1) > g(x_2)$ (i.e. $g(b)$ or $g(x_2)$ is smallest) then $x_* \in [x_1, b]$.

We could also conceivably have $g(x_1) = g(x_2)$, in which case the minimum must occur somewhere in (x_1, x_2).

There are now a couple options. We could choose x_1 and x_2 to be very close to each other, thereby nearly bisecting the interval in all four cases. This is essentially equivalent to performing a step of bisection to find a root of g', where g' at the midpoint is estimated by a finite difference approximation. With this method, we require two function evaluations to bisect the interval, which means we narrow the interval by $1/\sqrt{2} \approx 71\%$ per evaluation.

We can do a little better with a *golden section search*, which uses $x_2 = a + (b-a)/\phi$ and $x_1 = b + (a-b)/\phi$, where $\phi = (1+\sqrt{5})/2$ (the *golden ratio*). We then narrow to the interval $[a, x_2]$ or to the interval $[x_1, b]$. This only narrows the interval by a factor of ϕ^{-1} (or about 61%) at each step. But in the narrower interval, we get one of the two interior function values "for free" from the previous step, since $x_1 = x_2 + (a - x_2)/\phi$ and $x_2 = x_1 + (b - x_1)/\phi$. Thus, each step only costs one function evaluation.

4 Successive parabolic interpolation

Bisection and golden section searches are only linearly convergent. Of course, these methods only use coarse information about the relative sizes of function values at the sample points. In the case of root-finding, we were able to get a superlinearly convergent algorithm, the secant iteration, by replacing the linear approximation used in Newton's method with a linear interpolant. We can do something similar in the case of optimization by interpolating g with a *quadratic* passing through three points, and then finding a new guess based on the minimum of that quadratic. This *method of successive parabolic interpolation* does converge locally superlinearly. But even when g is unimodular, successive parabolic interpolation must generally be coupled with something slower but more robust (like golden section search) in order to guarantee good convergence.

Problems to ponder

1. Suppose I know $f(0)$, $f(1)$, and a bound $|f''| < M$ on $[0,1]$. Under what conditions could f possibly have a local minimum in $[0,1]$?

2. Suppose $f(x)$ is approximated on $[0,1]$ by a polynomial $p(x) = c_0 + c_1 x + \ldots + c_d x^d$, and we know that $|f(x) - p(x)| < \delta$ on the interval. Using MATLAB's `roots` function, how could we find tight subintervals of $[0,1]$ in which the global minimum of $f(x)$ might lie?