LONG MEMORY IN ECONOMICS: A COMMENT

Joann JASIAK
York University, Dept of Economics
4700 Keele Street, Toronto Ont. M3J 1P3, CANADA
e-mail: jasiakj@yorku.ca

The paper by Lardic and Mignon provides an extensive overview of literature on long memory models, their applications and estimation methods. The authors have given considerable attention to various empirical papers documenting long memory in economic and financial data. A common approach to preliminary assessment of long memory in the data relies on the principle that "...the presence of long memory [in the series of stock returns] is indicated by significant autocorrelations at long lags..." [see section 5.1.2.1]. Accordingly, long memory is often inferred from a slow, hyperbolic decay pattern of the estimated autocorrrelation function (ACF henceforth). However the behavior of an empirical ACF may be determined, to some extent, by factors other than a highly persistent linear structure of the dynamics. For example, there is a growing concern about spurious long memory effects due to nonlinearities [see the comment by C. Gourieroux, and references therein]. In contrast, there seems to be very little awareness about the quality of commonly used autocorrelation estimators. I will address these two issues in my comment and show some numerical results providing insights on technical problems in detecting long memory.

1 The strong versus weak noise assumption

Many long memory models and estimation methods rely on the assumption of a strong white noise error term. Let us examine to what extent the ACF of a long memory process can be altered by relaxing the assumption of a strong white noise. For this purpose we design a simple simulation experiment in which we generate two trajectories of the process $\{y_t\}$, where:

$$y_t = (1-L)^d \epsilon_t^*.$$

In the first experiment the process is generated from an i.i.d. standard normal sequence $\{\epsilon_t\}$ representing a strong white noise process, $\epsilon_t^* = \epsilon_t$. Thus our first sample contains realizations of $y_t = (1-L)^d \epsilon_t$.

In the second experiment we transform the strong white noise $\{\epsilon_t\}$ into a weak white noise $\{\tilde{\epsilon}_t\}$ by inducing serial correlation in the squares. The weak noise is defined as $\tilde{\epsilon}_t = \sigma_t \epsilon_t$, where the conditional variance $\sigma_t^2 = E_{t-1}(\epsilon_t^2)$ is modelled as:

$$\sigma_t^2 = 0.1 + 0.5\epsilon_{t-1}^2.$$

We generate the second sample by setting $\epsilon_t^* = \tilde{\epsilon}_t$ and computing $y_t = (1-L)^d \tilde{\epsilon}_t$.

Both noise processes ϵ_t and $\tilde{\epsilon}_t$ used to generate y_t were adjusted to have equal marginal means and variances.

We set the fractional integration parameter d equal to 0.45, and truncate the infinite polynomial approximation at 3000 terms. In each sample, the first 8000 simulated realizations of $\{y_t\}$ out of the initial 9000 are discarded, so that two samples of length 1000 are used for further analysis.

Figure 1 displays two autocorrelation functions up to lag 100, estimated by Splus. The solid line represents the ACF for the process generated from the strong noise, while the dotted line represents the ACF of the weak noise based process. Compared to the ACF of the strong process, the "weak" ACF takes larger values over short and intermediate lags, and decays at a slightly faster although hyperbolic rate up to, approximately, the lag 70. At high lags, the weak process shows a stable pattern of very persistent and low valued autocorrelations, often reported in financial data [1].

The numerical differences between the two ACF's at lags 1 to 10 can be observed from Table 1 given in the Appendix, where column 1 represents the "weak" ACF while column 2 gives the "strong" ACF.

The differences in ACF suggest that basic estimators of the coefficient of fractional integration d may not produce similar outcomes in both samples. In the first two columns of Table 2 shown in the Appendix we report the estimates of d, (along with standard errors in parentheses) obtained from the following two methods.

The first one is the Porter-Hudak method [Geweke, Porter-Hudak (1983)], where we estimate the periodogram from a Fourier transform of autocorrelations, as most software packages do. We use 100 autocorrelations, and retain 20 first components of the periodogram for the final regression.

[1] Note that this experiment is not intended to produce rigorous results. More replications are necessary to draw final conclusions.

The second method [referred to as "Granger regression"] [Ding, Engle, Granger (1993)] consists in estimating d from a regression of log autocorrelations on a constant and a logarithm of the lag. It is valid for large lags only. The value of d is obtained by adding 0.5 to the regression coefficient on the lag logarithm.

We see that the Porter Hudak method underestimates the true d in the weak noise model, while the estimate is quite accurate in the strong noise model. The regression method performs correctly in the strong noise model as well, and fails totally in the weak noise model. It seems that a regression on lags higher than 100 would provide a better result.

2 Accuracy of ACF estimators

We now extend our analysis to examine the performance of autocorrelation estimators included in commonly used software packages for time series analysis.

We considered the following programs:

1) S-PLUS : Copyright (c) 1988, 1999 MathSoft, Inc.,Version 5.1 Release 1 for IBM RS/6000, AIX 4.3.1 : 1999

2) SAS Copyright (c) 1989-1996 by SAS Institute Inc., Cary, NC, USA. Proprietary Software Release 6.12 TS020.

3) GAUSS Version 3.2.40 (Jun 1 1998) (C) Copyright 1984-1997 Aptech Systems, Inc. Maple Valley, WA

4) TSP version 4.3A, Copyright (c) 1995 TSP International,

5) RATS version 4.2 WinRats, run on a DELL celeron PC.

We estimated the first 100 autocorrelations from the sample of 1000 simulated realizations of the strong noise based long memory model. Figure 2 displays the estimated ACF's. Three distinct autocorrelation patterns seem to emerge from this experiment. The autocorrelation function with highest values, plotted by a dotted line is provided by GAUSS. The one below, indicated by a solid line represents the ACF produced by Splus. The ACF's estimated from SAS, TSP and RATS overlap in this figure and give the lowest (dashed) curve of ACF, admitting negative values, in contrast to Splus and GAUSS.

We can observe the numerical differences in autocorrelations at lags 1 through 10 again in Table 2 by comparing the results in columns $2:6$.

The distinct autocorrelation patterns can now be used to examine the sensitivity of the two aforementioned d estimators. The results are reported in Table 2, columns 2 to 6.

Let us first consider the Porter-Hudak estimator. As expected the estimates of d from autocorrelations computed by SAS, TSP and RATS are very close. They all are slightly overestimated.

They also differ from the d estimate by Splus (closest to the true value), and the one by GAUSS, which slightly underestimates the true value.

The Granger regression can not be applied to neither SAS,TSP, nor RATS autocorrelations due to the presence of many negative autocorrelations at long lags. It gives a quite accurate result for Splus, and a slightly worse result for GAUSS.

It might be insightful at this point to explore one of the potential sources of ambiguity in computing autocorrelations. Anybody who has ever attempted to write his own code for computing an autocorrelation estimator at lag k has probably faced the following dilemmas:

1) Should T (the sample size) or rather $(T-k)$ be used as the divisor in the numerator? Or, in other words, should the small sample or the asymptotic theory be followed?

2) Should the variance in the denominator be computed using T or $T-1$ as divisor? Or should a product of two standard deviations, one for y_t and one for y_{t-k} be computed instead. If yes, should T, $T-k$, or $T-k-1$ be used as divisors? Where, when and why?

It appears that depending on the purpose of research, and personal needs, various permutations of the above could help to reach the desired conclusion on long memory. One can artificially "lift up" the autocorrelations, creating a stronger persistence, or do the opposite to reduce the memory. Obviously this works to some extent only. The choice of estimator can have however a very strong effect on the autocorrelations at large lags.

References

[1] Ding, Z., Engle, R., and C. Granger (1993): "A Long Memory Property of Stock Market Returns and a New Model", Journal of Empirical Finance, 1,83-106.

[2] Geweke, J., and S. Porter-Hudak (1983): "The Estimation and Application of Long Memory Time Series Models", Journal of Time Series Analysis, 4, 221-38.

[3] Gourieroux, C., and J. Jasiak (1999): "Memory and Infrequent Breaks", Economics Letters, forthcoming.

[4] Gourieroux, C., and J. Jasiak (1998): "Nonlinear Autocorrelograms: An Application to Inter-trade Durations", Journal of Time Series Analysis, forthcoming.

APPENDIX

Table 1: Autocorrelations for lags 1 to 10

WEAK NOISE	STRONG NOISE				
Splus	Splus	SAS	GAUSS	TSP	RATS
0.6767	0.6262	0.5734	0.6647	0.5730	0.5734
0.5848	0.4945	0.4497	0.5674	0.4500	0.4497
0.5238	0.4477	0.3912	0.5215	0.3910	0.3912
0.5125	0.4170	0.3415	0.4823	0.3420	0.3415
0.4892	0.3780	0.2910	0.4419	0.2910	0.2910
0.4657	0.3349	0.2253	0.3899	0.2250	0.2253
0.4223	0.2969	0.2260	0.3901	0.2260	0.2260
0.4502	0.2841	0.1727	0.3483	0.1730	0.1727
0.4492	0.3118	0.1670	0.3430	0.1670	0.1670
0.4197	0.2873	0.1984	0.3676	0.1980	0.1984

Table 2: Estimation of d

WEAK NOISE	STRONG NOISE				
Splus	Splus	SAS	GAUSS	TSP	RATS
Porter-Hudak					
0.3635 (0.0436)	0.4458 (0.0543)	0.4874 (0.0590)	0.4143 (0.0437)	0.4878 (0.0590)	0.4873 (0.0590)
Granger regression					
aberr.	0.4650 (0.1123) lag 60:100	N.A.	0.4936 (0.0659) lag 40:100	N.A.	N.A.

Figure 1: ACF estimates for the strong and weak y(t)

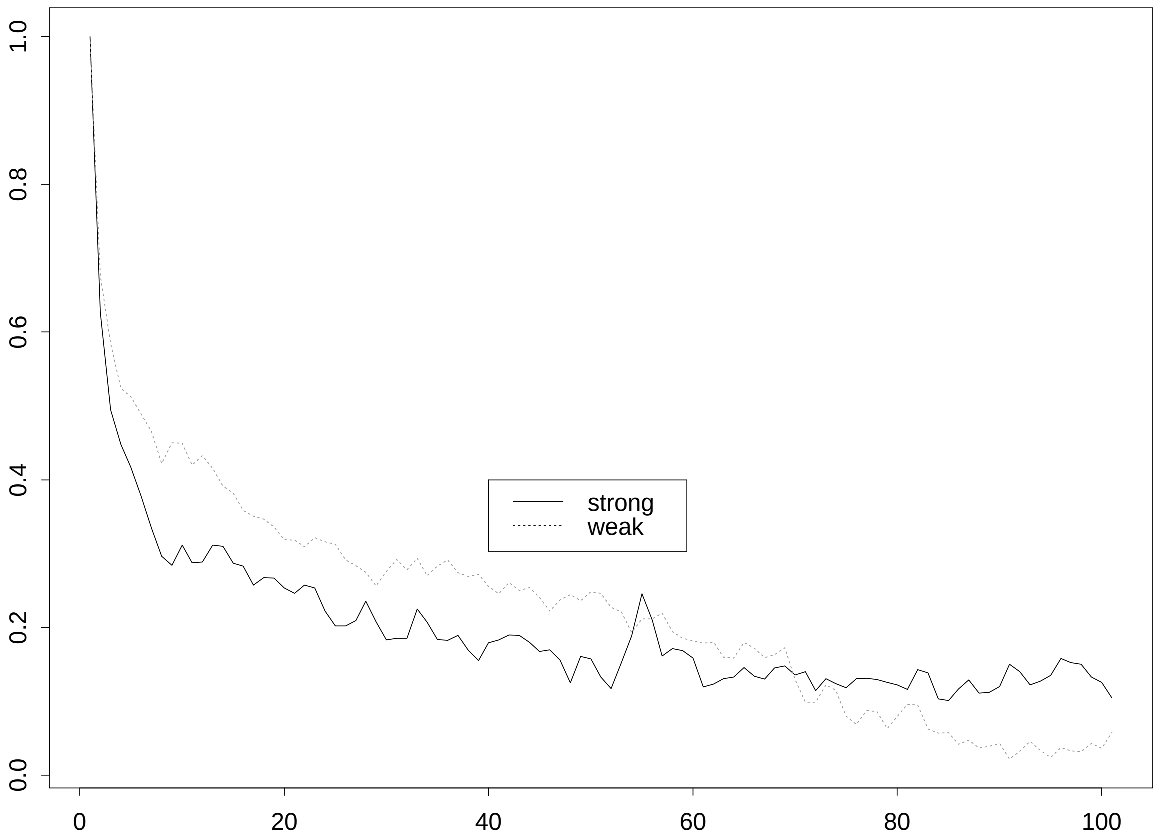

Figure 2: ACF estimates from software packages