NÚMERO 157

David Mayer-Foulkes

A GENERALIZED FAST ALGORITHM FOR BDS-TYPE STATISTICS

Resumen

Describimos un algoritmo rápido para caluclar el histograma de *m*-distancias en que se basan los estadísticos tipo BDS. El algoritmo generaliza uno que implementa LeBaron, calculando el histograma para cualquier conjunto finito de distancias simultáneamente. Reordenando el cálculo apropiadamente, el algoritmo utiliza menos memoria. Los dos algoritmos son comparados utilizando la implementación de LeBaron's en lenguaje C para MSDOS y la nuestra en Delphi (Pascal para Windows). El algoritmo generalizado es más rápido cuando se requieren más de dos valores de epsilon (el parámetro de distancia), y está implementado para calcular 255 distancias utilizando aritmética de enteros cortos..

Abstract

We provide a fast algorithm to calculate the m-dimensional distance histogram on which BDS-type statistics are based. The algorithm generalizes a fast algorithm due to LeBaron by calculating the histogram for any finite set of distances simultaneously. By reordering the calculation appropriately, the algorithm also requires less memory. The two algorithms are compared using LeBaron's MSDOS implementation in C and our Delphi (Windows Pascal) program. The generalized algorithm is faster when more than one value of epsilon (the distance parameter) is required, and is set up to calculate up to 255 values using short integer arithmetic.

Introduction

The interest in statistics capable of detecting non-linear dynamics is now well established in economics. Developing Grassberger and Procaccia's (G&P) (1983) Correlation Dimension (CD), Brock (1986a, 1986b), Brock et al. (1996), defined the BDS statistic testing the IID null whose applications include testing for non-linearity in stochastic processes. In contrast to these statistics, which calculate distances for all pairs of data points and which we call order 2 statistics, Mizrach (1991) defined the Simple Non-parametric Test (SNT), which only calculates distances from each data point to some fixed value such as the mean. These U-statistics involve much less calculation; we call them order 1 statistics and they can be applied for the same purposes. Combining the CD and BDS statistics Mayer (1995, 1996) defined the Correlation Dimension Ratio (CDR) (or Statistical Correlation Dimension), a statistic which tests the IID null, calculates dimensions greater than 1, and eliminates a downward bias present in the G&P and CD statistics. In further work, Mayer (1998) defines homogenized integral U-Statistics of orders 1 and 2, which use the same dimensional information that all of these statistics are based on.

The purpose of this paper is to describe the algorithm by which this basic dimensional information of order 2, BDS-type statistics is calculated by Mayer (1995, 1996, 1998). LeBaron (1997) describes the algorithms used to calculate the BDS statistic in computer programs first written by W. D. Dechert for DOS-based computer programs and then in C by LeBaron (see W. D. Dechert's Web page for the code and more information). The algorithm we present is a generalization which runs faster when the calculation is required for several values of ε, simplifies to LeBaron's for a single value of ε, and uses less memory. However, we do not calculate the significance of the BDS and related statistics as it is obtained by approximating to the normal distribution, for two reasons. First, the approximation to the normal distribution is not very good for the data sets usually used in economics. Second, the generalizations we work with have even more complicated expressions for this approximation, and demand even more calculation. Instead, we use bootstrap methods which have been stated in Mayer (1995, 1996, 1998).

The algorithm is implemented in a Windows user-friendly computer program together with a series of generalizations of the BDS statistic and the Simple Non-parametric Test (for which confidence intervals are calculated by bootstrapping), which is available from the author upon request[1].

The building block random variables

The order 2 statistics mentioned above are defined on the basis of some basic random variables which we now define for time series. Let Z^p, $p = 1, ..., N$ be N copies of a multivariate random variable Z. Define an m-history with lags of length τ by $\mathbf{z}(m)_i = (z_i, z_{i-\tau}, ..., z_{i-(m-1)\tau})$. For this to be well defined we need the index i to be

[1] Any request should be directed to mayerfou@dis1.cide.mx .

in the set

$$J(m,N) = \{i : (m-1)\tau + 1 \leq i \leq N\}. \tag{1}$$

Now the set of m-histories is

$$H(Z,m,N) = \{(\mathbf{z}(m)_i : i \in J(m,N)\}. \tag{2}$$

Let $J^2(Z,m)$ be the upper triangle of the Cartesian product,

$$J^2(m,N) = \{(i,j) \in J(m,N) \times J(m,N) : i < j\}. \tag{3}$$

Let I be the indicator function,

$$I(x,y) = \begin{cases} 1 & x \leq y, \\ 0 & x > y. \end{cases} \tag{4}$$

and write

$$b^m_{i,j}(\varepsilon) = I(\max_{1 \leq k \leq m} \left| z_{i-(k-1)\tau} - z_{j-(k-1)\tau} \right|, \varepsilon). \tag{5}$$

Then the "building block" random variable for BDS-type statistics is

$$C(m,\varepsilon,N) = \frac{1}{\#(J^2(m,N))} \sum_{(i,j) \in J^2(m,N)} b^m_{i,j}(\varepsilon). \tag{6}$$

This random variable can be used to define a whole family of statistics. The first example was the BDS statistic

$$BDS(\varepsilon,N) = C(m,\varepsilon,N) - C(1,\varepsilon,N)^m. \tag{7}$$

This can be modified, for example, to the Ratio Statistic (RS statistic)

$$RS(\varepsilon,N) = C(m,\varepsilon,N)/C(1,\varepsilon,N)^m \tag{8}$$

used by Mayer (1995, 1996). The (non-local) Correlation Dimension (CD) defined by Grassberger Procaccia is given by the limit of

$$CD(\varepsilon,N) = \ln(C(m,\varepsilon,N))/\ln(\varepsilon). \tag{9}$$

as $N \to \infty$ and $\varepsilon \to 0$. The (non-local) Correlation Dimension Ratio (CDR) studied by Mayer (1995) is instead the limit of

$$CDR(\varepsilon,N) = \ln(C(m,\varepsilon,N))/\left[m\ln(C(1,\varepsilon,N))\right]. \tag{10}$$

In practice the GP and CDR statistics are calculated as regressions of $C(m,\varepsilon,N)$ in terms of $C(1,\varepsilon,N)$ or $\ln(\varepsilon)$ for small values of ε . These are thus more complicated functions of the building block random variables. In Mayer (1998) we define homogenized integral U statistics. These first use a transformation of the data to the uniform or normal distribution and then calculate integrals with forms such as

$$\mathfrak{I}^j(N) = \int_{C(m,\varepsilon,N) \in [a,b]} f(C(m,\varepsilon,N), C(1,\varepsilon,N)^m) dC(m,\varepsilon,N). \tag{11}$$

That work includes comparative Monte Carlo tests of these and also the respective order 1 statistics calculated with the Windows program mentioned above.

This paper limits its scope to presenting the algorithm for calculating the building block random variables for order 2, $C(m,\varepsilon,N)$. Our interest in using many different values of ε simultaneously leads to an alternative algorithm (developed by Mayer in 1991) which calculates $C(m,\varepsilon,N)$ simultaneously for epsilon in a given discrete set.

The algorithm

To minimize the computational requirements in the calculation of $C(m,\varepsilon,N)$ the algorithm is recursive in m, uses mainly short-integer arithmetic, and has low memory requirements.

The algorithm calculates the sums in the definitions of $C(m,\varepsilon,N)$ (equation [6]) for a given number of dimensions $1 \leq m \leq M$ and for a given set of ε, $S_\varepsilon = \{\varepsilon_k : k = 0,...,K\}$. Any monotonically increasing formula for ε_k could be used. In the actual program we use the sets $S^K = \{\frac{k}{K+1}\varepsilon_{\max} : k = 0,...,K\}$ with $\varepsilon_{\max} = \theta \max\{|z_i - z_j| : 1 \leq i \neq j \leq N\}$, where θ is chosen by the user. In the experiments reported below $\theta = 1$. By setting $K \leq 255$ the algorithm uses mainly short-integers.

The sums in the definitions of $C(m,\varepsilon,N)$ range through the sets $J^2(Z,m)$. The algorithm covers this sets using a change of variables $(i,j) = (i,i+d)$, with $2 \leq d \leq N$, $1 \leq i \leq N-d$. The computer calculation uses loops in d,m,i, with $2 \leq d \leq N$, $1 \leq m \leq M$ and $(m-1)\tau + 1 \leq i \leq N-d$.

For each d, when $m = 1$ a vector $\mathbf{v}^1 = (v_1^1,...,v_{N-1}^1)$ is initialized setting integer entries

$$v_i^1 = \begin{cases} k & \text{if } \varepsilon_{k-1} < |z_i - z_{i+d}| \leq \varepsilon_k \text{ and } k > 0 \\ 0 & |z_i \quad z_{i+d}| \leq \varepsilon_0 \end{cases} \tag{12}$$

for $1 \leq i \leq N-d$ is formed. Now v_i^m is calculated iteratively in $m > 1$, setting for each i

$$v_i^m = \max\{v_i^{m-1}, v_{i-\tau}^{m-1}\},\ (m-1)\tau + 1 \leq i \leq N-d. \tag{13}$$

This operation uses only small integers when $K \leq 255$, and in the computer v_i^{m-1} replaces v_i^m, thus reducing memory requirements (the relevant part of the vector $\mathbf{v}^m$ gets shorter). Observe that $v_i^m = k$ iff $b_{i,i+d}^m(\varepsilon_k) = 1$ and $b_{i,i+d}^m(\varepsilon_{k-1}) = 0$. This follows inductively on m since it holds by construction for $m = 1$ and

$$b_{i,i+d}^{m+1}(\varepsilon) = \max\left[b_{i,i+d}^m(\varepsilon), b_{i-\tau,i-\tau+d}^m(\varepsilon)\right]. \tag{14}$$

As v_i^m are calculated through the loops, the sums

$$S(m,k) = \sum_{1 \leq d \leq N-1} \#(\{i : v_i^m = k\}) \tag{15}$$

are formed for $1 \leq m \leq M$, $0 \leq k \leq K$. By construction $S(m,k)$ is the frequency distribution of the m-dimensional distances lying in the interval $(\varepsilon_{k-1},\varepsilon_k]$, which can be written

$$S(m,k) = \#(\{(i,i+d) \in J^2(1,N) : \|\mathbf{z}(m)_{i+d} - \mathbf{z}(m)_i\| \in (\varepsilon_{k-1},\varepsilon_k]\}), \tag{16}$$

where $\|\cdot\|$ is the maximum norm. Once the loops in d,i,m have run, $C(m,\varepsilon_k,N)$ is found from the normalized cumulative distribution,

$$C(m,\varepsilon_k,N) = \frac{1}{\#(J^2(m,N))} \sum_{0 \leq l \leq k} S(m,l). \tag{17}$$

Two features distinguish this algorithm from the one presented in LeBaron (1997). The first is that we calculate $C(m,\varepsilon,N)$ for a set of values of ε simultaneously. The algorithms are equivalent for $K = 0$, in which case the operation in equation (13) can be considered to be logical instead of "arithmetic" (it is actually an if operation using

small integers). The second is that we reduce the memory requirements by calculating along diagonals. This feature could also be included in LeBaron's case $K = 0$. Something which we do not include is the sorting used by LeBaron before initializing $\mathbf{v}^1$ (see equation [12]), which serves the purpose of limiting the calculations to data points in an interval of values given by ε, because usually we are interested in the full range of values.

Experimental comparison of the two implementations

The comparison we make between the C language implementation of the algorithm described in LeBaron (1997) (available from Dechert's web page) and our own is not direct because the first uses sorting, while the program we use (in Delphi) does not, and besides carries some Windows overhead. (The calculation proceeds as a separate process [launching a thread] and there is a window showing the advance of the calculation. By the way, N is only limited by machine size and operating speed). We applied both algorithms to a uniform distribution on $[0, 1]$. LeBaron's algorithm was calculated for $\varepsilon = 1/2$ and $\varepsilon = 1$. Although this last value is not usually of interest in applications, here it has the effect of overriding the advantages of the sorting. The results are shown in Table 1.[2]

As expected, a single calculation is somewhat faster in LeBaron's implementation for large N, when sorting is disabled by choosing $\varepsilon = 1$. The run-time of our Windows program, which is almost independent of the number of values of ε used (up to 255) takes less than 2 LeBaron calculations, exceeding the length of a single calculation by approximately 40%. When sorting is useful, as for $\varepsilon = 1/2$, LeBaron's calculation is speeded by a factor of 6 to 9 times in these examples, so that our run time is still faster if the calculation is required for as many values of ε. LeBaron's implementation does not appear faster for $\varepsilon = 1$ and $N = 1000$, since sorting still takes place, although it is not useful.

Table 1. Comparison of the times for the calculation of $C(32, \varepsilon, N)$ (seconds)

Algorithm and values of ε	$N = 7500$	$N = 1000$
LeBaron, $\varepsilon = 1/2$	9.3	0.16
LeBaron, $\varepsilon = 1$	61.4	1.05
Mayer, $\varepsilon \in S^{10}$	83	0.95
Mayer, $\varepsilon \in S^{255}$	86	1.05

Conclusion

We provide a fast algorithm to calculate BDS-type statistics which generalizes the one presented by LeBaron (1997) by calculating the basic m-dimensional histogram $C(m, \varepsilon, N)$

[2] The calculations were carried out in a Pentium 333Mhz, 128Mb Ram.

for any finite set of values of ε simultaneously. By reordering the calculation appropriately, the algorithm also requires less memory. The algorithm is in principle slightly slower for calculating $C(m,\varepsilon,N)$ for one value of ε, because it replaces some binary operations with short-integer 'if's, but it is faster when more values of ε are used. These results are confirmed when LeBaron's MSDOS implementation of his algorithm in C is compared to our implementation of the generalized algorithm in Delphi, when sorting in LeBaron's implementation is disabled. When sorting is useful, LeBaron's program runs faster, but does not catch up with the generalized algorithm if the calculation is required for enough values of ε.

References

[1] Barnet, W. A., Gallant, R., Hinich, M. J., Jungeilges, J. A., Kaplan, D. T., and Jensen, M. J. (1996), "An experimental design to compare tests of nonlinearity and chaos", Chapter 6, *Nonlinear Dynamics and Economics*, Proceedings of the Tenth International Symposium in Economic Theory and Econometrics, Edited by W. A. Barnett, Alan P. Kirman and Mark Salmon, Cambridge University Press.

[2] Barnet, W. A., Gallant, R., Hinich, M. J., Jungeilges, J. A., Kaplan, D. T., and Jensen, M. J. (1997), "A single blind controlled competition among tests for nonlinearity and chaos", *Journal of Econometrics*, 82, 157-192.

[3] LeBaron, Blake (1997) "A Fast Algorithm for the BDS Statistic", Studies in Nonlinear Dynamics and Econometrics, July, 1997, 2(2): 53-59.

[4] Brock, W. A. (1986a), "Distinguishing Random and Deterministic Systems: Abridged Version", *Journal of Economic Theory* 40 168-195.

[5] Brock, W. A. (1986b), "Theorems on Distinguishing Deterministic from Random Systems", *Dynamic Econometric Modelling, Proceedings of the third International Symposium in Economic Theory and Econometrics* (Edited by W. A. Barnett, E. R. Berndt and H. White), 247-265, Cambridge University Press, New York.

[6] Brock, W. A.; Dechert, W. D.; Sheinkman, J.A. and LeBaron, B. (1996), "A Test for Independence Based on the Correlation Dimension", *Econometric Reviews* 15(3): 197-235.

[7] De Lima, P. J. F. (1996), "Nuisance Parameter Free Properties of Correlation Integral Based Statistics", *Econometric Reviews* 15(3), 237-259.

[8] Denker, M. and G. Keller (1983), "On U-statistics and V. Misses Statistics for Weakly Dependent Processes", *Zeitschrift fur Wahrscheinlichkeits-theorie und Verwandte Gebiete*, 64, 505-522.

[9] Grassberger, Peter and Itamar Procaccia (1983), "Measuring the strangeness of strange attractors", *Physica 9D*, 189-208.

[10] Mayer-Foulkes, D. (1995), "A Statistical Correlation Dimension", *Journal of Empirical Finance* 2 277-293.

[11] Mayer-Foulkes, D., and Raúl Anibal Feliz (1996), "Nonlinear dynamics in the stock exchange", *Revista de Análisis Económico* Vol. 11, No 1, pp 3-21.

[12] Mayer-Foulkes, D. (1998), "Homogenized Integral U-Statistics for Test of Non-

Linearity", Documento de Trabajo del CIDE, División de Economía, N° 118.
[13] Mizrach, B. (1991). "A simple Nonparametric test for independence." Working paper, Department of Finance, the Wharton School.
[14] Mizrach, B. (1992). "Multivariance nearest-neighbour Forecasts of EMS Exchange Rates." *Journal of Applied Econometrics* 7; Supplement, S151-63.
[15] Mizrach, B. (1994). "Using U-statistics to detect business cycle non-linearities." Chapter 14 in Willi Semmler (ed) *Business Cycles: Theory and Empirical Investigation*, Boston, Kluwer Press, 107-29.
[16] Ramsey, J. B., C. L. Sayers and P. Rothman (1990), "The Statistical Properties of Dimension Calculations using Small Data Sets: Some Economic Applications", *International Economic Review*, Vol 31, No 4.
[17] Serfling, R. J. (1980), *Approximation Theorems of Mathematical Statistics*, Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons.