Athens University of Economics and Business
Department of Economics

Postgraduate Program - Master's in Economic Theory
Course: Econometrics II
Prof: Stelios Arvanitis
TA: Alecos Papadopoulos

Semester: Spring 2017-2018

May 24, 2018

Implementing QMLE for ARMA models and the Newton-Raphson numerical algorithm

In this short note we show explicitly the twists and turns needed to implement maximum likelihood estimation for an ARMA model, including a quick mathematical exposition and geometric intuition for the Newton-Raphson numerical algorithm.

By calling the approach "QMLE"(quasi-MLE) we mean

- That we will implement a model where we will assume the normal density/likelihood function
- We will be modest enough to acknowledge that the true likelihood is... likely not normal.

The full technical apparatus needed to consider and reliably estimate ARMA models by maximum likelihood won't be repeated here (neither the regularity conditions for the MLE).

We will treat the ARMA(1,1) model for clarity, but the approach generalizes immediately to ARMA(p,q) models.

1. QMLE for ARMA models

We have a data sample $\{y_1,...,y_T\}$ and we assume the following ARMA (1,1) model for it:

$$y_t = ay_{t-1} + \theta v_{t-1} + v_t, \quad v_t \sim \text{i.i.d. } \mathrm{N}(0,1) \qquad [1]$$

Setting the variance of the error process equal to zero is a simplification to focus on the essence.

Following automatically the steps to implement maximum likelihood estimation, we would write

$$v_t = y_t - ay_{t-1} - \theta v_{t-1} \qquad [2]$$

and we would consider the joint density of the elements of the error process turned into the (average) log-likelihood of the sample (ignoring constants)

$$\ln L = -\frac{1}{2T}\sum_{t=1}^{T} v_t^2 = -\frac{1}{2T}\sum_{t=1}^{T}\left(y_t - ay_{t-1} - \theta v_{t-1}\right)^2$$

The problem we face is obvious: the log-likelihood does not contain only data and unknown parameters, but also the unobservable terms of the lagged error process. It cannot be computed or maximized as such. A second minor issue is that we cannot start the index of the sum at $t=1$ because we would be asking for data on $t=0$ also, and we don't have that either.

Let's first correct this small issue:

$$\ln L = \frac{-1}{2T}\sum_{t=2}^{T}\left(y_t - ay_{t-1} - \theta v_{t-1}\right)^2 = \frac{-1}{2T}\left(\left(y_T - ay_{T-1} - \theta v_{T-1}\right)^2 + \ldots\left(y_2 - ay_1 - \theta v_1\right)^2\right) \qquad [3]$$

Now for the big issue: how can we express the error elements in terms of data and unknown parameters? Writing [2] for $t=2$ and $t=3$ we note that

$$v_2 = y_2 - ay_1 - \theta v_1 \quad \text{and} \quad v_3 = y_3 - ay_2 - \theta v_2$$

So if we just impose the additional assumption $v_1 = 0$ we will get

$v_2 = y_2 - ay_1$ and $v_3 = y_3 - ay_2 - \theta v_2 = y_3 - ay_2 - \theta(y_2 - ay_1)$

e.t.c. With just this one additional assumption, we can write *all* error terms present in [3] in terms of available data and unknown parameters. Let's attempt to do that, writing first the oldest terms:

$$L = -\frac{1}{2T} \times \left[(y_2 - ay_1)^2 + (y_3 - ay_2 - \theta(y_2 - ay_1))^2 + (y_4 - ay_3 - \theta(y_3 - ay_2 - \theta(y_2 - ay_1)))^2 + \ldots \right] \quad [4]$$

Ok, it's getting very tedious and very complicated just to write it out explicitly, let alone take correct derivatives manually. Happily, we don't have to do it, we let the software do it for us.

The generalization to ARMA(p,q) is obvious: if the assumed MA lag-order is equal to q, we have to impose the assumption $v_1 = v_2 = \ldots = v_q = 0$.

Clearly, $v_1 = 0$ or $v_1 = v_2 = \ldots = v_q = 0$ are wrong conditions/assumptions. But in the weakly stationary framework we work in here, as long as we have a sample of some length, their effect will eventually wear out and won't really impact the quality of our inference.

2. The Newton-Raphson numerical algorithm

Something else that is clear from [4], is that, even if we managed to write explicitly the gradient of the log-likelihood (its first derivatives), we wouldn't obtain an analytical solution for the parameters under estimation (unknowns only on one side, data only on the other). If we don't get that, the software doesn't either.

What can we do? Just guessing values for the parameters ad hoc and calculating the value of the gradient to see whether it equals zero (or the value of the log-likelihood itself for that matter), is very inefficient and may take forever. So we need an approach where

we start from some perhaps arbitrary values, but then we have a way to make the next "guess" better (as in "closer to the argmax"), using the previous guess.

There are very many such "numerical calculation" algorithms, the oldest used in maximum likelihood being the "**Newton-Raphson algorithm**" (and many of the others being refinements, extensions, or similar in spirit to it). This is 17th century staff, in case you wonder.

2.1 Newton-Raphson for maximization problems

To connect clearly the algorithm to the maximization logic of the QMLE, we will assume that we have a single parameter to estimate, so we want to maximize over θ the log-likelihood $L(\theta)$ where the data sample has been suppressed in notation.

We first apply a 2nd-order Taylor expansion on the function (silently assuming that the log-likelihood has the properties that allow for such an expansion) around some perhaps arbitrarily chosen by us (and so known) value θ_1 :

$$L(\theta) = L(\theta_1) \; + \; L'(\theta_1)(\theta-\theta_1) \; + \; \tfrac{1}{2}L''(\theta_1)(\theta-\theta_1)^2 + R_2(\theta,\theta_1)$$

So taking the first-derivative of the left-hand side is equivalent to taking the same derivative on the right hand side. The only problem is the existence of the unkowable Taylor remainder $R_2(\theta,\theta_1)$. Since θ_1 is chosen without any guidance, don't think that $R_2(\theta,\theta_1)$ will be "small" - it may be really large.

The usual exposition here goes something like "ignore the remainder and take the first derivative on the rest", but we can provide a bit more intuition. Move the remainder to the left-hand side,

$$L(\theta) - R_2(\theta,\theta_1) = L(\theta_1) \; + \; L'(\theta_1)(\theta-\theta_1) \; + \; \tfrac{1}{2}L''(\theta_1)(\theta-\theta_1)^2 \qquad [5]$$

...and we see that by attempting to maximize the right hand side, *in reality we attempt to maximize the difference of the value of the log-likelihood from the Taylor remainder*:

$$\frac{\partial}{\partial\theta}\Big[L(\theta)-R_2(\theta,\theta_1)\Big]=\frac{\partial}{\partial\theta}\Big[L(\theta_1)\ +\ L'(\theta_1)(\theta-\theta_1)\ +\ \tfrac{1}{2}L''(\theta_1)(\theta-\theta_1)^2\Big]$$

Think about that. Maximizing this difference means that $R_2(\theta,\theta_1)$ gets small and/or that $L(\theta)$ gets large. *Either tendency is a desirable direction to take*, so it appears that we are onto something here. We then set the derivative equal to zero,

$$\frac{\partial}{\partial\theta}\Big[L(\theta_1)\ +\ L'(\theta_1)(\theta-\theta_1)\ +\ \tfrac{1}{2}L''(\theta_1)(\theta-\theta_1)^2\Big]=0\ \Rightarrow\ L'(\theta_1)+L''(\theta_1)(\theta-\theta_1)=0$$

$$\Rightarrow\ \theta^*=\theta_1-\frac{L'(\theta_1)}{L''(\theta_1)} \qquad [6]$$

Equation [6] gives us the optimal value for θ *given* the arbitrarily chosen θ_1. Informally, it is the best that we can do at this stage, given the constraint of our initial ignorance. So we do ***not*** expect that $\theta^*=\hat{\theta}_{MLE}$, i.e. that it is the actual argmax of the log-likelihood. But θ^* has a great advantage over θ_1: it is *optimally* chosen (under the ignorance constraint), not arbitrarily. And also, nobody said that we have to stop here. *Now,* instead of an arbitrarily chosen value for θ, we have a better value for it. So why not start over? So we name $\theta^*=\theta_2$ and we write again

$$L(\theta)-R_2(\theta,\theta_2)=L(\theta_2)\ +\ L'(\theta_2)(\theta-\theta_2)\ +\ \tfrac{1}{2}L''(\theta_2)(\theta-\theta_2)^2$$

$$...\Rightarrow\ \theta_3=\theta_2-\frac{L'(\theta_2)}{L''(\theta_2)} \qquad [7]$$

So, θ_3 optimizes in relation to θ_2 that was the optimal choice given the arbitrary θ_1 ... we have now put two rounds of optimal choice between us and the initial arbitrary value. Next, a third round of optimization, etc. It sounds not naive to expect that such a procedure, after a finite number of steps, say m, will lead us to

$$\theta_m = \hat{\theta}_{MLE} : L'\left(\hat{\theta}_{MLE}\right) = 0 \qquad [8]$$

And indeed it can do that, subject of course to certain "nice" properties of the likelihood function. For example, from the above equation we see that we would not want the second derivative of the likelihood to become zero or very close to zero, since we want it in the denominator.

In any case, note that what we obtain is a "stationary" point of the function, a *candidate exrtremum*, not necessarily a maximum, and not necessarily the *global* maximum.

It is in order to have assurance that we are hunting a maximum that we usually require the log-likelihood to be a concave function (in the parameters).

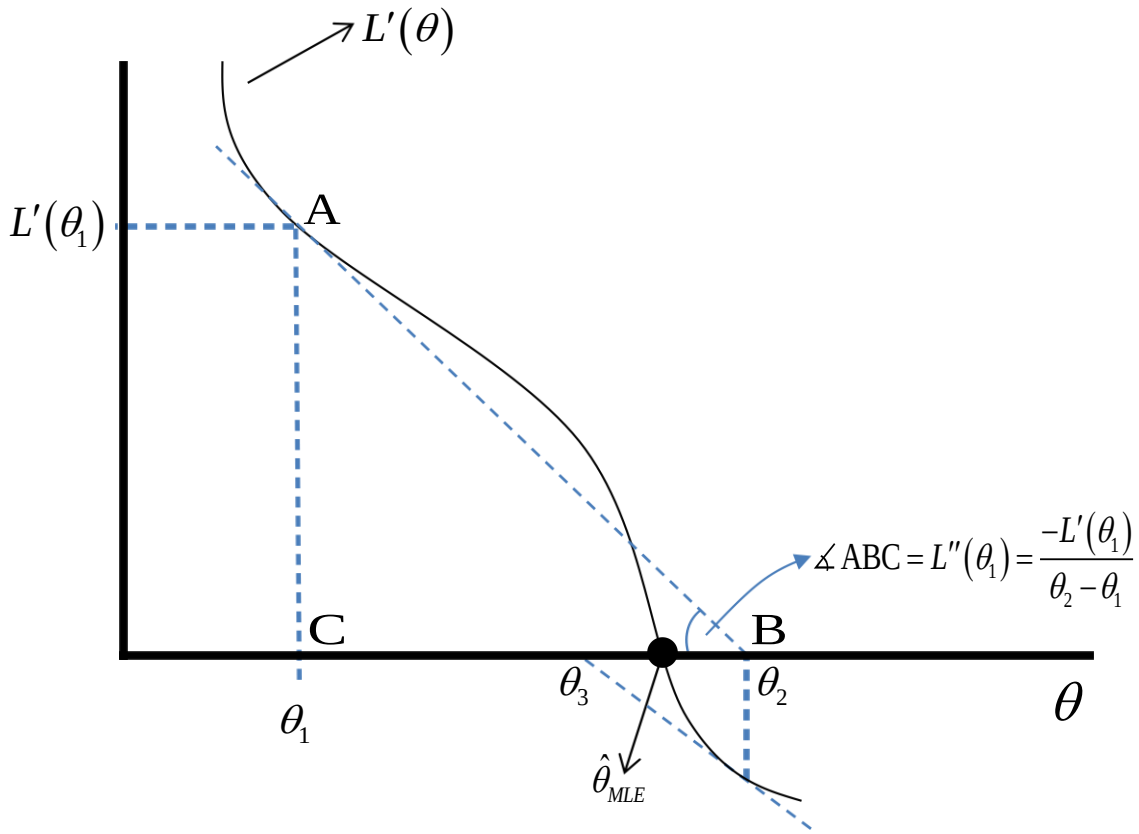
Still, if the likelihood is not monotonic in the unknown parameter, it may be the case, that by unfortunate choice of the initial θ_1 , we are lead by the algorithm to a local maximum, not the global one.

This is why it is mandatory for good empirical research to estimate a model using more than one (and more than two) "starting values" for the numerical algorithms, i.e different values for θ_1 in our case. Additionally, we should exploit any out of sample information to infuse these initial guesses with some realism (say, that they most likely be "smaller than unity" for example, given the data, the model, the real world situation we examine, etc). Sometimes we estimate them through some other estimation procedure like OLS.

The reason why many numerical algorithms exist, is that we have not found one that always converges to where it is supposed to, and does so more efficiently than any other. So the Newton-Raphson algorithm may fail in certain cases. Usually, statistical and econometric software offer various alternative algorithms to choose from.

2.2 Newton-Raphson - geometrical intuition

We now present a diagram that shows the geometric logic of the Newton-Raphson algorithm.



The descending line is the graph of the gradient of the log-likelihood, and we want to arrive at the black dot and $\hat{\theta}_{MLE}$ that gives $L'\left(\hat{\theta}_{MLE}\right) = 0$.

We start at an arbitrary θ_1. We find the corresponding point in the graph of $L'(\theta_1)$ (point A) and draw the tangent to it until we cross the horizontal axis (point B). The angle ABC equals the value of the derivative of $L'(\theta_1), L''(\theta_1)$. At the same time, it is equal to the ratio of the height $\text{AC} = L'(\theta_1)$ divided by the length $\text{BC} = \theta_2 - \theta_1$ (and the minus sign since the slope is negative). Rearranging the expression shown in the diagram, we obtain

$$\theta_2 = \theta_1 - \frac{L'(\theta_1)}{L''(\theta_1)}$$

But this is exactly what we will obtain if we maximize with respect to θ the Taylor expansion of $L'(\theta)$ around θ_1 (without the remainder). We then start from θ_2, find the corresponding point of $L'(\theta_2)$, draw the tangent to get θ_3, etc. We see that we tend to get closer to $\hat{\theta}_{MLE}$. --