SNewton: safeguarded Newton methods for function minimization

John C. Nash

2017-04-10

Safeguarded Newton algorithms

So-called **Newton** methods are among the most commonly mentioned in the solution of nonlinear equations or function minimization. However, as discussed in https://en.wikipedia.org/wiki/Newton%27s_method#History, the **Newton** or **Newton-Raphson** method as we know it today was not what either of its supposed originators knew.

This vignette discusses the development of some safeguarded variants of Newton methods for function minimization in **R**. Note that there are some resources in **R** for solving nonlinear equations by Newton-like methods in the packages **nleqslv** and **pracma**.

The basic approach

If we have a function $f(x)$, with gradient $g(x)$ and second derivative (Hessian) $H(x)$ the first order condition for an extremum (min or max) is

$g(x) = 0$

To ensure a minimum, we want

$ H(x) > 0 $

The first order condition leads to a root-finding problem.

It turns out that x need not be a scalar. We can consider it to be a vector of parameters to be determined. This renders $g(x)$ a vector also, and $H(x)$ a matrix. The conditions of optimality then require a zero gradient and positive-definite Hessian.

The Newton approach to such equations is to provide a guess to the root x_try and to then solve the equation

$ H(x_t) * s = - g(x_t)$

for the search vector s. We update x_t to $x_t + s$ and repeat until we have a very small gradient $g(x_t)$. If $H(x)$ is positive definite, we have a reasonable approximation to a (local) minimum.

Motivations

A particular interest in Newton-like methods its theoretical quadratic convergence. See https://en.wikipedia.org/wiki/Newton%27s_method. That is, the method will converge in one step for a quadratic function $f(x)$, and for "reasonable" functions will converge very rapidly. There are, however, a number of conditions, and practical programs need to include safequards against mis-steps in the iterations.

The principal issues concern the possiblity that $H(x)$ may not be positive definite, at least in some parts of the domain, and that the curvature may be such that a unit step $x_t + s$ does not reduce the function f. We therefore get a number of possible variants of the method when different possible safeguards are applied.

Algorithm possibilities

There are many choices we can make in building a practical code to implement the ideas above. In tandem with the two main issues expressed above, we will consider

- the modification of the solution of the main equation $ H(x_t) * s = - g(x_t)$ so that a reasonable search vector s is always generated
- the selection of a new set of parameters $x_new = x_t + step * s$ so that the function value $f(x_new)$ is less than $f(x_t)$.

The second choice above could be made slightly more stringent so that the Armijo (??ref) condition of sufficient-decrease is met. Adding a curvature requirement gives the Wolfe condisions. See https://en.wikipedia.org/wiki/Wolfe_conditions. The Armijo requirement is generally written

$f(x_t + step * s) < f(x_t) + c * step * g(x_t)^T * s$

where c is some number less than 1. Typically $ c = 1e-4 = 0.0001 $. Note that the product of gradient times search vector is negative for any reasonable situation, since we are trying to go "downhill".

Some choices to compute the search vector

The primary concern in solving for s is that the Hessian may not be positive definite. This means that we cannot apply fast and stable methods like the Cholesky decomposition to the matrix. At the time of writing, we consider the following approaches:

- Attempt to solve $ H(x_t) * s = - g(x_t)$ with **R** directly, and rely on internal checks to catch any cases where the solution fails. We can use `try()` to stop the program in this case.
- Use a Levenberg-Marquardt (??ref) stabilization to ensure that we have an augmented Hessian that is positive definite. Essentially, we create $H_aug = H + \lambda * I$ where I is the unit matrix of the size of H
- Use the singular value decomposition and drop any singular planes where the singular values fall below some threshhold. Note that deciding the threshhold is possibly a non-trivial matter.

Note that within the above general choices for solution, we could try to specify how the solution is obtained, since there are various ways to solve linear equations and to find the singular value decomposition. (??refs, discussion??)

Choosing the step size