Self-Similar Processes via the Crossings Tree

Owen Jones

University of Southampton

Yuan Shen

University of Warwick

Abstract

Self-similar processes appear in telecommunications, finance, medicine and hydrology.

By considering the crossings of size 2^{-n} made by a continuous process, for $n = 0, 1, 2, ...$, one can build a tree of crossings which encodes the sample path. If the process is self-similar, then the number of subcrossings associated with a crossing of size 2^{-n} will be independent of n, and the expected number of subcrossings will be a simple function of the Hurst index H. These observations lead to a test for self-similarity and an estimator for H.

We can also define a class of self-similar processes, by setting the crossing tree to be a branching process. We call these EBP-processes, for 'Embedded Branching Process'. They are easily fitted to data, can be efficiently simulated on-line, and can be used for forecasting.

Self-Similar Processes

Continuous process X is self-similar if

$$X(t) \stackrel{\mathrm{d}}{=} a^{-H} X(at), \text{ for } H \in (0,1).$$

H is the Hurst index, and measures space/time scaling.

The canonical example is Fractional Brownian Motion (FBM), B_H, which is the continuous Gaussian process with autocovariance function

$$\mathrm{Cov}\,(B_H(s), B_H(t)) = \frac{\sigma^2}{2}(s^{2H} + t^{2H} - |t-s|^{2H}).$$

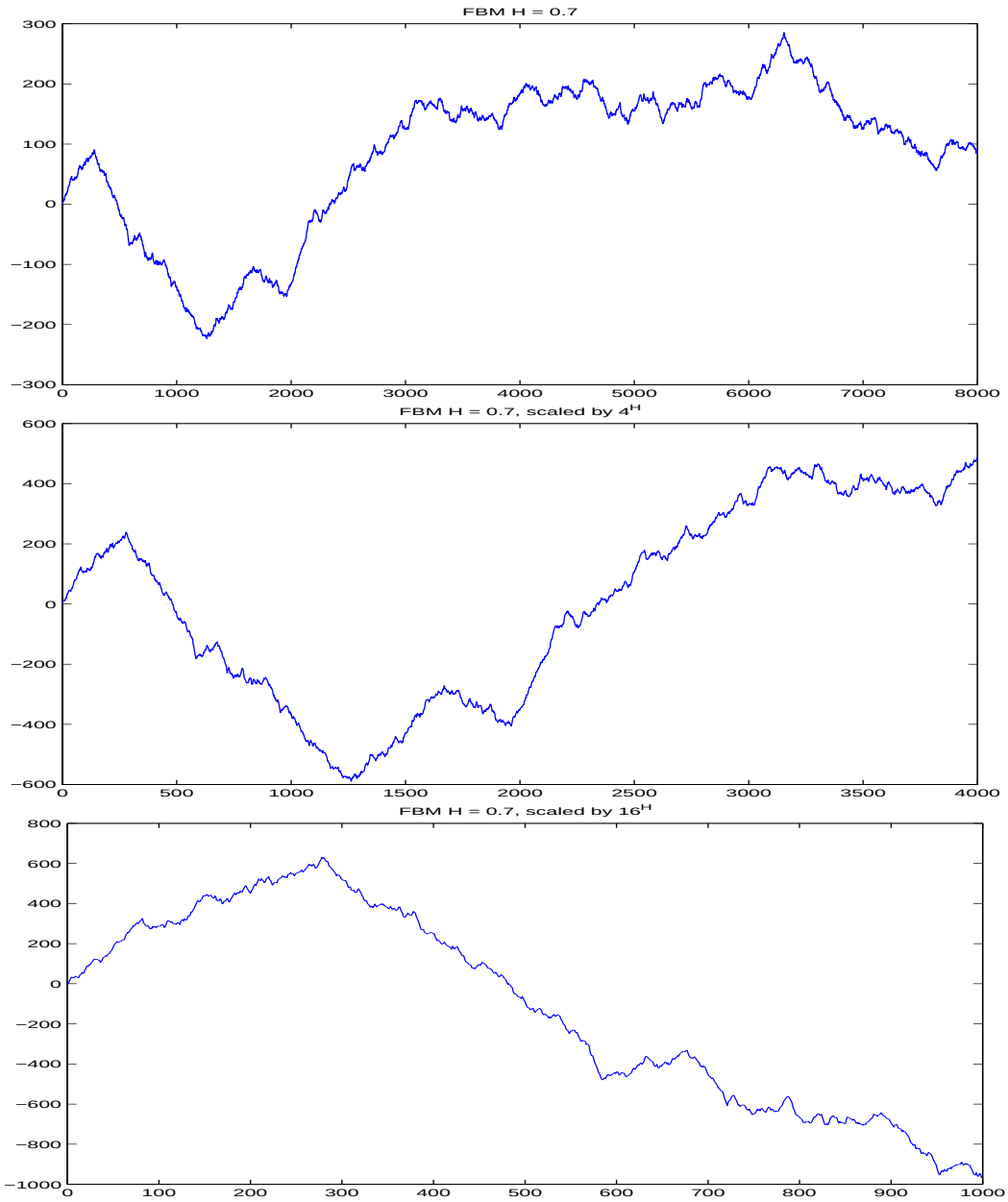
Increments are normal: $B_H(s+t) - B_H(s) \sim N(0, \sigma^2 t^{2H})$.

Check the autocovariance of $B_H(at)$:

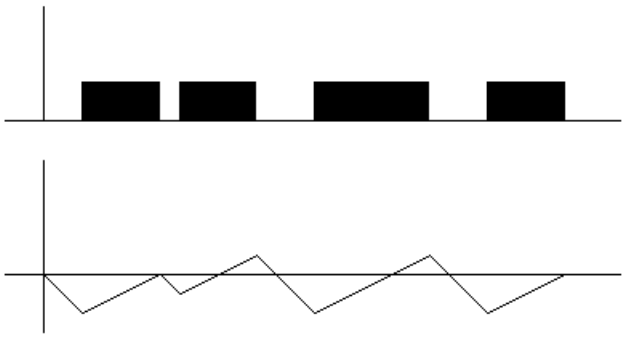
$$\begin{aligned} &\tfrac{\sigma^2}{2}((as)^{2H} + (at)^{2H} - |a(t-s)|^{2H}) \\ &= a^{2H}\tfrac{\sigma^2}{2}(s^{2H} + t^{2H} - |t-s|^{2H}). \end{aligned}$$

which is the autocovariance of $a^H B_H(t)$.

FBM H = 0.7 at 3 different scales. The rescaled processes all behave similarly.



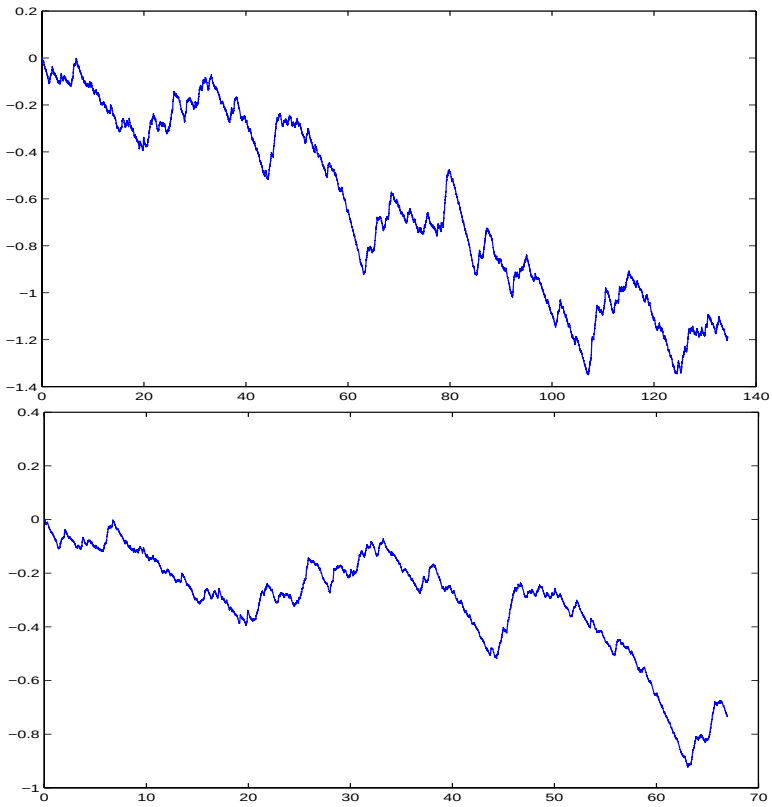
Telecommunications: Packet Arrivals



We take as raw data packet arrival data, that is packet arrival times and packet lengths, which we represent as an on-off process.

We obtain a continuous-time representation by integrating the on-off arrival process, then subtracting mt where m is the mean arrival rate.

Here the Bellcore trace BCAug89 is represented as a continuous process. The trace is plotted at two-scales to illustrate self-similarity



Other examples of self-similar processes include: financial time series; ECG and EEG traces; wind and rainfall; hydrological time series.

Time Series Viewpoint

Suppose $X : \mathbb{R}_+ \to \mathbb{R}$, put $Y(n) = X(n+1) - X(n)$, then self-similarity for X is equivalent to

$$Y \stackrel{\mathrm{d}}{=} m^{1-H}\overline{Y}^m$$

where

$$\begin{aligned} \overline{Y}^m(k) &= \frac{1}{m}\sum_{n=km}^{(k+1)m-1} Y(m) \\ &= \frac{1}{m}(X((k+1)m) - X(km)). \end{aligned}$$

If Y is self-similar and stationary with finite variance, then its autocorrelation has the form

$$\begin{aligned} \rho(k) &= \tfrac{1}{2}(|k+1|^{2H} - 2k^{2H} + |k-1|^{2H}) \\ &\sim ck^{2H-2} \text{ as } k \to \infty. \end{aligned}$$

If Y is also Gaussian then X is FBM. Call Y Fractional Gaussian Noise (FGN) in this case.

Long-Range Dependence

Stationary time series Y is long-range dependent (LRD) if $\rho(k)$ decays slowly enough that

$$\sum_{k=0}^{\infty} \rho(k) = \infty.$$

Self-similarity $\Longrightarrow \rho(k) \sim ck^{2H-2} \Longrightarrow$ LRD for $H \in (\frac{1}{2}, 1)$.

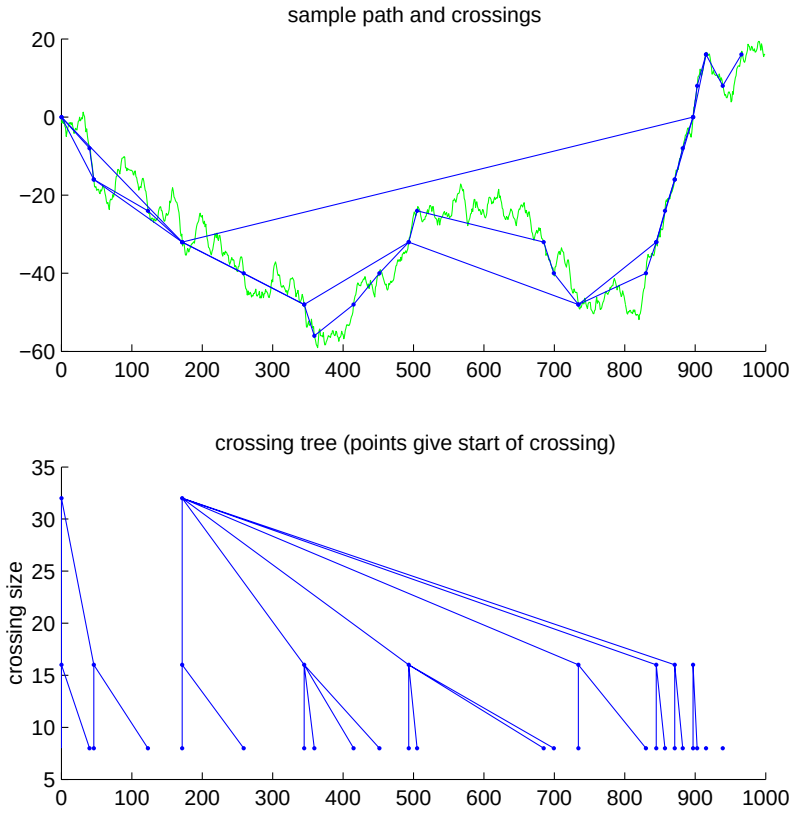
Conversely, if $\rho(k) \sim ck^{2H-2}$ for some $H \in (\frac{1}{2}, 1)$, then $\overline{Y}^m$ converges to a 2nd-order self-similar process as $m \to \infty$. That is, it converges to a process with autocorrelation

$$\rho(k) = \tfrac{1}{2}(|k+1|^{2H} - 2k^{2H} + |k-1|^{2H}).$$

Note that this description of LRD requires finite variances, which is not the case in many practical situations.

Crossings Viewpoint

Given any continuous process X, we get a discrete process X^0 by observing X when it hits points in $\mathbb{Z}$.



We get a sequence of discrete processes X^0, X^1, X^2, ..., by observing X when it hits points in $\mathbb{Z}$, $2\mathbb{Z}$, $4\mathbb{Z}$, ... There is a natural tree structure to the crossings.

For any continuous process we can construct a crossings tree from crossing of size

$$\ldots, 2^{-2}, 2^{-1}, 1, 2, 2^2, \ldots.$$

If X is self-similar, then the number of sub-crossings that make up a crossing of size 2^k does not depend on k.

The expected number of sub-crossings μ tells us the space/time scaling.

If we scale space by 2^k then must scale time by μ^k to get crossings of same length.

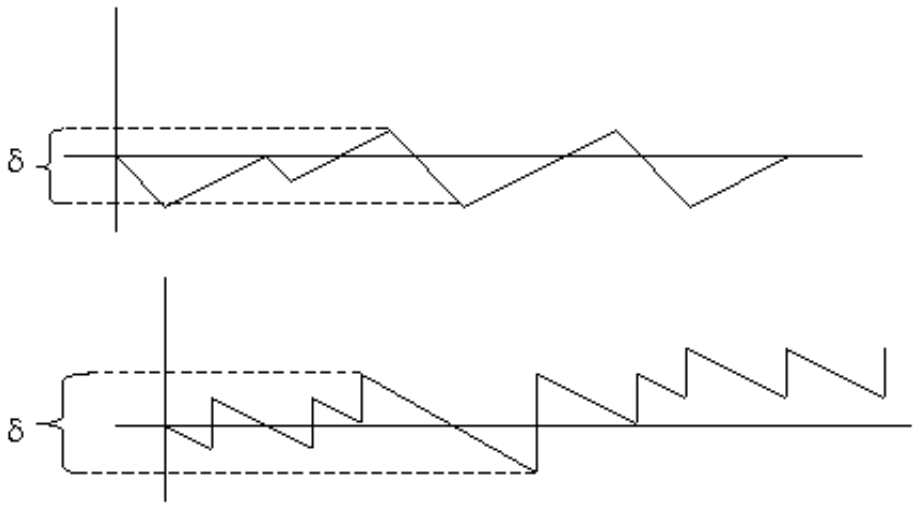
$$X(t) \stackrel{\mathrm{d}}{=} 2^{-k} X(\mu^k t) = (\mu^k)^{-\log 2/\log \mu} X(\mu^k t).$$

So the Hurst index is

$$H = \log 2/\log \mu.$$

Testing for Self-Similarity

Start by determining scale for smallest crossings. Call this the resolution δ.
δ has to be large enough to cover linear segments or jumps.



Build the crossing tree up from the bottom, then count family sizes.

$$\begin{array}{c} Z_1^1, Z_2^1, \ldots, Z_{n_1}^1 \\ Z_1^2, Z_2^2, \ldots, Z_{n_2}^2 \\ \vdots \\ Z_1^m, Z_2^m, \ldots, Z_{n_m}^m \end{array}$$

Here Z_i^k is the number of subcrossings of size $\delta 2^{k-1}$ that make up the i-th crossing of size $\delta 2^k$.

If the sequence $Z_1^k, Z_2^k, \ldots$ is ergodic, then we can estimate the distribution of Z_i^k empirically. Let

$$\begin{aligned} p^k(x) &= \mathsf{P}(Z_i^k = x), \\ \mathbf{p}^k &= (p^k(0), p^k(1), \ldots). \end{aligned}$$

If the process X is self-similar, then

$$\mathbf{p}^k = \mathbf{p}^l \text{ for all } k \text{ and } l$$

In practice we find that self-similarity holds for only a finite range of scales. For any $k < l$, we use a contingency table to test the hypothesis

$$Z_i^k \stackrel{\mathsf{d}}{=} Z_i^{k+1} \stackrel{\mathsf{d}}{=} \ldots \stackrel{\mathsf{d}}{=} Z_i^l.$$

Estimate for H

$H = \log 2 / \log \mu$ where μ is the expected number of subcrossings. If $Z_1^k, Z_2^k, \ldots$ is ergodic, then $\widehat{\mu}_k = \overline{Z}^k$ is a consistent estimator for μ.

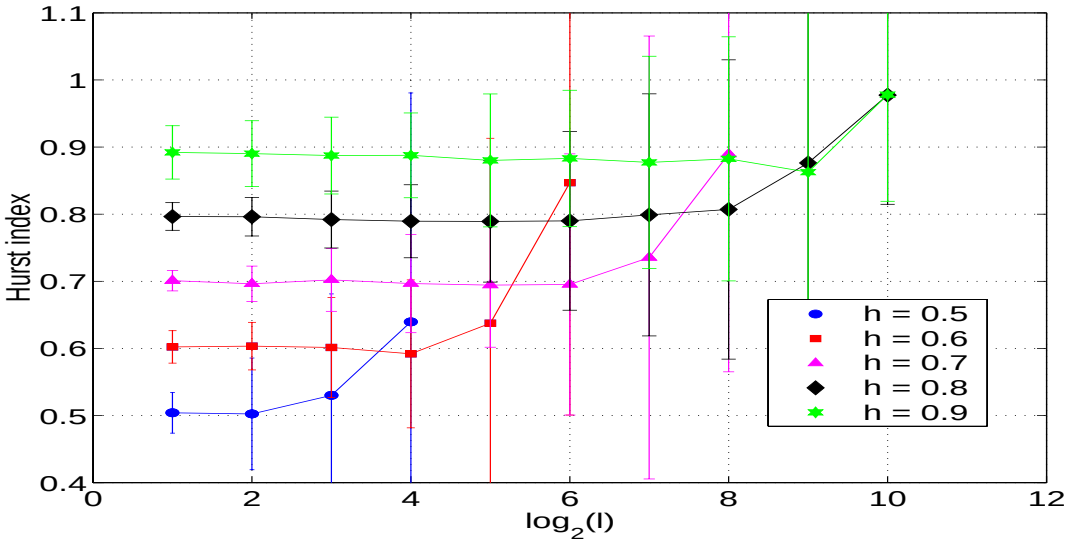
Estimator for scaling index at scale $\delta 2^k$ is

$$\widehat{H}_k = \log 2 / \log \widehat{\mu}_k.$$

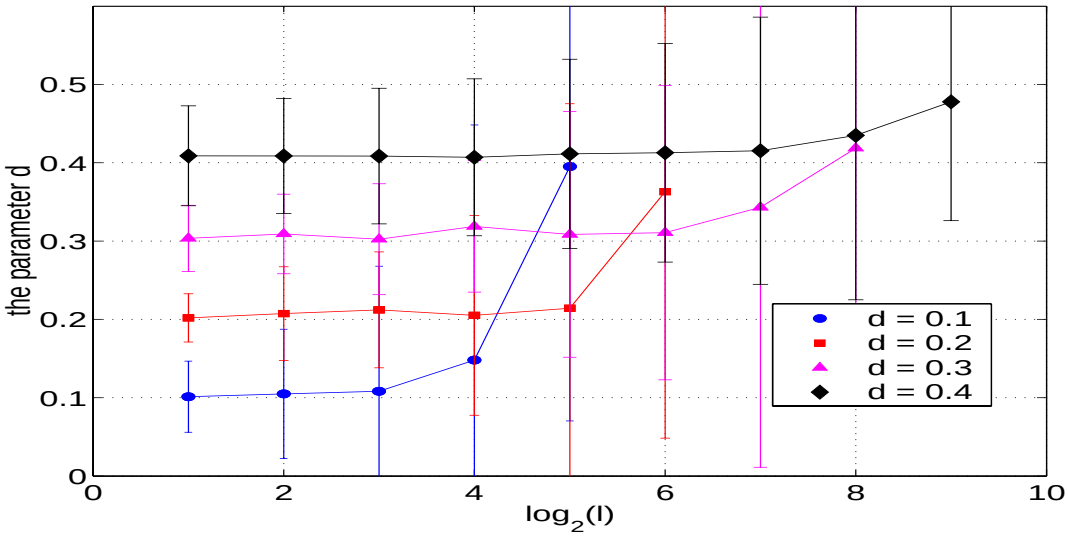
If we believe that self-similarity holds over scales $\delta 2^k$ to $\delta 2^l$, then we can combine these levels to get a more accurate estimate

$$\begin{aligned} \widehat{\mu}_{k,l} &= \frac{n_k \overline{Z}^k + n_{k+1} \overline{Z}^{k+1} + \cdots + n_l \overline{Z}^l}{n_k + n_{k+1} + \cdots + n_l} \\ \widehat{H}_{k,l} &= \log 2 / \log \widehat{\mu}_{k,l}. \end{aligned}$$

FBM



ARFIMA(0,d,0) ($H = d + 0.5$)



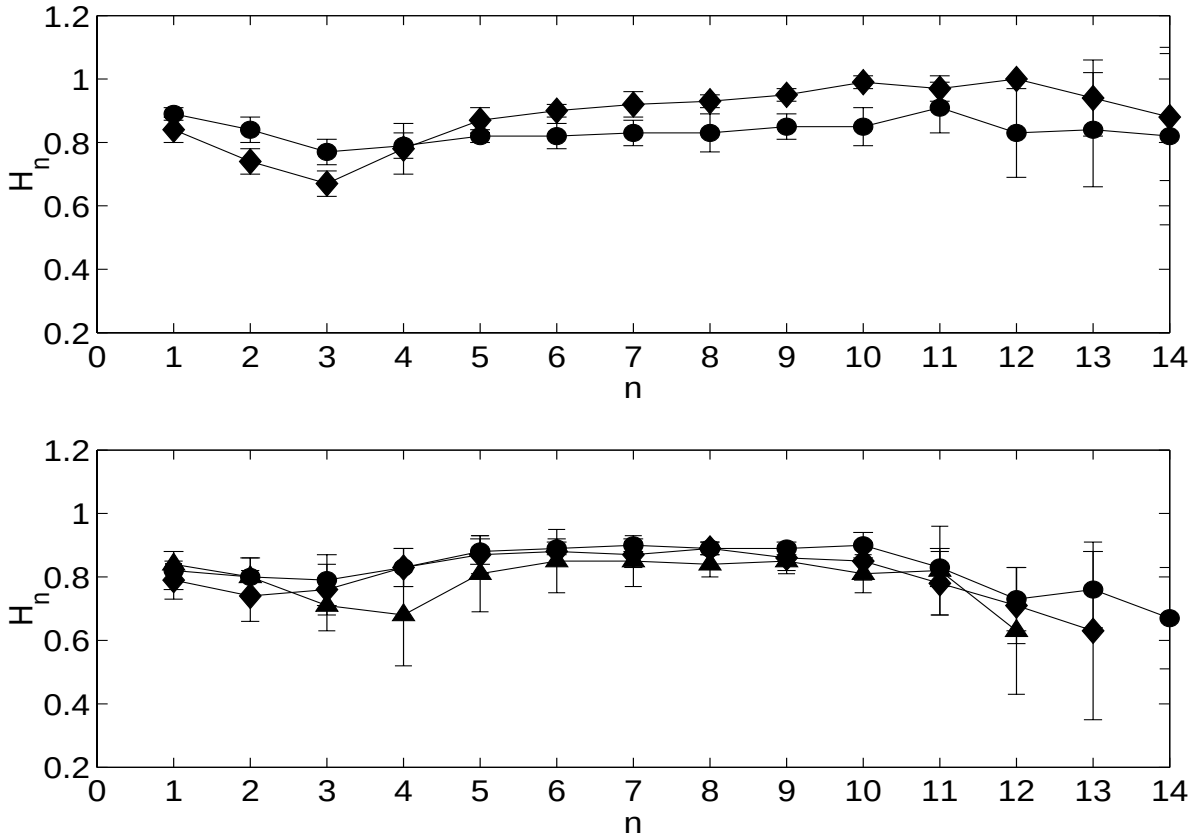
Estimates of $\hat{H}_l$ for simulated traces at different scales l. Error bars obtained from repeated simulations.

Comparative Results

process	H	DFA	WAV	R/S	EBP
FBM	0.5	−0.014(75)	−0.043(92)	+0.067(37)	+0.007(09)
	0.6	−0.010(53)	−0.016(67)	+0.023(63)	+0.006(09)
	0.7	−0.023(48)	−0.024(56)	+0.005(31)	−0.001(08)
	0.8	−0.048(78)	−0.019(39)	−0.013(14)	−0.002(14)
	0.9	−0.043(50)	−0.014(60)	−0.049(25)	−0.008(20)
ARFIMA	0.6	−0.014(64)	−0.039(97)	+0.025(70)	+0.004(11)
	0.7	−0.011(57)	−0.026(62)	−0.001(51)	+0.003(12)
	0.8	−0.042(83)	−0.000(60)	−0.005(33)	+0.005(18)
	0.9	−0.026(78)	−0.019(53)	−0.048(21)	+0.010(31)

Comparison of the bias and the standard deviation (in brackets $*1000$) of H estimates between four different estimation methods: detrended fluctuation, wavelets, rescaled analysis and embedded branching. Standard deviations obtained by Monte Carlo trials.

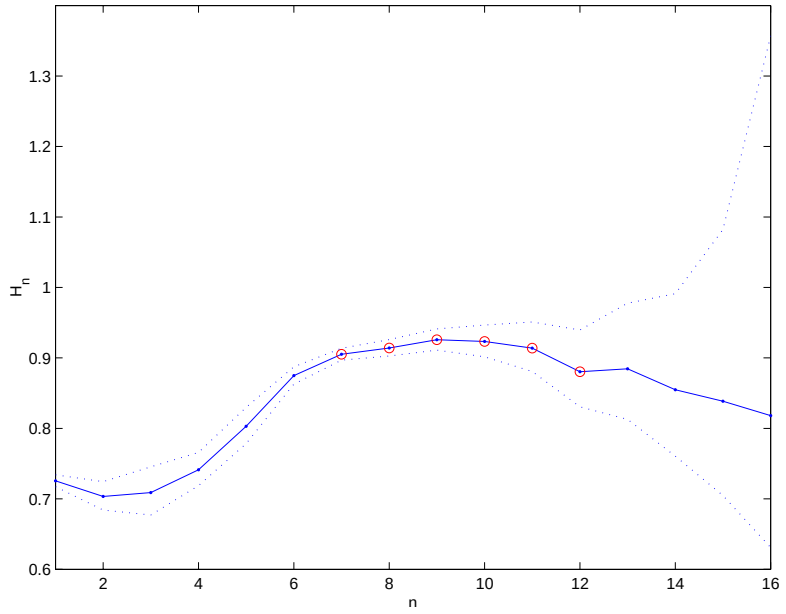
Packet Arrivals: LAN and WAN Data

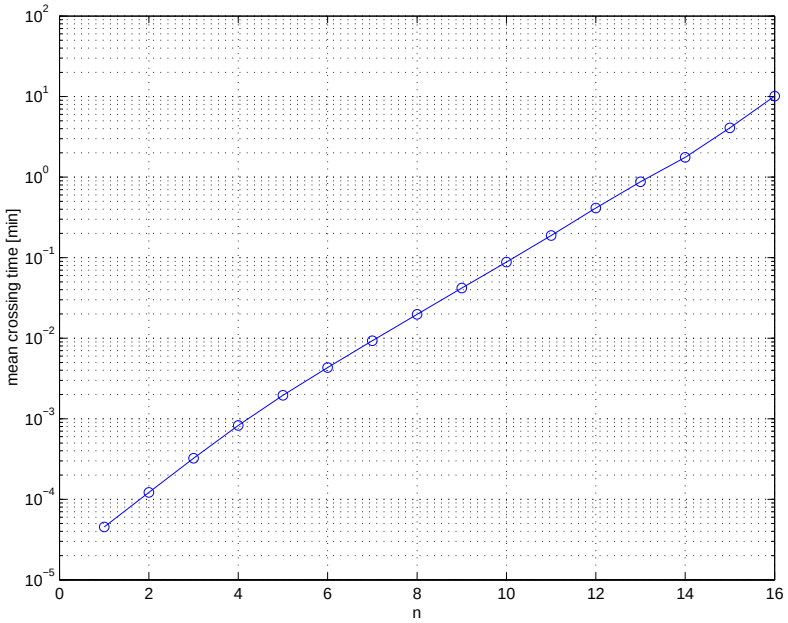


Estimates of $\widehat{H}_n$ for Bellcore packet traces at different scales n. Confidence intervals are derived from estimates of Var $\widehat{\mu}_n$.

Traces are BCAug89, BCOct89 (top), LBL3, LBL4 and LBL5 (bottom).

Packet Arrivals: York Data





Estimates of $\widehat{H}_n$ for York University packet traces at different scales n. Confidence intervals are derived from estimates of Var $\widehat{\mu}_n$.

Maximum Likelihood Estimate for H

If we assume that the crossings tree is a *branching process*, then we can obtain its likelihood function. In this case the Z_i^k are i.i.d. and, supposing we observe levels k to l,

$$\widehat{H}_{k,l} \text{ is the MLE of } H.$$

Moreover, we have that $\widehat{\mu}_k$ is unbiassed (though not $\widehat{\mu}_{k,l}$) and given n_l, for $h = l - k$

$$\widehat{\mu}_{k,l} \approx N\left(\mu, \frac{\sigma^2}{n_l}\frac{\mu^{2h+3} - 1 - (2h+1)\mu^{h+1}(\mu-1)}{(\mu^{h+1}-1)^2(\mu-1)}\right),$$

where $\mu = \mathsf{E}Z_i^j$ and $\sigma^2 = \mathsf{Var}\, Z_i^j$.

The MLE of σ^2, given we observe levels k to l, is

$$\widehat{\sigma}^2_{k,l} = \frac{1}{\sum_{j=k}^{l} n_j}\sum_{j=k}^{l}\sum_{i=1}^{n_j}(Z_i^j - \widehat{\mu}_{k,l})^2.$$

Transform Bias

A bias is introduced by the log transform $\widehat{H} = \log 2 / \log \widehat{\mu}$. This bias is proportional to $\operatorname{Var} \widehat{\mu}$, so decays quickly as the sample size increases, and can be estimated given an estimate of $\operatorname{Var} \widehat{\mu}$.

Finite Sample Bias

We observe $X(t)$ over a finite time interval $[0, T]$. Large crossings are thus biassed to be shorter than they should, which results in an underestimate of μ (overestimate of H) at large scales.

The bias can be quantified in special cases, and decays quickly.

Confidence interval for H

Consider the estimator $\hat{H}_k$ based on a single level k. Under the hypothesis that the crossings tree is a branching process, the subcrossing sizes Z_i^k are all independent.

In practice, we can find small but significant correlation in the sequence $Z_1^k, Z_2^k, \ldots$.
For FBM we observe empirically that

$$\rho(r) = \text{Corr}\,(Z_i^k, Z_{i+r}^k) \approx cr^{-\alpha} \text{ for } r \geq 1$$

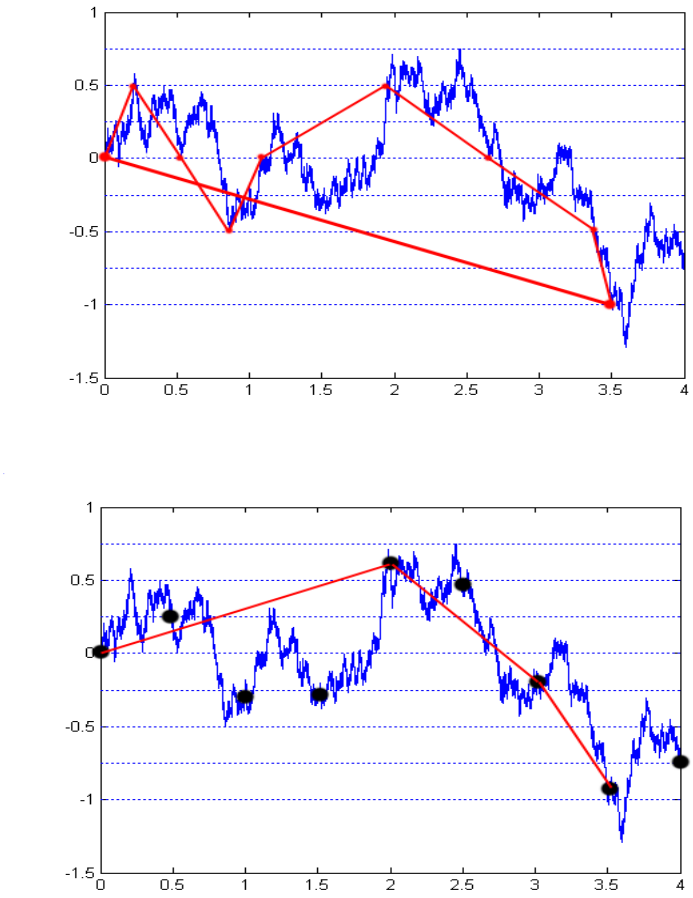
(c.f. analogous result for wavelets).

We must account for this when estimating $\text{Var}\,\hat{\mu}_k$. If $\alpha \leq 1$ then the Z_i^k sequence is long-range dependent.

Application to Time Series

If we only observe the process at regular time points then we may not see all the small crossings.

Top diagram gives a sample path and all its crossings. Bottom diagram gives crossings observed if the process is sampled at regular points (the solid dots).

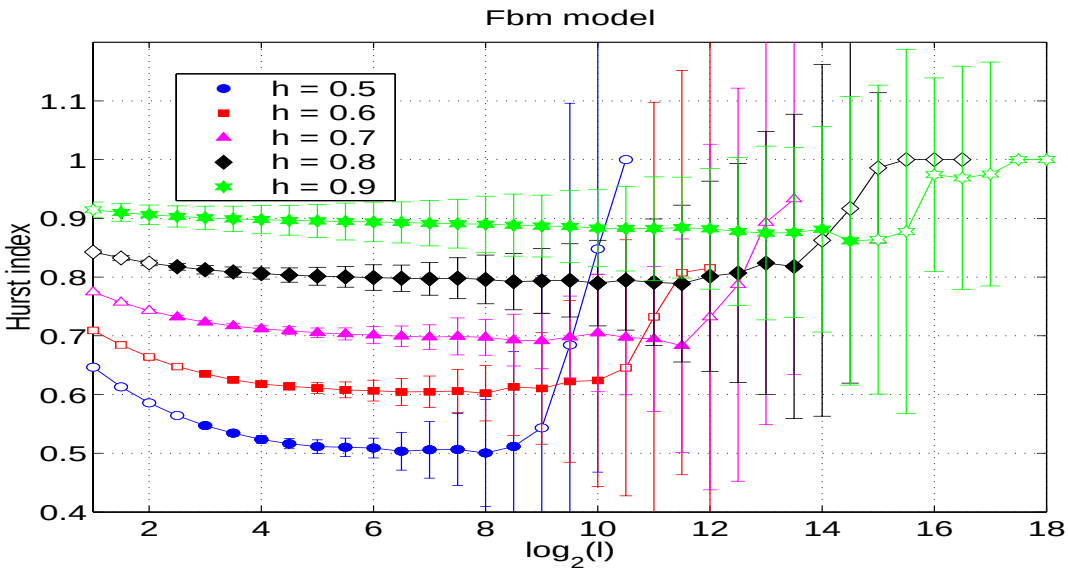


Missing crossings introduce a bias in $\widehat{\mu}$, which we can fix at the expense of increasing the variance, simply by increasing the base resolution δ.

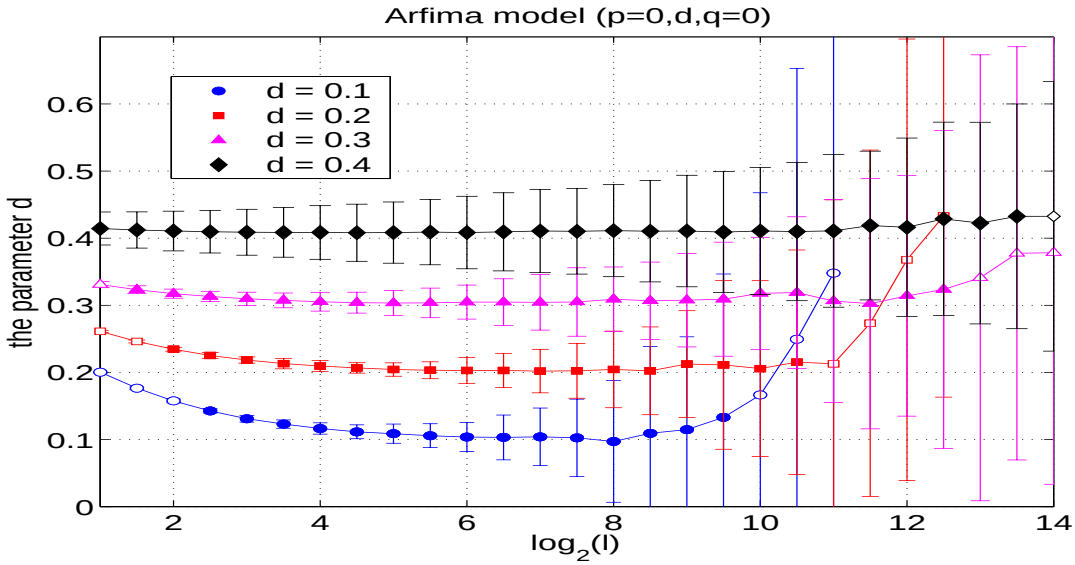
We can get some accuracy back by using the 'Random Midpoint Displacement' algorithm to simulate extra points in between the observed values. This is a quick fix with no theoretical backup however.

We require an estimate of H to apply the Random Midpoint Displacement algorithm, necessarily this must come from the data, so our estimate for H based on the augmented data will not be as accurate as we think it is. Also, the Random Midpoint Displacement algorithm is known to have problems of its own.

FBM Time Series



ARFIMA Time Series ($H = d + 0.5$)



Estimates of $\hat{H}_l$ for simulated traces at different scales l. Error bars obtained from repeated simulations.

Comparative Results

process	H	DFA	WAV	R/S	EBP
FBM	0.5	$-0.002(04)$	$-0.002(02)$	$+0.005(09)$	$+0.007(09)$
	0.6	$+0.000(07)$	$+0.000(04)$	$+0.004(07)$	$+0.007(07)$
	0.7	$-0.004(07)$	$-0.003(05)$	$+0.000(06)$	$+0.001(06)$
	0.8	$-0.008(19)$	$-0.004(05)$	$-0.007(3)$	$+0.001(09)$
	0.9	$-0.022(29)$	$-0.003(07)$	$-0.029(2)$	$+0.000(11)$
ARFIMA	0.6	$-0.002(03)$	$-0.003(04)$	$+0.003(10)$	$+0.004(08)$
	0.7	$-0.003(05)$	$-0.003(05)$	$+0.001(07)$	$+0.004(07)$
	0.8	$-0.001(06)$	$-0.001(05)$	$-0.005(04)$	$+0.005(09)$
	0.9	$-0.004(07)$	$-0.003(05)$	$-0.027(03)$	$+0.011(16)$

Comparison of the bias and the standard deviation (in brackets $*1000$) of H estimates between four different estimation methods: detrended fluctuation, wavelets, rescaled analysis and embedded branching. Standard deviations obtained by Monte Carlo trials.

Time Heterogeneous H

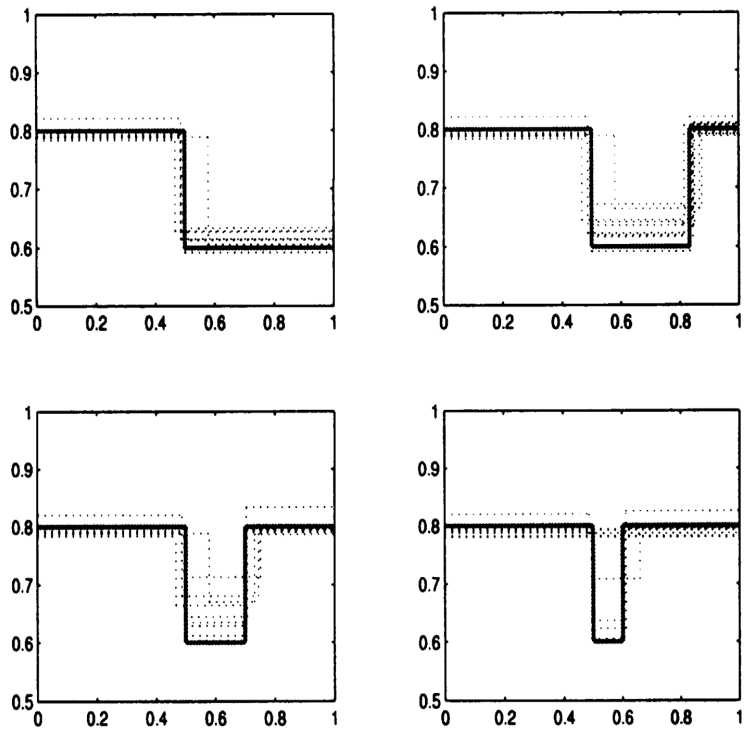
Fix scale at $\delta_k = \delta 2^k$. This splits the time into crossings, with expected length $t_0 \mu^k$, where $t_0 \approx \delta^{1/H}$ is the expected time to move distance δ.

Each crossing of size δ_k gives rise to a subset of the crossings tree, from which we can estimate H and the subcrossing distribution.

The contingency table test for equality of subcrossing distributions provides a tool for detecting changes in H.

Using simulated traces we tested the ability of the contingency table to detect a change in H. FBM with different values of H were concatenated then change points were estimated and H calculated for each region.

In the following figures the solid line is the target value of H. The dotted lines give estimates for 10 different simulation runs



To test for false positives FBM with constant H was checked for change points. 2 out of 10 traces seemed to have a change in H

Changing m and H in packet traces

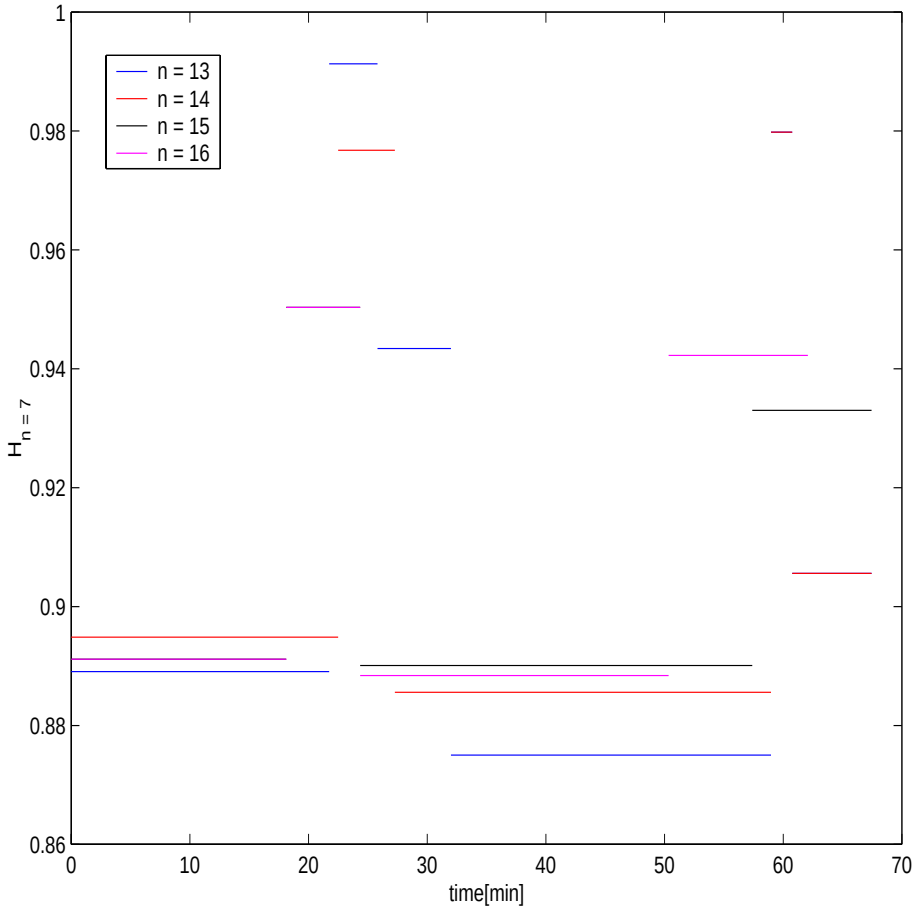
A self-similar process is obtained from an on-off packet arrival trace by integrating then subtracting mt, where m is the mean arrival rate. The crossing tree will reflect changes in either H or m (or both).

Experiments with simulated traces indicate that the contingency table test is more sensitive to changes in m than to changes in H.

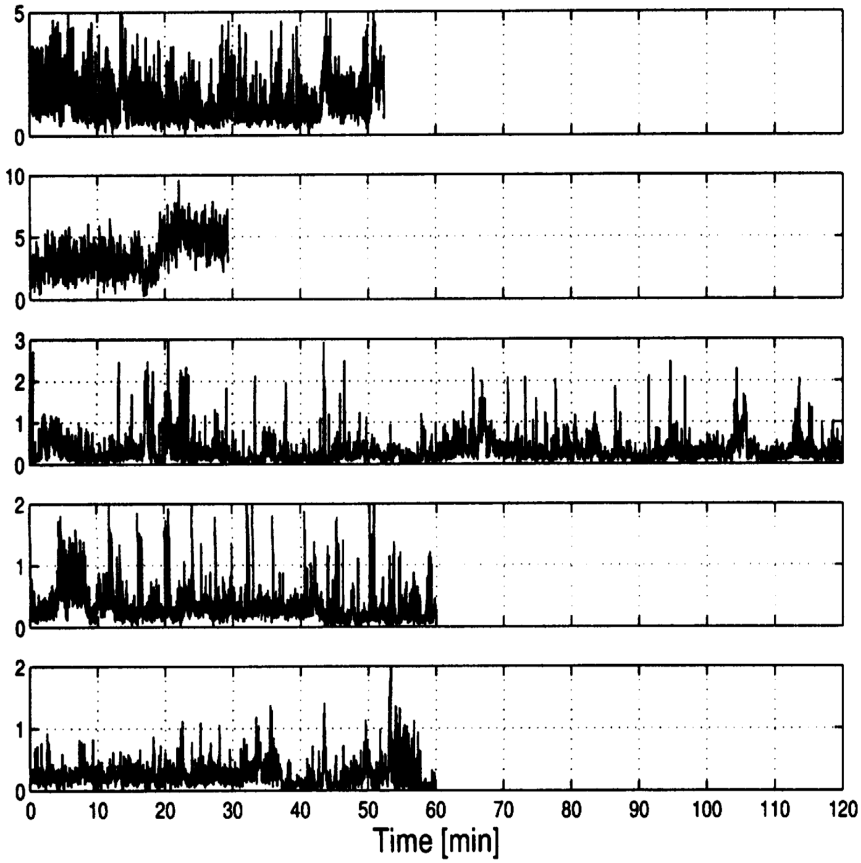
Non-stationarity of York traffic data

Time variation of Hurst index for York data. Using four different scales δ_k to split the trace into blocks, changes in H and m were estimated. Each gave very similar results.

The following figure gives plots of H against time for each choice of δ_k



Non-stationarity of Bellcore traffic data



Bellcore traces BCAug89, BCOct89, LBL3, LBL4 and LBL5.

Using blocks defined by crossings of size 2^{10} (crossing times in the order of 5 minutes) the following changes in H were detected.

Trace	Time (min.)	H
BCAug89	(0.0 51.0]	0.82 $\pm$ 0.02
BCOct89	(0.0 20.0]	0.81 $\pm$ 0.03
	(20.0 28.7]	0.74 $\pm$ 0.03
LBL3	(0.0 120.0]	0.90 $\pm$ 0.02
LBL4	(0.0 54.3]	0.90 $\pm$ 0.02
	(54.3 60.0]	0.81 $\pm$ 0.10
LBL5	(0.0 33.9]	0.80 $\pm$ 0.04
	(33.9 57.4]	0.87 $\pm$ 0.03

EBP Processes

Every continuous process has an embedded crossing tree, and given the crossing tree we can generate the process.

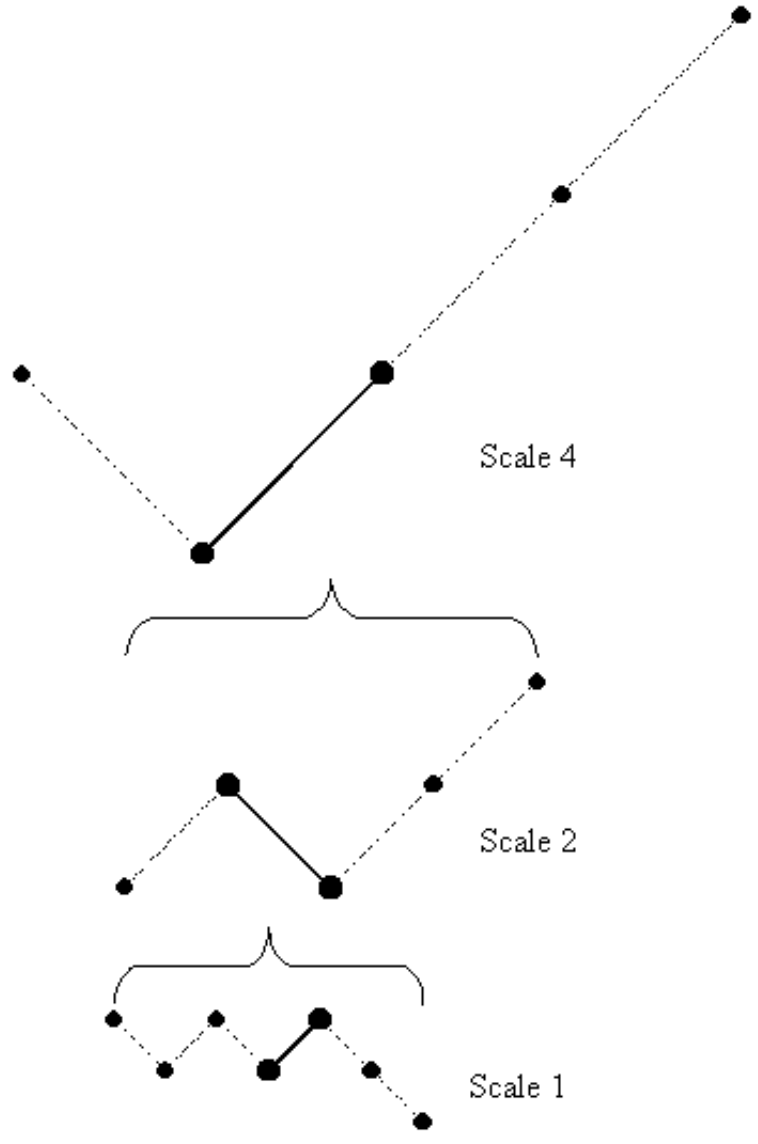
We obtain a large class of self-similar processes by generating a crossing tree from a **branching process**. Call this class of processes Embedded-Branching-Process (EBP) processes.

Use EBP processes to model self-similar processes. They are easy to fit, efficient to simulate, and can be used for forecasting.

Simulating EBP Processes

Can simulate EBP process "on-line". That is, having simulated the first n crossings, we can generate the next crossing on demand.

The simulation uses a representation of the process as an ∞-dim Markov chain.



A state of the Markov chain records, for each level $k \in \mathbb{Z}$, the number and orientation of subcrossings, and which of these subcrossings is current (that which contains $X(t)$).

The crossings are nested, in that the subcrossings at level k are a refinement of the current crossing at level $k+1$.

We can increment crossings of size 2^k one at a time, until we reach the end of the current 2^{k+1} crossing. At this point we increment a crossing of size 2^{k+1}, and use it to generate a consistent set of 2^k subcrossings, then set the current 2^k to be the first of these.

In practice we can truncate the sequence of crossings below at level $k = 0$. That is, we take crossings of size 1 as our smallest crossings.

To simulate $X(t)$ we start with crossings of size 1. The first time we reach the end of a crossing of size 2^{k+1}, we need to know where it sits in the current crossing of size 2^{k+2}.

This is equivalent to sampling at random from the current generation of a branching process, and asking the size of the family the selected individual lies in. If

$$p(j) = \mathsf{P}(j \text{ subcrossings}),$$

then the sampling distribution for the number of subcrossings in the current crossing is

$$jp(j)/\mu.$$

To finish the simulation, we need the crossing times for the crossings of size 1. These are i.i.d., with distribution equal to the normed limit W of the branching process. We have an efficient method for simulating W.

The final algorithm takes time $O(n)$ and storage $O(\log n)$ to generate n steps. It is on-line in that, having generated n steps, you can generate step $n+1$ on demand.

Forecasting for EBP Processes

The ∞-dim Markov representation of an EBP process can also be used for forecasting.

The current state of the Markov chain can be inferred from the crossings tree. Given this, we use Monte Carlo simulations of the process to estimate its future behaviour.