Homework 3: Improving Estimation by Nonlinear Least Squares

36-350, Fall 2011

Due at the start of class, Wednesday, 21 September 2011

INSTRUCTIONS: Submit a single plain text file, whose name clearly includes both your Andrew ID and the assignment number. Inside the file, clearly indicate which parts of your responses go with which problems. Raw R output is not acceptable, and will be marked down accordingly; you are communicating with a human being, and need to write in a human language. Files in other formats (Word, PDF, etc.) will receive a grade of zero. Homework not submitted through Blackboard will receive a grade of zero.

Direct objective: Practice with writing and organizing functions.

Indirect objectives: Fitting statistical models; testing small pieces of the code before trusting them.

Read the slides for lectures 4 and 5, lab 2, and section 4.2 of the textbook before beginning.

BACKGROUND: Just as a function of one variable $f(u)$ is at an extremum when the derivative is zero, $df/du = 0$, a function of multiple variables, say $g(u, v)$, is at an extremum when all the partial derivatives are zero, $\partial g/\partial u = \partial g/\partial v = 0$. In the last lab, we estimated the parameter a in a nonlinear model,

$$Y = y_0 N^a + \text{noise} \tag{1}$$

by minimizing the mean squared error

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - y_0 N_i^a)^2 \tag{2}$$

We did this by "gradient descent": we approximated the derivative of the MSE, and adjusted a by an amount proportional to that, stopping when the derivative became small. Our procedure assumed we knew y_0. In this assignment, we will see how to estimate two parameters at once.

1. Write a function, called `mse()`, which calculates the mean squared error of the model in Eq. 1 on a given data set. Your function should take three arguments: a numeric vector of length two, the first component standing for y_0 and the second for a; a numerical vector containing the values of N; and a numerical vector containing the values of Y. The function should

return a single numerical value. The latter two arguments should have as the default values the columns `pop` and `pcgmp` (respectively) from the `gmp` data frame in Lab 2. Your function may not use `for()` or any other loop.

Hint: Look at the slides for Lecture 4. (20)

2. Check that, with the default data, you get the following values. (5)

```
> mse(c(6611,0.15))
[1] 207057513
> mse(c(5000,0.10))
[1] 298459915
```

3. Write a function, `mse.grad()`, which approximates the gradient of the mean squared error. It should take three arguments: a vector of length 2 giving the point at which we want the gradient; the increment for y_0; and the increment for a. Provide default values for the latter two arguments. It should return a length two vector containing the gradient. (20)

4. Check that you get the following values (5)

```
> mse.grad(c(6611,0.15),10,1e-5)
[1] 1.650303e+05 1.429197e+10
> mse.grad(c(5000,0.10),7,-1e-5)
[1]     -109811.2 -7129496031.7
```

5. Write a function, `my.nls()`, which estimates the parameters y_0 and a of the model (1) by minimizing the mean squared error, and minimizes the MSE by gradient descent. It should take the following arguments: an initial guess for y_0; an initial guess for a; a vector containing the N values; a vector containing the Y values; the increments for calculating the gradient; the factor by which the gradient is multiplied before subtracting it from the parameter vector; the threshold below which the gradient is considered effectively zero; and the maximum number of iterations. All arguments except the initial guesses should have suitable default values. It should return a list with the following components: the final guess for y_0; the final guess for a; the final gradient; the number of iterations taken; a flag for whether the function stopped before running out of iterations.

   Your function must call those you wrote in earlier questions, and the appropriate arguments to `my.nls()` should be passed on to them.

   Hint: See the slides for lecture 4. (30)

6. What parameter estimate do you get when starting from $y_0 = 6611$ and $a = 0.15$? From $y_0 = 5000$ and $a = 0.10$? If these are not the same, why do they differ? (10)

7. Suggest (but do not implement) a way to quantify the uncertainty in this estimate of the parameters. (10)