CIDE

NÚMERO 189

David Mayer

A SORTING-ASSISTED FAST ALGORITHM FOR FRACTAL DIMENSION AND BDS-TYPE STATISTICS

Resumen

Presentamos un algoritmo rápido para calcular el histograma de distancias $C(m, \varepsilon)$ en el que se basan los estadísticos de tipo BDS, que se apoya en el ordenamiento de los datos. El algoritmo calcula el histograma simultáneamente para conjuntos de valores de m y ε, como el algoritmo rápido generalizado de Mayer (2000) sin perder mucha velocidad. Cuando el conjunto de valores de ε se limita a valores pequeños, el algoritmo es de orden N. Implementamos también el algoritmo de cajas de orden N de Grassberger (1990), incluyendo los casos $m = 1$ y $\tau > 1$. Utilizando corridas experimentales encontramos que el algoritmo basado en el ordenamiento rebasa en un orden de magnitud al de cajas cuando N toma valores en los cientos de miles.

Abstract

We present a sorting-assisted fast algorithm to calculate the distance histogram $C(m, \varepsilon)$ on which BDS-type statistics are based. The algorithm calculates $C(m, \varepsilon)$ for sets of values of m and ε simultaneously, as in Mayer's (2000) generalized fast algorithm, without loosing much speed. When the ε set has a small bound, the algorithm is order N. We also implement Grassberger's (1990) order N box-assisted algorithm including the cases $m = 1$ and $\tau > 1$. In experimental runs we find that the sorting-assisted overtakes the box-assisted algorithm by an order of magnitude for values of N in the hundred thousands.

Introduction

Achieving a fast calculation of Grassberger and Procaccia's (G&P) (1983) Correlation Dimension and related BDS-type statistics (Brock 1986a, 1986b; Brock et al. 1996; Mayer, 1995, 1996, 1998) promises to be useful in several areas, such as electronic experiments, EEG observations (Theiler, 1995), the analysis of seismic data, financial time series, etc. It has thus received the attention of several authors. Working to calculate the basic distance histogram $C(m,\varepsilon,N)$ (see the definition below) for small values of epsilon, Theiler (1987) and then Grassberger (1990) use box-assisted algorithms which improve on other techniques involving k-dimensional trees (Bingham, 1989), obtaining order-N algorithms. Working instead with less data to calculate the full histogram, as demanded by interest in economic time series, LeBaron (1997) introduces a fast algorithm including Theiler's (1990) suggestion to first sort the data. Mayer (2000) generalizes this algorithm to calculate the histogram for many dimensions simultaneously, but without using sorting. In this paper we introduce sorting and use some of the principles of the generalized fast algorithm to obtain a calculation of order N when $C(m,\varepsilon,N)$ is required for only small values of epsilon.. We compare the performance of this algorithm with a somewhat extended implementation of Grassberger's (1990) box-assisted algorithm, which obtains $C(m,\varepsilon,N)$ for all dimensions simultaneously, including $m=1$, and allows $\tau > 1$ (see below).

These last three algorithms are implemented in a Windows user-friendly computer program together with a series of generalizations of the BDS statistic and the Simple Non-parametric Test (Mizrach, 1991), for which confidence intervals are calculated by bootstrapping. The program is available from the author upon request[1].

The paper is organized as follows. In section 2 we write down the basic definitions.. In section 3 we describe the algorithm. In section 4 we describe our implementation of Grassberger's (1990) box-assisted algorithm. In section 5 we present the experimental results comparing the performance of these two algorithms and the generalized fast algorithm (Mayer, 2000). In section 6 we make some final remarks.

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 $Z \in \mathbb{R}^m$. 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 in the set

$$J(m,N) = \{i : (m-1)\tau + 1 \leq i \leq N\}. \tag{2.1}$$

Now the set of m-histories is

$$H(Z,m,N) = \{(\mathbf{z}(m)_i : i \in J(m,N)\}. \tag{2.2}$$

[1] Any request should be directed to mayerfou@dis1.cide.mx. The program can be downloaded at http://www.cide.edu/investigadores/David_M/HomePage.htm.

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

$$J^2(m,N) = \{(i,j) \in J(m,N) \times J(m,N) : i < j\}. \tag{2.3}$$

Let I be the indicator function,

$$I(x,y) = \begin{cases} 1 & x \leq y, \\ 0 & x > y. \end{cases} \tag{2.4}$$

Write $\|\mathbf{z}(m)_i - \mathbf{z}(m)_j\| = \max_{1 \leq k \leq m} \left| z_{i-(k-1)\tau} - z_{j-(k-1)\tau} \right|$ for the maximum norm and let

$$b_{i,j}^m(\varepsilon) = I(\|\mathbf{z}(m)_i - \mathbf{z}(m)_j\|, \varepsilon). \tag{2.5}$$

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

$$C(m,\varepsilon,N) = \frac{1}{\#(J^2(m,N))} \sum_{(i,j) \in J^2(m,N)} b_{i,j}^m(\varepsilon). \tag{2.6}$$

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

$$BDS(\varepsilon,N) = C(m,\varepsilon,N) - C(1,\varepsilon,N)^m. \tag{2.7}$$

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

$$RS(\varepsilon,N) = C(m,\varepsilon,N)/C(1,\varepsilon,N)^m \tag{2.8}$$

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

$$CD(\varepsilon,N) = \ln(C(m,\varepsilon,N))/\ln(\varepsilon). \tag{2.9}$$

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

$$CDR(\varepsilon,N) = \ln(C(m,\varepsilon,N))/[m\ln(C(1,\varepsilon,N))]. \tag{2.10}$$

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. In Mayer (1998) we define a whole series of other statistics based on these building block random variables.

The algorithm using sorting

The order N sorting-assisted fast algorithm we shall describe calculates $C(m,\varepsilon,N)$ simultaneously for

$$(m,\varepsilon) \in \{1,...,M\} \times \{\varepsilon_1,...,\varepsilon_K\}, \tag{1}$$

where $\varepsilon_1 < ... < \varepsilon_K = \varepsilon_{\max}$. The value $\varepsilon_{\max}$ is chosen to be relatively small, so that the number of neighboring pairs of Z-realizations satisfying $|z_i - z_j| \leq \varepsilon_{\max}$ is of order N. In the actual application we choose $K \leq 254$[2] so that most calculations are carried out with short integer arithmetic.

To give a simple explanation of the algorithm and to prove its properties we first introduce an equivalence class structure on the set of $\varepsilon_{\max}$-neighboring pairs (z_i, z_j).

[2] We use $K = 254$ instead of 255 because this saves on the application of some if statements in the main algorithms.

We refer to this set by defining the set of ordered pairs of indices

$$J_{\varepsilon_{\max}}(N) = \{(i,j) \subset J(1,N) \times J(1,N) : i < j \text{ and } |z_i - z_j| \leq \varepsilon_{\max}\}. \quad (2)$$

We shall say that two such ordered pairs (i_1, j_1), $(i_2, j_2) \in J_{\varepsilon_{\max}}(N)$ index *contiguous* ordered pairs (z_{i_1}, z_{j_1}), (z_{i_2}, z_{j_2}) if $i_1 - i_2 = j_1 - j_2 = \sigma\tau$, where σ is 1 or -1. This means that if we take any two m-histories $\mathbf{z}(m)_i$, $\mathbf{z}(m)_j$ containing the pairs z_{i_1}, z_{i_2}, and z_{j_1}, z_{j_2} in the same positions, the pairs will be next to each other, and that when the maximum norm $||\mathbf{z}(m)_i - \mathbf{z}(m)_j||$ is formed the contribution of these pairs is less than $\varepsilon_{\max}$. We now define an equivalence relation $\sim$ on $J_{\varepsilon_{\max}}(N)$ whose equivalence classes are the maximal chains of contiguous neighboring pairs. Define

$(i,j) \sim (i',j')$ if and only if one of the following holds:

1) $(i,j) = (i',j')$.

2) There exists a sequence (i_k, j_k), $k = 0, ..., n$ of elements of $J_{\varepsilon_{\max}}(N)$ such that $(i,j) = (i_0, j_0)$, $(i',j') = (i_n, j_n)$ and (i_{k-1}, j_{k-1}), (i_k, j_k), index contiguous ordered pairs for $k = 1, ..., K$.

The relation $\sim$ is reflexive by construction, and it is clearly symmetric and transitive, so it is an equivalence relation. It is clear that the equivalence classes are sets of indices in maximal chains of contiguity.. Each equivalence class is of the form

$$E = \{(i,j), (i-\tau, j-\tau), ..., (i-(\bar{m}-1)\tau, j-(\bar{m}-1)\tau)\} \quad (3)$$

and satisfies $i < j$, $(i-(m-1)\tau, j-(m-1)\tau) \in J_{\varepsilon_{\max}}(N)$ for $m = 1, ..., \bar{m}$, and $(i+\tau, j+\tau)$, $(i-\bar{m}\tau, j-\bar{m}\tau) \notin J_{\varepsilon_{\max}}(N)$. It is also clear that if two m-histories $\mathbf{z}(m)_s$, $\mathbf{z}(m)_t$ satisfy $||\mathbf{z}(m)_s - \mathbf{z}(m)_t|| \leq \varepsilon_{\max}$ then the sequence of pairs $\{(s,t), (s-\tau, t-\tau), ..., (s-(m-1)\tau, t-(m-1)\tau)\}$ indexing the differences taken in the calculation of the maximum norm $||\mathbf{z}(m)_s - \mathbf{z}(m)_t||$ is a subset of one of the equivalence classes E of $\sim$.

We now explain the construction of the order N sorting-assisted fast algorithm. Essentially, the algorithm proceeds in two steps. The first is to identify the equivalence classes E of $\sim$. To do this we use sorting to obtain an order N calculation. The second is to calculate the contribution of each equivalence class E to $C(m, \varepsilon_k, N)$ for $1 \leq m \leq M$, $1 \leq k \leq K$. To do this we follow ideas contained in Mayer's (2000) generalized fast algorithm.

Identifying the equivalence classes E

The first step of the algorithm is to sort the sequence $z_1^1, ..., z_N^1$. Here the upper index represents the first entry of vectors $z \in \mathbb{R}^w$.[3] We obtain indices $i(1), ..., i(N)$ for which $z_{i(1)}^1 \leq ... \leq z_{i(N)}^1$. We write $i(n)$ and $n(i)$ for the corresponding bijection and its inverse.

[3] Of course, any other entry could be chosen.

The second step of the algorithm is to find

$$Q(p) = \max\{q \geq 1 : z^1_{i(p+q)} - z^1_{i(p)} \leq \varepsilon_{\max}\}. \quad (4)$$

$$Q = \max\{Q(p) : 1 \leq p \leq N-1\}. \quad (5)$$

We assume that $\varepsilon_{\max}$ is not so small that Q does not exist, and that as N becomes large $\varepsilon_{\max}$ is chosen sufficiently small for Q to be bounded.[4] Thus the search carried out to find Q is of order N. Note that since the sequence $\{z_i\}$ has been sorted,

$$Q(p+1) \geq Q(p) - 1 \quad (6)$$

since $z^1_{i(p+Q(p))} - z^1_{i(p+1)} \leq z^1_{i(p+Q(p))} - z^1_{i(p)} \leq \varepsilon_{\max}$. This can be taken into account to shorten the calculation of Q.

Introduce the notation

$$[i,j] = (\min\{i,j\}, \max\{i,j\}). \quad (7)$$

The next step is to initialize an $(N-1) \times Q$ matrix $T = [T_{pq}]$ at zero. The matrix will mark with a 1 those pairs of indexes $[i(p), i(p+q)]$ of neighboring pairs of Z-realizations whose maximal equivalence class E has already been found.[5]

Now comes the main iterative procedure. For $p = 1$ to $N-1$ and $q = 1$ to $Q(p)$, we do the following. If $T_{pq} = 1$, go to the next (p,q). Otherwise, first determine if $[i(p), i(p+q)] \in J_{\varepsilon_{\max}}(N)$.[6] If the condition is met, find the maximal E to which $[i(p), i(p+q)]$ belongs. Now mark all those entries $T_{p'q'}$ of the table for which $[i(p'), i(p'+q')]$ is in E, by setting $T_{p'q'} = 1$, so that each equivalence class E is calculated only once. More precisely, using the notation for E in (3), for $m = 1, ..., \bar{m}$ we define

$$p(m) = \min\{n(i-(m-1)\tau), n(j-(m-1)\tau)\} \quad (8)$$

$$q(m) = \max\{n(i-(m-1)\tau), n(j-(m-1)\tau)\} - p(m), \quad (9)$$

and set $T_{p(m)q(m)} = 1$.

Since by this procedure each equivalence class E is identified exactly once, the number of times each distance $|z_i - z_j|$ is calculated is also reduced to one. Once each equivalence class E has been identified, its contribution to $C(m,\varepsilon,N)$ for each $(m,\varepsilon) \in \{1,...,M\} \times \{\varepsilon_1, ..., \varepsilon_K\}$ is calculated as described below.

Contribution to $C(m,\varepsilon,N)$ of each equivalence class E

First, there is a simple case which is worth mentioning. If $K = 1$ and only $C(m,\varepsilon_{\max},N)$ is to be calculated, the contribution of each equivalence class E to $C(m,\varepsilon_{\max},N)$ is $\bar{m}+1-m$ for $1 \leq m \leq \bar{m}$, where we use the notation in (3). Otherwise a more elaborate calculation, adapted from Mayer (2000) is needed, which

[4] This is an assumption which also implies that Grassberger's (1990) box-assisted algorithm is of order N.

[5] The implementation uses a vector containing T_{pq} for $1 \leq q \leq Q(p)$ only (and $1 \leq p \leq N-1$) so as to save memory space.

[6] In the case $m = 1$ a sufficient condition is $q \leq Q(p)$. In the general case, $Q(p)$ need not be kept in memory if $z^1_{i(p+q)} - z^1_{i(p)} \leq \varepsilon_{\max}$ is tested for again.

we now describe.

For $m = 1$ set vector $\mathbf{v}$ at $\mathbf{v}^1 = (v_1^1, ..., v_{\bar{m}}^1)$, where

$$v_s^1 = \begin{cases} k & \text{if } \varepsilon_{k-1} < \left|z_{i-(s-1)\tau} - z_{j-(s-1)\tau}\right| \leq \varepsilon_k \text{ and } k > 0 \\ 0 & \left|z_{i-(s-1)\tau} - z_{j-(s-1)\tau}\right| \leq \varepsilon_0 \end{cases} \tag{10}$$

for $1 \leq s \leq \bar{m}$. Vector $\mathbf{v}$ consists of short integers if $K \leq 255$ and the maximum length it can take is $N - 1$. This is the only memory space needed for this part of the calculation. In practice, the initial vector $\mathbf{v}^1$ may be calculated at the same time that each maximal chain of contiguity E is being determined.

Next, $\mathbf{v}^m$ is calculated iteratively for $1 < m \leq \bar{m}$, by setting

$$v_s^m = \max\{v_s^{m-1}, v_{s+1}^{m-1}\}, \ 1 \leq s \leq \bar{m} + 1 - m. \tag{11}$$

This is a short integer calculation when $K \leq 254$, and in the computer v_s^{m+1} replaces v_s^m (the relevant part of the vector $\mathbf{v}^m$ gets shorter). Observe that by construction

$$v_s^m = k \Leftrightarrow \varepsilon_{k-1} < ||\mathbf{z}(m)_{i-(s-1)\tau} - \mathbf{z}(m)_{j-(s-1)\tau}|| \leq \varepsilon_k$$

This follows inductively on m since it holds by construction for $m = 1$ and

$$||\mathbf{z}(m+1)_{i-(s-1)\tau} - \mathbf{z}(m+1)_{j-(s-1)\tau}|| \tag{12}$$

$$= \max\left[||\mathbf{z}(m)_{i-(s-1)\tau} - \mathbf{z}(m)_{j-(s-1)\tau}||, ||\mathbf{z}(m)_{i-s\tau} - \mathbf{z}(m)_{j-s\tau}||\right]. \tag{13}$$

As each v_s^m is obtained for each $s = 1$ to $\bar{m}+1-m$ and for each E, the number of times a distance index k has been obtained for each m is updated, adding one to a computer variable, so that by the time the algorithm has run for all E the sum

$$S(m,k) = \sum_E \#(\{s : v_s^m = k\}) \tag{14}$$

has been formed.

Observe that $\bar{m}$ is bounded by $Q + 1$. Thus the number of operations carried out for each E, which is of order $\bar{m}^2$, is bounded uniformly in N.

Write

$$J^2(m,N) = \{(i,j) \in J(m,N) \times J(m,N) : i < j\}. \tag{15}$$

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

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

$C(m,\varepsilon_k,N)$ is found from the normalized cumulative distribution,

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

Summary of the properties of the algorithm

The main advantages of the algorithm we have described are the following.

1) The algorithm uses a number of operations of order N.
2) The algorithm calculates each maximum norm $|z_i - z_j|$ intervening in the

maximum norms $||\mathbf{z}(m)_i \quad \mathbf{z}(m)_j||$ only once.

3) Much of the calculation, including the extension to dimensions $m > 1$, is carried out with short integer arithmetic when $K \leq 255$.

4) An even shorter calculation is available in the case $K = 1$.

A disadvantage of the algorithm is that it needs to establish the dimensions of the table T before the main calculation. This requires the calculation of the differences $z^1_{i(p+q)} - z^1_{i(p)}$. However, the number of calculations this involves is less than those in 2) because of property (6) above.

Implementation of the box-assisted fast algorithm

We now describe our implementation of Grassberger's (1990) optimized box-assisted algorithm. We shall calculate $C(m,\varepsilon,N)$ simultaneously for the same set of values of (m,ε) as before (see [1] above). Our implementation uses a two-dimensional grid of square boxes, extends the algorithm to the cases $\tau > 1$ (see the definition of the m-histories $\mathbf{z}(m)_i$), and includes the calculation of $C(1,\varepsilon,N)$, which in principle would only need a one-dimensional grid of boxes.

For the description of our implementation, let us suppose that z_i is normalized so that $0 \leq z_i \leq 1$.[7] The idea is to subdivide the region $[0,1] \times [0,1]$ of the plane $(z^1_i, z^1_{i-\tau})$ into a grid of square boxes so that when we check whether $|(z_i, z_{i-\tau}) - (z_j, z_{j-\tau})| \leq \varepsilon_{\max}$ we only do so for those pairs (i,j) for which $|(z^1_i, z^1_{i-\tau}) - (z^1_j, z^1_{j-\tau})| \leq \varepsilon_{\max}$. Recall that the superindex refers to the first coordinate of the multivariate z (any other given coordinate could be used). In contrast with Grassberger (1990), in our implementation τ intervenes in the second coordinate, in a manner consistent with the definition of the m-histories $\mathbf{z}(m)_i$. The optimal length of the side of each box is bounded below by $\varepsilon_{\max}$. For this size of grid we need only check those pairs $(z^1_j, z^1_{j-\tau})$ which lie in boxes adjacent to the box containing $(z^1_i, z^1_{i-\tau})$. If the boxes get smaller the algorithm becomes longer because the number of relevant boxes which need to be searched increases. If the boxes get bigger more pairs in the adjacent boxes may be checked than is necessary, but the array of boxes will be smaller so the length of the calcualtion may be reduced.

We assign to each pair $(z^1_i, z^1_{i-\tau})$ the box (p_i, q_i) given by $(\nu(i), \nu(i-\tau))$ where for $1 \leq i \leq N$

$$\nu(i) = \begin{cases} k & \text{if } i \leq M \text{ and } k-1 < z_i/n_{\text{box}} \leq k, \text{ with } k \geq 1 \\ 1 & \text{if } i \leq M \text{ and } z_i = 0 \text{ (a non-generic case)} \\ 0 & \text{if } i < 1 \text{ (so that } z_i \text{ does not exist)} \end{cases} \tag{18}$$

Here n_{box} is the number of boxes on each axis, which we define as follows:

$$n_{\text{box}} = \min\{\text{Trunc}(1/\varepsilon_{\max}), g(N), \bar{n}_{\text{box}}\}.$$

[7] We thus use bounds on z rather than work with a torus as in Grassberger (1990). This speeds the algorithm except for the calculation of the bounds themselves, which, however, are also used in the program for other purposes.

The bound $n_{\text{box}} = \text{Trunc}(1/\varepsilon_{\max})$ corresponds to the box size $\varepsilon_{\max}$. n_{box} is kept smaller than some maximum size $\bar{n}_{\text{box}}$ determined by memory requirements. It is also kept smaller than some $g(N)$. Grassberger (1990) recommends $g(N) = \sqrt{N}$, which is what we use. The length of the calculation is quite sensitive to the choice of n_{box}. Note also that the calculation of $C(1, \varepsilon, N)$ is of order $Ng(N)$. This concerns us in the case when N is relatively small, when the calculation may be repeated many times in the application of the reshuffling test. In practice we used $\bar{n}_{\text{box}}$ equal to 1000 or 8000. This implies that the grid requires 10^6 or 64×10^6 slots of memory each holding a long integer. The case $j > 1$ in (18) occurs if $i - \tau < 1$, and must be considered for the calculation of $C(1, \varepsilon, N)$.

The algorithm now proceeds as follows to form the sums $S(m, k)$ defined in equation (14). For each i from 1 to N we examine all previously considered $j < i$ for which the first box coordinates are adjacent, so $|p_j - p_i| \leq 1$, and add 1 to $S(1, k)$ if $\varepsilon_{k-1} < |z_i - z_j| \leq \varepsilon_k$. No other boxes need be considered for the calculation of $C(1, \varepsilon, N)$ since $|p_j - p_i| > 1$ implies $|z_i^1 - z_j^1| > \varepsilon_{\max}$ and therefore $|z_i - z_j| > \varepsilon_{\max}$. Next, for those pairs for which the condition $|z_i - z_j| \leq \varepsilon_{\max}$ held and for which also the second box coordinate is adjacent, so $|q_j - q_i| \leq 1$, we determine sequentially for $2 \leq m \leq M$ whether $|z_{i-(m-1)\tau} - z_{j+(m-1)\tau}| \leq \varepsilon_{\max}$, in which case we add 1 to $S(m, k)$ if $||\mathbf{z}(m)_i - \mathbf{z}(m)_j||$, which has been updated recursively at the same time by maximizing the sequence of norms, lies in $(\varepsilon_{k-1}, \varepsilon_k]$. This sequential calculation stops with the first m for which $|z_{i-(m-1)\tau} - z_{j+(m-1)\tau}|$ exceeds $\varepsilon_{\max}$. Finally, an entry is made in box (p_i, q_i) recording that i belongs to it, following the listing method described by Grassberger (1990).

Experimental results

Figures 1 to 4 are logarithmic plots of the program run times when m takes the values $\{1, ..., 32\}$ or $\{1, 2\}$, and when ε takes the values $\{\frac{\varepsilon_{\max}}{250}, \frac{2\varepsilon_{\max}}{250}, ..., \varepsilon_{\max}\}$ or $\{\frac{\varepsilon_{\max}}{5}, \frac{2\varepsilon_{\max}}{5}, ..., \varepsilon_{\max}\}$. For $\varepsilon_{\max}$ we use 10^{-3} and 10^{-6}. Overall, it is apparent that the sorting-assisted algorithm has more overhead than the box-assisted one, but that it retains its order N property for larger N and larger $\varepsilon_{\max}$, as can be observed from the lines with gradients 1 and 2. The box-assisted algorithm is of order N^2 for $N > 7000$ while the sorting-assisted algorithm remains of order N through to the maximum value $N = 700000$ used here. For such N the sorting assisted algorithm is one order of magnitude faster than the box-assisted algorithm, while for N up to about 7000 the later is faster.

The full generalized fast algorithm is quite comparable with its order N counterparts up to $N = 700$, with the advantage that it calculates $C(m, \varepsilon, N)$ for all values of ε. However, for larger N the duration of the calcualtion increases dramatically. For $N = 70000$ the calculation takes longer than 8300 seconds (with a logarithm close to 4.0).

Comparison of Figures 1 to 4 also shows that including in the calculation many values of m and ε does lengthen it noticeably.

Final Remarks

We provide a sorting-assisted fast algorithm to calculate BDS-type statistics which generalizes the LeBaron's (1997) algorithm following Theiler's (1990) suggestion to first sort the data. This algorithm calculates the m-dimensional histogram $C(m,\varepsilon,N)$ for a finite set of values of m and of $\varepsilon \leq \varepsilon_{\max}$ simultaneously, thus reducing the overall calculation time. When $\varepsilon_{\max}$ is small the number of calculations performed by the algorithm is of order N. We compare this sorting-assisted algorithm with a somewhat extended implementation of Grassberger's (1990) box-assisted algorithm, which obtains $C(m,\varepsilon,N)$ for all dimensions simultaneously, including $m = 1$, and which allows for $\tau > 1$, and also with Mayer's (2000) generalized fast algorithm. We find that the box-assisted algorithm is faster for values of N up to several thousand, while for N in the hundred thousands it is overtaken by the sorting-assisted algorithm. The generalized fast algorithm performs quite comparably for small values of N. One of the main advantages of all of these algorithms is that they are not slowed down by including in the calculation many values of m and ε simultaneously.

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] Grassberger, Peter (1990), "An optimized box-assisted algorithm for fractal dimensions", *Physics Letters A* 148, 63-68.

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

[12] 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.

[13] Mayer-Foulkes, D. (1998), "Homogenized Integral U-Statistics for Test of Non-Linearity", Documento de Trabajo del CIDE, División de Economía, N° 118.

[14] Mayer, D. (2000), "A generalized fast algorithm for BDS-type Statistics", *Studies in Nonlinear Dynamics and Econometrics*, Volume 4, Number 1, April.

[15] Mizrach, B. (1991). "A simple Nonparametric test for independence." Rutgers University Working Paper #95-23.

[16] Mizrach, B. (1992). "Multivariance nearest-neighbour Forecasts of EMS Exchange Rates." *Journal of Applied Econometrics* 7; Supplement, S151-63.

[17] 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.

[18] 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.

[19] Serfling, R. J. (1980), *Approximation Theorems of Mathematical Statistics*, Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons.

[20] Theiler, J. (1995). "On the evidence for low-dimensional chaos in an epileptic electroencephalogram", *Physics Letters A* 196: 335-341.

[21] Theiler, J. (1990). "Estimating Fractal Dimension", *Journal of the Optical Society of America* A 7:10551073.

[22] Theiler, J. (1987) "Efficient algorithm for estimating the correlation dimension from a set of discrete points", *Physical Review A* 36: 4456-4462.

Program Runtimes for Different Values of N (Logarithmic Scales)

Figure 1
Values of Epsylon: 250; Dimensions: 32

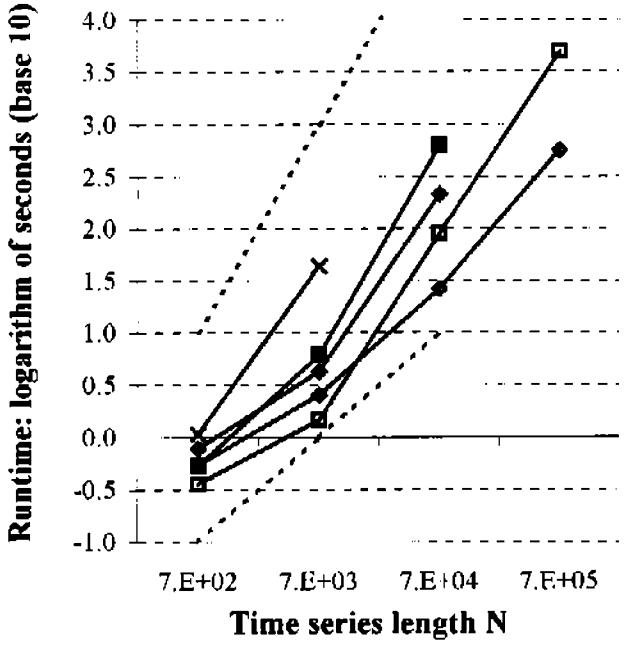

Figure 2
Values of Epsylon: 250; Dimensions: 2

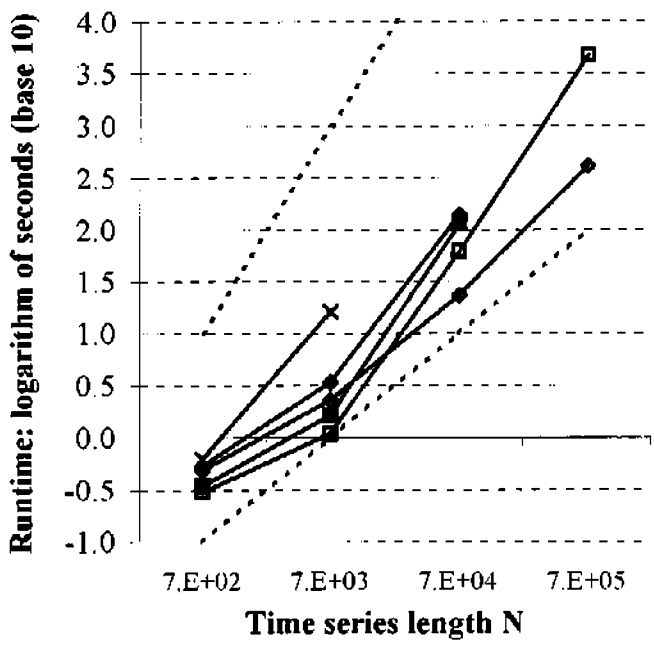

Figure 3
Values of Epsylon: 5; Dimensions: 32

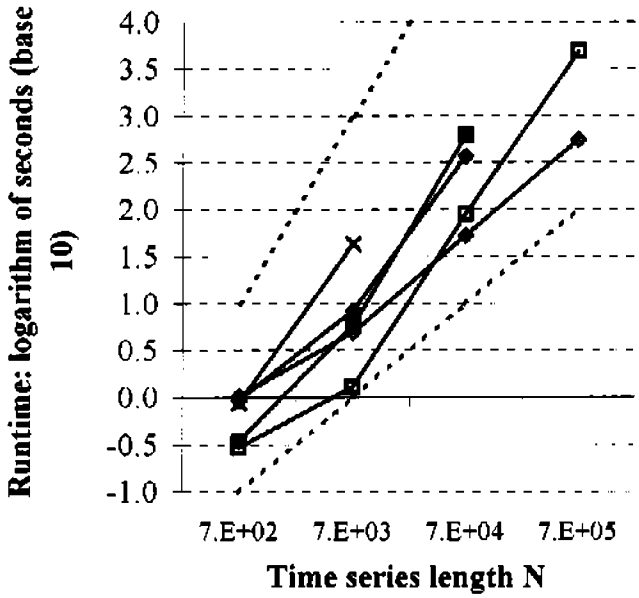

Figure 4
Values of Epsylon: 5; Dimensions: 2