Lab session 5: NLS hints & Intro to likelihood estimation

Adam Theising[*]

February 21st, 2018

Contents

Getting started...

- Questions/comments?...
- Will get you our answer key for assignment 2 by end of the week. Grades and comments on your code to follow.
- Hopefully you've taken a peek at assignment 3; its a LONG one, and will take time and effort... start early!
- To get you guys moving ahead with that, we'll start today with a discussion of how to approach these problems. Then I'll segue into some basic math stats that will hopefully serve as a primer for where Brian is heading next: maximum likelihood estimation.

1 Hints for assignment 3

My goal with these particular hints is to give you some guidance so that you don't waste time going down the wrong path with your code. Hopefully they prove helpful to you! Here's a general process I'd recommend for all the problems this week:

1. Read in and clean/format your data (as necessary).
2. Specify the functional form of your model in a separate .m file. By doing this ONCE, you can use numerical methods to calculate all necessary Jacobians/Hessians.
3. Pin down some reasonable starting values - see below for more thoughts on this.
4. Estimate parameters - compare results across algorithms (GN or NR), starting values, minimum step sizes, etc. to ensure your estimates make sense and obtain *global* minima.
5. Post-estimation hypothesis testing.
6. Write up requested results and/or output tables.

[*]theising@wisc.edu

Last thing before diving into some more specific hints - I'd highly recommend the Madsen et al (2004) paper linked to in the assignment if you're interested in learning more about descent methods or alternative nonlinear least squares algorithms. It highlights a number of useful computational tricks.

Question 1: This week's assignment begins by estimating another non-linear model with the Australian wool export data. Though the assignment takes 4 pages to describe the question, its premise is straightforward: design an NLS program that offers the choice of either the Gauss-Newton or Newton-Raphson algorithm. *Pay attention to constructing strong code here! You will use it for the rest of the assignment and also intermittently throughout the rest of the semester.*

In class, Brian walked through his generalized Gauss-Newton algorithm. This code should be **very** instructive. Remember, both algorithms operate on similar principles and the main difference is just the step length between iterations:

$$\beta_{n+1} = \beta_n - v_n P_n \frac{\partial S(\beta)}{\partial \beta}\bigg|_{\beta=\beta_n}$$

where v_n is your variable step length and

$$P_n = \begin{cases} \frac{1}{2}\left(Z(\beta_n)'Z(\beta_n)\right)^{-1} & \text{for Gauss-Newton} \\ (H(\beta_n))^{-1} & \text{for Newton-Raphson} \end{cases}$$

Some directed questions to get you reflecting on what the above equations mean in practice - in which algorithm will you need to make use of numerical gradients? In which will you need a numerical Hessian? Which functions from lecture can you use for these?[1]

Other assorted tips:

- How to pick starting values? There are many ways: run a simple and similar OLS regression, pick parameter values that theory suggests are reasonable, or run an initial, coarse grid search. Whatever you do, justify your logic when answering the question.
- How to create an option for both the GN and NR algorithms? Hint: use MATLAB's switch function. You can either have completely separate functions for GN and NR and use a *switch* or *if* command to call the appropriate one, OR you can embed the switch function inside Brian's *nls.m* file at the appropriate moment to attain identical results.
- Having trouble attaining convergence or getting error messages about nonsingular matrices? Try scaling your data - ie dividing or multiplying RHS or LHS vars by 100s or 1000s - so that they are similar in magnitude. If you do this intelligently, it shouldn't affect your parameter estimates.
- How to test for convexity of SSE function at estimated parameter values? Check your notes for something about the SSE Hessian being positive semi-definite... eigenvalues?
- Call stack example:

Call Stack			
Function name	Purpose	Arguments	Global variables
`NR`	Estimate NLS model via NR algorithm	various	various
`hessian_bwg`	Compute Hessian of the SSE function	fct name; betas	none
`sse_fn`	Compute SSE of desired model	betas	`lhsvar`; `func_name`
`CES`	Compute predicted values for model	betas	`rhsvar`

[1] Here's your functions: *Grad.m* and *hessian_bwg.m*

Questions 2 and 3: Given you've done a nice job on your code for Q1, these question should prove fairly straightforward in terms of coding! The challenges here will lie mostly in hypothesis testing. For Q2, Brian's blurb describing the Box-Cox transformation is very thorough, but let me give an additional example, and also tie in the Box-Cox transformation with the CES production function you'll see in Q3.

It's common to represent production (Q) as a function of capital (K) and labor (N). The most well known production functions are probably the Cobb-Douglas and CES functions. Here, however, suppose that $\tau(\cdot)$ is a Box-Cox transformation $\left(\tau(x,\lambda) = \frac{x^\lambda - 1}{\lambda}\right)$ as described in the assignment. Then imagine a production function as:

$$\tau(Q) = \alpha\tau(K) + (1-\alpha)\tau(N)$$

If we solve for Q by inverting the Box-Cox transformation, we have:

$$Q = \left(\alpha K^\lambda + (1-\alpha)N^\lambda\right)^{1/\lambda}$$

This should look familiar if you take a peek at Q3- it's a simple form of the CES production function! Econometrically speaking, when $\lambda = 1$, we have a linear production function ($Q = \alpha K + (1-\alpha)N$) and if we apply l'Hopital's rule as $\lambda \to 0$, we have a Cobb-Douglas production function ($Q = K^\alpha N^{1-\alpha}$) which could be log-linearized easily.[2] And at the heart of it all, if we relax restrictions and let the data speak, we can estimate λ directly via NLS, allowing for a fairly flexible functional specification. And see, we're back where we started...

One additional point to highlight for Q3. In parts (e) and (f), Brian asks you to evaluate the marginal products of capital and labor when they are at *their mean predicted values*. To be clear, this means you should find the predicted quantity $(\hat{Q})$ for each observation, given your optimally estimated parameters, and then find the mean of these predicted values. Use these mean values of $\hat{Q}$ for your hypotheses. In part (g), however, you are to test whether average MP_L equals average MP_K while averaging MPs *across all observations*. Again, to be clear, here you should should find the MPs for each observation, *and then* take the mean.

Finally, take care when constructing your data for Q3. As noted at the beginning of the question, *Labor used* is a function of the total labor force AND unemployed AND hours worked per week.

Moving on...

2 The likelihood function and joint probability

At the risk of getting a bit ahead of Brian's lectures, let's work through a very light discussion of maximum likelihood estimation (MLE), mostly getting a taste (or a refresher) of the joint probability theory that underlies this genre of estimation. If this doesn't make sense on first pass, hopefully it will make it easier when Brian goes through in more detail. If this is painfully obvious, then good!

Maximum likelihood represents another estimation technique (as opposed to least squares) to obtain parameter values. There are many ways to skin such a chicken; we began with nonlinear least squares because that is the immediate extension from the linear to the nonlinear modeling world. But now that we're thinking in a likelihood framework, instead of minimizing a given criterion like SSE or LAD[3], we maximize the likelihood that the sample we have would be actually generated by the data process. In order to do this, we must make decisions about what we think the distribution of the data looks like. But first...

[2] See, sometimes micro theory *is* actually useful. ;)

[3] Least absolute deviation - i.e. sum of the absolute values of errors.

2.1 *IID assumption*

Before getting into some examples, we should first note a basic assumption we make on the joint distribution of our random outcome variables, y_i. Suppose we have a sample of observations for some outcome variable. We assume the random variables are *independent and identically distributed*. This basically boils down to the following condition:

$$f(y_1, \ldots, y_n) = f(y_1)f(y_2)\cdots f(y_n) = \prod_{i=1}^{n} y_i$$

This means that the probability of observing y_1 and y_2 together are independent events. An example: in the case of a coin flip, the probability of observing two flips achieving heads using a fair coin is just:

$$f(H, H) = f(H)f(H) = 0.5 \times 0.5 = 0.25$$

However, if we assume that the probability of one event influences the probability of observing another event, this IID condition no longer holds. For instance, suppose we are concerned with the joint probability of it snowing outside and getting into a car accident. It seems pretty likely that the probability of a snow event will impact the probability of an accident, so we can no longer say that the joint distribution of these two random variables are independent. In particular, for an accident (y_1) during snow (y_2), we need to consider the *conditional distribution*:

$$f(y_1|y_2) = \frac{f(y_1, y_2)}{f(y_2)}$$

Where this expression essentially denotes the probability of an accident given that it has snowed. But since we're keeping it basic today, lets assume IID random variables, meaning that $f(y_1|y_2) = f(y_1)$, which in turn implies $f(y_1, y_2) = f(y_1)f(y_2)$.

2.2 *Likelihood vs log-likelihood*

The basic idea of maximum likelihood is that given some assumptions on the shape of our data, we want to maximize the joint probability of observing the data we have. To do so, assume that our outcome variable has a joint pdf $f(\mathbf{y}|\theta)$[4] Then, given that $\mathbf{Y} = \mathbf{y}$ is an observed data sample, the function of θ defined by $\mathcal{L}(\theta|\mathbf{y}) = f(\mathbf{y}|\theta)$ is called the *likelihood function*. Note the distinction between a pdf and a likelihood function: *in the latter, we consider* $\mathbf{y}$ *to be a fixed sample point (i.e. observed data!) and* θ *to be varying over the parameter space.*

So given this definition, we have a likelihood function that is represented by the product of n pdfs. As should be obvious by the estimator's name, the maximum likelihood estimator of the parameter θ based on a sample $\mathbf{Y}$ is $\hat{\theta}(\mathbf{Y})$: the parameter value at which the likelihood function $\mathcal{L}(\theta|y)$ attains its maximum.

If this likelihood function is differentiable in θ_i - think continuous distributions like Uniform, Gamma, Normal, etc., and some discrete distributions - then possible candidates for the MLE are the values of $(\theta_1, \ldots, \theta_n)$ that solve:

$$\frac{\partial}{\partial \theta_i}\mathcal{L}(\theta|\mathbf{y}) = 0$$

It's important to note that solutions to this equation are only *candidates* for the MLE; as usual, this first order condition is a necessary, but not sufficient condition for the global maximum. To confirm that any MLE is the global maximum, one must typically verify that the second-order condition holds negative, deal with the possibility of local maxima, as well as checking LE estimates at boundaries where the FOC may not be equal to zero for a maximum.

[4] Learn more about some common probability distribution functions here. Else, Chapter 3 of Casella and Berger's *Statistical Inference* is an concise but complete treatment aimed at graduate level students.

So this seems straightforward in theory, but it can quickly become complex when dealing with products of nonlinear probability distribution functions. A common way to ease this complexity is to *monotonically transform* our likelihood function. We often rely on the *log-likelihood* function, $\ell(\theta|\mathbf{y}) = ln(\mathcal{L}(\theta|\mathbf{y}))$. The log function is an ideal monotonic transform here: it preserves ordering and is strictly increasing on $(0, \infty)$ which implies that the extrema of $\mathcal{L}(\theta|\mathbf{y})$ and $\ell(\theta|\mathbf{y})$ coincide. Let's run through a very simple example from Greene (Chapter 14) that will make this more clear.

We consider an *iid* random sample $y_1, \ldots, y_n$ with a pdf given by:

$$f(y_i|\theta) = \frac{e^{-\theta}\theta^{y_i}}{y_i!}$$

As you might know, this is the pdf for a Poisson distribution. Then for n observed outcomes of y, our likelihood function is given by:

$$\mathcal{L}(\theta|\mathbf{y}) = f(y_1, \ldots, y_n|\theta) = \prod_{i=1}^{n} f(y_i|\theta) = \frac{e^{-n\theta}\theta^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!}$$

Okay, who wants to solve the first order condition for that?! Didn't think so - it's doable but we can make it easier. Take the log of both sides to obtain the log-likelihood function and we have:

$$\ell(\theta|\mathbf{y}) = -n\theta + \ln(\theta)\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \ln(y_i!)$$

Ah much better. Take the derivative with respect to our parameter θ to find the maximum:

$$\begin{aligned} \frac{\partial \ell(\theta|\mathbf{y})}{\partial \theta} &= -n + \frac{1}{\theta}\sum_{i=1}^{n} y_i = 0 \\ \Rightarrow \hat{\theta}_{ML} &= \frac{\sum_{i=1}^{n} y_i}{n} = \bar{y}_n \end{aligned}$$

So our MLE for θ in this simple exercise is just the average of our $\mathbf{y}$ sample. Greene plots out the likelihood and log-likelihood functions (see below) for a Poisson-distributed sample of size 10: $\mathbf{y} = \{5, 0, 1, 1, 0, 3, 2, 3, 4, 1\}$. If you calculate the MLE for this sample- you'll find $\hat{\theta} = 2$.

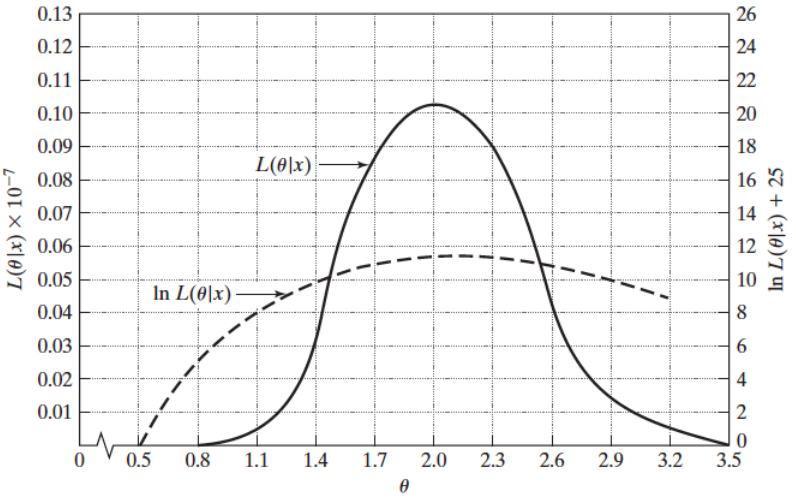

FIGURE 14.1 Likelihood and Log-Likelihood Functions for a Poisson Distribution.

2.3 *MLE versus LS*

As a final note, it's natural to be curious about the nature of the relationship between ML estimators and LS estimators. In fact - and I'm sure Brian will go through this more carefully in lecture - one can show that MLE and LS are identical for linear regressions with normality assumed. To whet your appetite, here's a quick sketch.

Recall that $\hat{\beta} = (X^{'}X)^{-1}X^{'}Y$ is the closed-form solution for an OLS estimator. Suppose our data is given by the model:

$$y = x^{'}\beta + \varepsilon$$

With $\mathbb{E}(y|x) = x^{'}\beta$ and variance, σ^2. Let's assume that y is normally distributed, with pdf:

$$f(y|x,\beta) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(y - x_i\beta)^2}{\sigma^2}\right)$$

Then our log-likelihood function is written as:

$$\ell(\beta|x,y) = \sum_{i=1}^{N}\ln\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(\frac{-(y - x_i\beta)^2}{\sigma^2}\right)\right] = -N\ln(\sqrt{2\pi\sigma^2}) - (\sigma^2)^{-1}\sum_{i=1}^{N}(y_i - x_i\beta)^2$$

Taking the first-order conditions of this log-likelihood function with respect to a particular β_k yields:

$$\frac{\partial\ell(\beta|x)}{\partial\beta_k} = \frac{1}{\sigma^2}\sum_{i=1}^{N}x_{ik}(y_i - x_{ik}\beta_k) = 0$$

Extending this in matrix notation, we finally can show that:

$$\frac{1}{\sigma^2}\left(X^{'}Y - X^{'}X\beta\right) = 0 \Rightarrow \hat{\beta}_{ML} = (X^{'}X)^{-1}X^{'}Y$$