ECONOMETRICS II, Fall 2020.

Bent E. Sørensen

Numerical maximization of function.

Consider a function $f(x)$, where x is a vector. In ML and GMM, the vector is the vector of parameters. We need f to be concave with single maximum (strictly speaking on a compact set, to rule out convergence to another maximum).

We do a second order Taylor expansion:

$$f(x) = f(x_0) + Df_0(x - x_0) + 0.5(x - x_0)'D^2 f_0(x - x_0)\,,$$

where the derivatives Df_0 (a row vector) and $D^2 f_0$ (a matrix) are evaluated at x_0. Let us, for now, assume that this approximation is the true function. Let us find the maximum of that function by taking the first order derivative and setting it to 0. We have

$$Df_0\prime + D^2 f_0(x - x_0) = 0$$

so the solution is

$$x = x_0 - D^2 f_0^{-1} Df_0\prime\,.$$

Notice that this is not going to go well unless $D^2 f_0$ is positive definite (implying full rank, so it can be inverted). Let us check what we get if the function actually is linear-quadratic, as in

$$F(x) = A + BX + X'CX$$

We have solution $X = -C^{-1}B$. If we look at

$$F(\beta) = -(Y - X\beta)'(Y - X\beta)\ ,$$

we have $Df_0 = 2X'(Y - X\beta)$ which is $2X'Y$ evaluated at $\beta = 0$ and $D^2F_0 = -2X'X$. We find $\beta = (X'X)^{-1}X'Y$.

For the Newton algorithm, we chose an initial value, call it x_0. We then find the maximum in the second order approximation:

$$x_1 = x_0 - D^2 f_0^{-1} D f_0\prime .$$

But this involved an approximation, so x_1 is not the global maximum for the original function. But now let us do a second order approximation starting from x_1 which is closer to the maximum of the original function. So we find the derivatives Df_1 (a row vector) and $D^2 f_1$ (a matrix) are evaluated at x_1. We then use the second order expansion and find the maximum, we know the formula now, and we find a new maximum (for the approximation)

$$x_2 = x_1 - D^2 f_1^{-1} D f_1\prime .$$

Keep doing this and you will (usually) end up a the global maximum (convergence). If the model is "badly conditioned," it may not work. Here you should think of OLS. A linear model is said to be badly conditioned if $X'X$ is nearly singular (multicollinearity) which leads to unstable estimates. You do not like to invert something that is almost singular. For the non-linear model $D^2 f$ plays the role of $X'X$ and if that is near singular, it may be hard to converge. The problem is much harder in the non-linear model, because if you need 100 iterations before convergence, you find the Hessian (2nd derivative) 100 times and if just one of those are singular, your procedure may crash. Too bad. What to do? Try some other starting values. Sometimes you maybe have a simpler version of the model that you can use to find good starting values. The more you ask of the data, the better data you need. Just as for the linear model. But the more non-linear your model is, the more this matters.

In most cases in econometrics, when we maximize a likelihood or minimize the GMM weighted sum of squares, we do not find the derivatives and the Hessian. For a function f, $f\prime$ evaluated at x_0 is $\lim \epsilon \rightarrow \frac{f(x+\epsilon)}{\epsilon}$. The computer will choose a very small ϵ and approximate $f(x_0)\prime = \frac{f(x+\epsilon)}{\epsilon}$ for a small ϵ. As a technical note, do not choose units such that x is of the order 10^{-9} or something crazy like that, you program will use a small ϵ value, but it has to be small relative to x. You probably don't want x is of the order 10^9 either. Those things do not matter in the linear model (except for making you coefficients have a silly scale), but they may matter for the Newton algorithm to work. If you have a very difficult model, the Hessian may become singular because of the numerical approximation and even if it doesn't calculations may take a very long time, most of that time spent on finding derivatives. (I have a paper published in the *Journal of Econometrics*, where we simply could not get a solution in "finite time" unless we found and programmed up the derivative. I forget if we also did the Hessian, likely not.)

Most programs come with some bags of tricks. The formula for x_1 takes a step away from x_0, it is cheap to try to see if taking two steps or a half step get us closer to the maximum. Or programs may not find a new Hessian in each step or somehow approximate the Hessian. So most econometric packages give you some choice of algorithm. Usually, if one doesn't work, you model is to hard for the data to pin down and the others won't work either.