The Relevance of Long-Range Dependence in Disk Traffic and Implications for Trace Synthesis

Bo Hong
Storage Systems Research Center
University of California, Santa Cruz
hongbo@cse.ucsc.edu

Tara M. Madhyastha
Dept. of Computer Engineering
University of California, Santa Cruz
tara@cse.ucsc.edu

Abstract

Accurate disk workloads are crucial for storage systems design, but I/O traces are difficult to obtain, unwieldy to work with, and unparameterizable. I/O traces are often bursty and difficult to characterize. Although good models of I/O workloads would be extremely useful, such bursty traces cannot accurately be modeled using exponential or Poisson arrival times. Much experimental evidence suggests that I/O traces are self-similar, which researchers have hoped might help to model bursty traces. In this paper, we show that self-similarity at large time scales does not significantly affect disk behavior with respect to response times. This allows us to generate synthetic arrival patterns at relatively small time scales, improving the accuracy of trace generation. The relative error of our method, with input parameters suitable for the workload, ranges from approximately 8% to 12%.

1. Introduction

Performance analysis and architecture of storage systems depend heavily upon traces and simulations. Unfortunately, I/O traces are difficult to obtain, extremely large and unwieldy, and cannot be parameterized. Thus, system researchers often resort to benchmarks, whose accuracy usually depends heavily on the underlying workload models.

Ideally, one would like to monitor any disk workload and model it accurately (with respect to some important performance metrics) with some small number of parameters. This vision is far from reality; however, this paper identifies parameters that can capture request interarrival burstiness.

We consider an I/O trace that consists of a set of timestamped values each containing a disk offset, a read/write flag, and a length. This low-level description accommodates a primitive application-level or SCSI I/O interface. We wish to model these streams accurately enough so that accesses synthesized from the model cause a storage hierarchy to behave "similarly" to the real trace. For a single disk drive, similar behavior is measured by checking whether the distributions of response times as well as queue lengths resemble those created by the original workload.

We show that long-range dependence in I/O traffic does not significantly affect disk behavior. This allows us to generate synthetic interarrival patterns at relatively small time scales, improving the accuracy of trace generation.

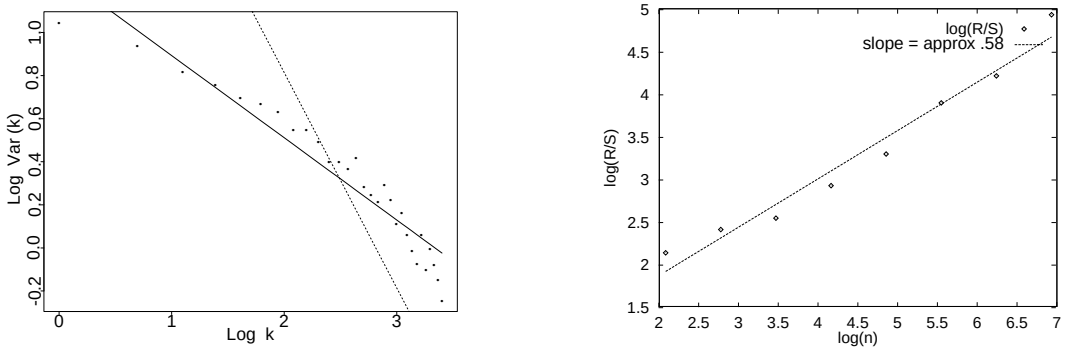
The paper is organized as follows. We describe related work in Section 2. We introduce self-similarity and the Hurst coefficient as a way to approximate long-range dependence and show that long-range dependence has little effect upon disk response times in Section 3. Binomial multifractals can generate bursty traffic; Section 4 describes this model. In Section 5 we describe a novel I/O request synthesis technique using multifractal models. We evaluate our method in Section 6 and conclude with directions for future work in Section 7.

2. Related Work

Generating realistic disk traces is a difficult and unsolved problem [5]. Much experimental evidence shows that disk I/O, file, network, and Web traffic shares some common properties, such as burstiness and long-range dependence, with self-similar and multifractal processes [4, 6, 7, 10]. These properties cannot accurately be modeled using exponential or Poisson arrival times.

Several researchers have used self-similarity to model bursty traces, particularly network traces. Chen *et al.* [2] examined ATM variable bit rate traffic and found that the higher the Hurst coefficient, a measure of self-similarity, the burstier the traffic. Multifractal models, or generalizations of self-similar traffic models, have been shown to model some kinds of traffic more effectively than self-similar models [3]. Wang *et al.* [19] proposed to use binomial multifractals to model bursty disk traffic. The model

Proceedings of the 22nd IEEE/13th NASA Goddard Conference on Mass Storage Systems and Technologies (MSST 2005)
0-7695-2318-8/05 $20.00 © 2005 IEEE



(a) Variance plot

(b) R/S plot

Figure 1. Estimating the Hurst parameter.

is parsimonious, depending only upon a single parameter, bias p, which can be estimated from real traces. Wang *et al.* [18] further proposed a two-dimensional multifractal model that characterizes both the temporal and spatial behaviors of data accesses and captures the spatio-temporal correlations using the joint entropy of the two-dimensional disk request arrival events (time and space).

Grossglauser and Bolot [8] demonstrated that it was not useful to model long-range dependence in network traffic at time scales disproportionate to the performance metrics under observation. Neidhardt and Wang [12] showed that queuing behavior depends not only on the Hurst coefficient, but a combination of system parameters. Our approach is to investigate whether this is true for I/O traffic, and whether this fact is useful for improving multifractal synthesis techniques.

3. Relevance of Long-Range Dependence in Disk Traffic

Some researchers have claimed that bursty I/O workloads have a structure that might help to model them called *self-similarity*. Informally, in this context, to say a time series is self-similar implies that it looks qualitatively the same at different time scales. Self-similar traffic has the property of *long-range dependence*: the data set exhibits a slow decay in its autocorrelation function. This correlation structure is significant because self-similar traffic may be more bursty than that generated by other sources. However, we show here that long-range dependence, as measured by the Hurst coefficient, has little effect upon disk response times.

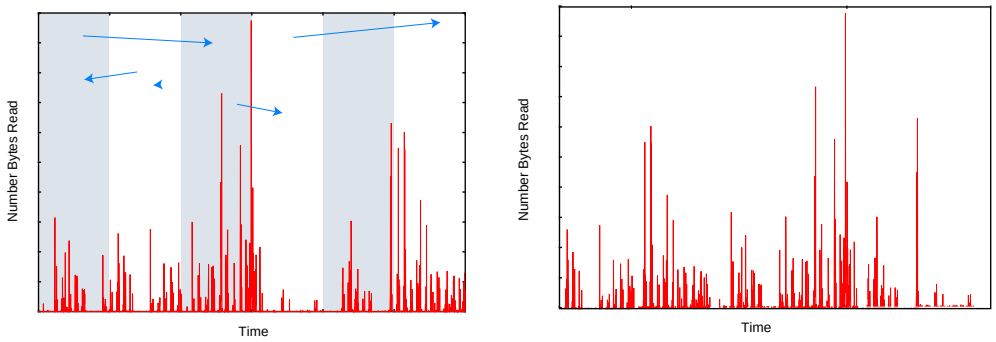
3.1. Self-Similarity

A more rigorous definition of self-similarity from [1] is as follows: Let Y_t be a stochastic process with continuous time parameter t. Y_t is called self-similar with self-similarity parameter H, if for any positive stretching factor c, the rescaled process with time scale ct, $c^{-H}Y_{ct}$, is equal in distribution to the original process Y_t. The parameter H is also known as the Hurst coefficient, and a value of H between $\frac{1}{2}$ and 1 indicates the degree of self-similarity. Generally speaking, the Hurst coefficient is a predominant way to quantify long-range dependence in a stochastic process. A comprehensive treatment of the Hurst parameter is presented by Meakin [11].

There are several exploratory analytic tools that are used to estimate H; two such methods are applied to a small UNIX workstation disk trace [14] in Figures 1(a) and 1(b). The first method, shown in Figure 1(a), is the variance plot. We plot the logarithm of the variance of an aggregated (averaged) series against the logarithm of the aggregation level. The slope of this plot should be equal to $H-1$. The second method, shown in Figure 1(b), is the R/S plot. This method plots (in logscale) the R/S statistic, or the *rescaled adjusted range* against $\log n$. The rescaled adjusted range is the data range normalized by the standard deviation. The precise definition of how to calculate this statistic can be found in [1, 17]. If there is long-range dependence in the process, the slope of the curve generated by this plot provides an estimate of H; if not, $\log R/S$ should be randomly scattered around a straight line with a slope of 0.5 [1].

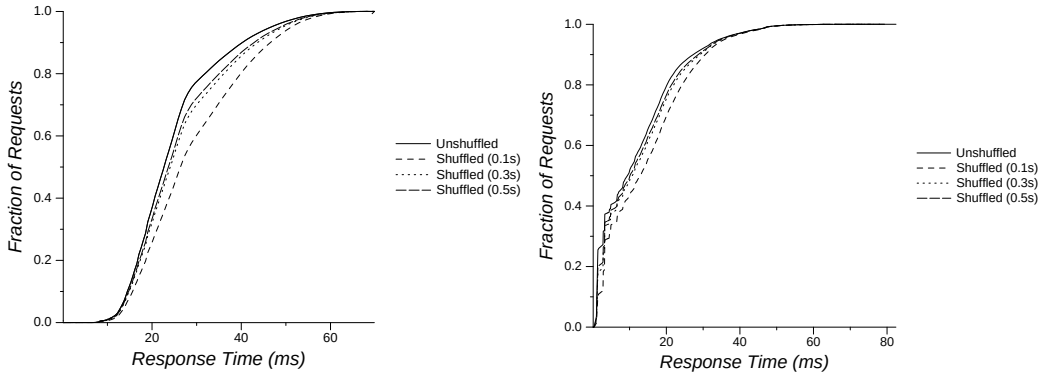
3.2. Long-Range Dependence in Disk Traffic

Our hypothesis is that previous events cannot affect disk behavior beyond a certain threshold, determined by system parameters, so modeling long-range dependence at large time scales is unnecessary. To test this hypothesis, we study how disk response time changes as we gradually destroy long-range dependence in the traces by shuffling increasingly smaller intervals. This approach is identical to the experimental approach taken by [8], and is illustrated in Figure 2. Figure 2(a) shows a trace that has been divided into



(a) Original/unshuffled trace (b) Shuffled trace

Figure 2. Shuffling traces removes long-range dependence.



(a) Cello (b) Snake

Figure 3. Response time distributions for traces shuffled using small intervals.

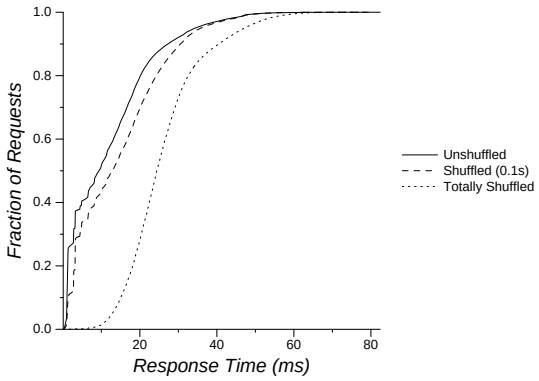


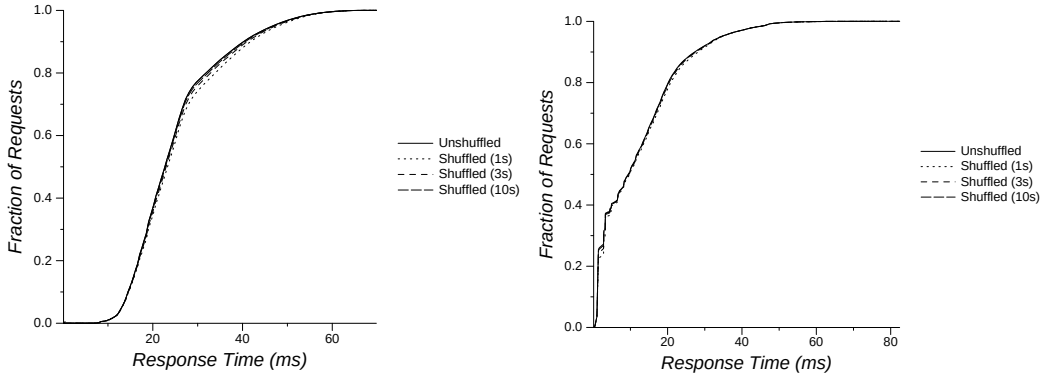
Figure 4. In the limit, all temporal locality is lost when we shuffle a snake trace.

six intervals. These intervals are then randomly rearranged to create a new trace (Figure 2(b)). Within each interval, the temporal relationships are preserved, but the new trace has no long-range dependence beyond the interval width.

Our selected workloads, described in more detail in [14], are the cello news disk trace (HP2204A) and the snake usr2 disk trace (HP97560) gathered between 05/30/92 and 06/06/92. The average I/O loads on these disks are small: approximately three requests for the cello news disk and one request for the snake usr2 disk per second. However, the maximum queue length can be very large: over 1000 requests cello and over 60 requests on snake. In general snake is more bursty than cello, and the logical sequentiality (percentage of requests that are at adjacent disk addresses or addresses spaced by the file system interleave factor) of cello and snake is 2% and 29%, respectively.

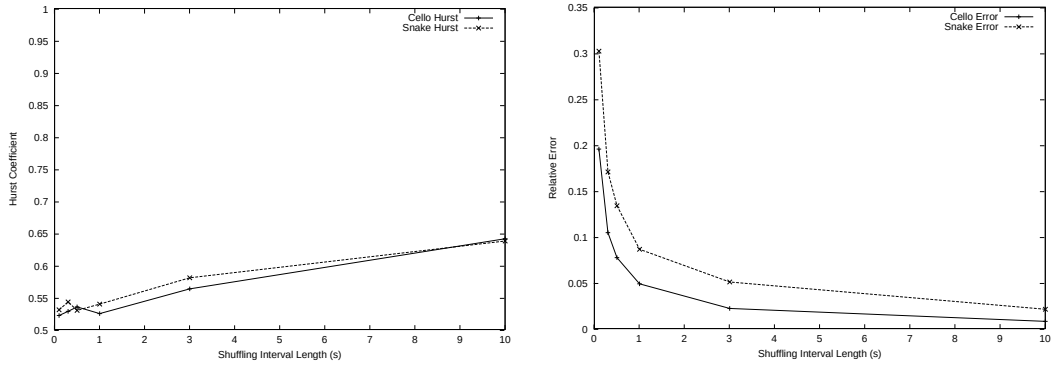
We examine the numerical metric of disk performance that was previously used to validate disk models [15]: the root mean squared (RMS) horizontal distance between the cumulative distribution functions (CDF) of I/O response times. The distributions of queue lengths of traces shuffled at intervals more than one second are similar to those of the real traces [9], and are not presented separately.

We vary the shuffle interval length from 10 seconds to 0.1 second and use both the shuffled and original traces to drive the Pantheon disk simulator [20]. Shuffling traces using intervals smaller than 0.5 second results in significantly skewed disk response time distributions, as shown in Fig-



(a) Cello (b) Snake

Figure 5. Response time distributions for traces shuffled using large intervals.



(a) Hurst coefficient (b) Relative error (RMS)

Figure 6. Effect of shuffling.

ures 3(a) and 3(b). In the limit, we destroy all temporal locality by randomizing all events and the obtained response time distribution curve extremely differs from the original one, as shown in Figure 4. In contrast, although at interval sizes of one second and above there is virtually no long-range dependence left in the shuffled traces, the relative errors are very small, as shown in Figures 5(a) and 5(b).

Shuffling removes long-range dependence in I/O traces. The Hurst coefficient is a measure of long-range dependence; as intuition dictates, the smaller the time interval, the fewer long-range correlations are preserved and the lower the Hurst coefficient, as shown in Figure 6(a). In comparison, the original Hurst coefficient is 0.89 for cello and 0.79 for snake, respectively.

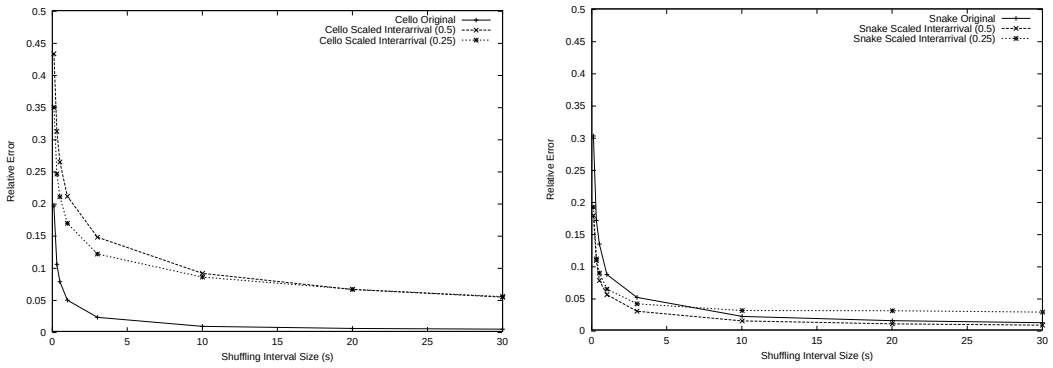
Despite the lack of long-range dependence, particularly indicated by the fluctuation of the Hurst coefficient at small intervals (less than one second), the relative error, measured by RMS [15], for the shuffled traces is relatively small and decreases with the increase of the shuffle interval length, as shown in Figure 6(b). In general the relative error is larger for snake than for cello: at one second it is approximately 5% for cello and 9% for snake. The error for snake can be bounded under 5% by using shuffle intervals larger than five seconds.

3.3. Choosing an Interval

To better understand how to select an appropriate interval length to bound the error for each trace, we study the relative error as a function of workload burstiness. We multiply the traced interarrival times by a factor of 0.5 or 0.25 to artificially increase the burstiness of the traces, creating more disk request queuing.

Figures 7(a) and 7(b) show the effect of scaling on relative error for cello and snake, respectively. In general, as the shuffle interval length increases, the relative error decreases. However, the interval length necessary to maintain the same relative error does not automatically increase as we scale down the traced interarrival time, and the curves are quite different for the two workloads.

Cello has fewer long “gaps” between activities than snake. Almost all of the cello requests have interarrival times less than 0.1 second; in contrast, 63% of the snake requests arrive after the previous one within 0.1 second,



(a) Cello (b) Snake

Figure 7. Effect of shuffling on relative error for scaled traces.

and 7% of the snake requests have interarrival times longer than 10 seconds (see Figure 10). Shuffling has less effect on the request interarrival time distribution for cello than for snake. Cello is less sequential than snake so shuffling does not perturb access spatial locality for cello as much as for snake. Thus, the error caused by shuffling the original trace is lower for cello than for snake.

When the traced interarrival times are shortened, queue lengths increase. For cello, this improves average seek time thanks to request scheduling optimization. Shuffling changes this queuing behavior and causes higher errors than in the original trace. Snake has a disk cache and is more sequential but less busy than cello, resulting in much lower average queue length. Therefore, this queuing effect is not as important for snake. In addition, shortening request interarrival times can, to some degree, preserve more temporal relations in the workload under the same shuffle interval length. Thus, errors for the shuffled scaled snake traces are actually lower than those for the shuffled original trace at small intervals, and increase with larger shuffle intervals.

3.4. Modern Traces

The traces described here were from 1992. A reconfigured cello was re-traced in 1999, but we have not yet been able to repeat our experiments on those traces. However, to generalize our results on long-range dependence to modern traces, we study the characteristics of new cello news disk traces, obtained from a Seagate ST19171W disk, for one week (09/09/1999 to 09/15/1999). The I/O load has increased to about 16 requests per second but the maximum queue length is still in the same range as cello in 1992, from 700 to 1300. The logical sequentiality of cello (1999) is less than 1%. The Hurst coefficient is 0.89, similar to that of the 1992 cello trace. Experiments on the scaled 1992 cello traces (Figure 7) approximate the characteristics of the modern traces with respect to the Hurst coefficient and mean interarrival time. We believe that the shuffle interval required to minimize errors for modern traces may be slightly longer, but otherwise the behavior is the same.

We can conclude from our experiments that long-range dependence does not significantly affect disk behavior with respect to response times. For the purpose of performance evaluation, we need only consider I/O activities at time scales related to the system we are evaluating. For measuring disk response times and queuing behaviors, an appropriate interval length is estimated empirically to be between 3 and 10 seconds for the cello and snake workloads.

4. Binomial Multifractals

Self-similarity is a measure of fractal-like scaling behaviors over multiple time scales, characterized by the single Hurst parameter. In contrast, multifractals are a generalization of monofractal self-similar processes that allow for time-dependent scaling laws, and are based on multiplicative schemes. They have a bursty appearance similar to that of real I/O traffic. We introduce binomial multifractals, for the purpose of modeling I/O traffic, below. A rigorous introduction to binomial measures and multifractals can be found in [13].

4.1. Property of Self-Similarity

We first define a self-similar binomial measure on the unit interval in a recursive construction. Figure 8 shows the first two stages of the construction, which starts with the uniform probability measure μ_0 on the unit interval I = [0, 1] with mass 1 (Figure 8a). At the first stage (Figure 8b), I is split into two equal-length subintervals I_0 = [0, 1/2] and I_1 = [1/2, 1] and the masses $m_0 = p$ $(p > 1/2)$ and $m_1 = 1 - m_0 = 1 - p$ are spread uniformly on them. The density on I_0 and I_1 is $2p$ and $2(1-p)$, respectively. At the second stage (Figure 8c), I_0 is split into two equal-length subintervals I_{00} = [0, 1/4] and I_{01} = [1/4, 1/2] and

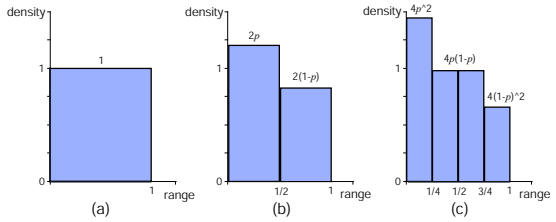


Figure 8. Recursive process in binomial measure generation. Start from (a) with a uniform probability measure, divide the mass with probability p in (b), divide again in (c).

the masses $m_{00} = p^2$ and $m_{01} = p(1-p)$ are spread uniformly on them; I_1 is split into two equal-length subintervals I_{10} = [1/2, 3/4] and I_{11} = [3/4, 1] and the masses $m_{10} = p(1-p)$ and $m_{11} = (1-p)^2$ are spread uniformly on them; The density on I_{00}, I_{01}, I_{10} and I_{11} is $4p^2$, $4p(1-p)$, $4p(1-p)$ and $(1-p)^2$, respectively. This construction continues recursively. Formally, at stage n, $n \in N$, each interval $I_{\varepsilon_1\varepsilon_2\cdots\varepsilon_{n-1}}$ in stage $n-1$ is split into two equal-length subintervals $I_{\varepsilon_1\varepsilon_2\cdots\varepsilon_{n-1}\varepsilon_n}$ with mass $m_{\varepsilon_1}m_{\varepsilon_2}\cdots m_{\varepsilon_{n-1}}m_{\varepsilon_n}$, $\varepsilon_i = 0,1$. Therefore, $\mu(I_{\varepsilon_1\varepsilon_2\cdots\varepsilon_n}) = m_{\varepsilon_1}m_{\varepsilon_2}\cdots m_{\varepsilon_n}$. This defines a sequences of measures μ_n on the unit interval I, which converge weakly towards a probability measure μ, the binomial measure. From the procedure of construction, it is clear that μ is strictly self-similar, as shown in Figure 9.

We extend this construction by randomizing the allocation of the mass in the recursive subdivisions. In this case, we may randomly choose the left multiplier as m_0 or m_1 (each with probability of 0.5), instead of always choosing m_0. Such randomization leads to binomial multifractals and makes them difficult to repeat, analyze, and predict. Real I/O traces could resemble more closely multifractals because of the unpredictability in workloads.

4.2. Property of Burstiness

Roughly speaking, the Hurst coefficient H describes global burstiness. However, local burstiness in disk I/Os is more interesting in practice. Multifractals can represent local burstiness, as described by the local Hölder exponent and multifractal spectrum of binomial measures.

For any $x \in [0,1)$, there is a unique subinterval $I_{\varepsilon_1\varepsilon_2\cdots\varepsilon_n}$ containing it in stage n. Let us denote it as $I^{(n)}(x)$. For convenience, we assume $m_0 > m_1$. For some x, the density on $I^{(n)}(x)$, defined as $\frac{\mu(I^{(n)}(x))}{|I^{(n)}(x)|} = \frac{m_{\varepsilon_1}m_{\varepsilon_2}\cdots m_{\varepsilon_n}}{2^{-n}}$, tends to infinity when $n \to \infty$, as shown by the points in the leftmost subintervals in Figures 8 and 9. Here $\mu(I^{(n)}(x))$ is the mass on $I^{(n)}(x)$ and $|I^{(n)}(x)|$ is the length of $I^{(n)}(x)$. The coarse graining in this interval has the property of burstiness. We can use a singularity exponent, the Hölder exponent, $\alpha(x)$

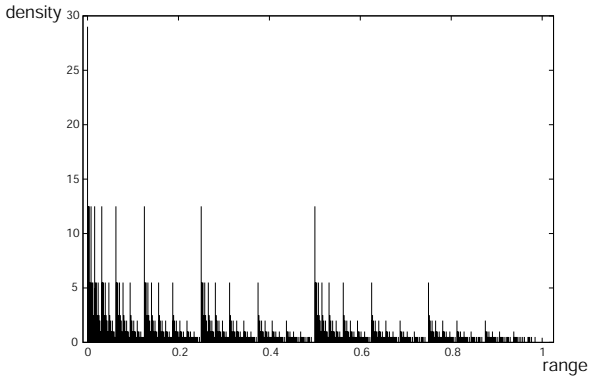


Figure 9. Probability density function for a sample binomial multifractal with bias 0.7.

to describe how fast the value approaches infinity:

$$\begin{aligned}\alpha(x) &= \lim_{n\to\infty}\alpha^{(n)}(x)\\ &= \lim_{n\to\infty}\frac{\log_2\mu(I^{(n)}(x))}{\log_2|I^{(n)}(x)|}\\ &= \lim_{n\to\infty}\frac{\log_2 m_{\varepsilon_1}m_{\varepsilon_2}\cdots m_{\varepsilon_n}}{\log_2 2^{-n}}\\ &= -\lim_{n\to\infty}\frac{\log_2\prod_{i=1}^{n}m_{\varepsilon_i}}{n}. \qquad (1)\end{aligned}$$

The multifractal spectrum $f(\alpha)$ describes the global distribution of Hölder exponent $\alpha(x)$:

$$\begin{aligned}f(\alpha) &= \lim_{n\to\infty}f^{(n)}(\alpha)\\ &= \lim_{n\to\infty}\frac{\log_2 N^{(n)}(\alpha)}{n}, \qquad (2)\end{aligned}$$

where $N^{(n)}(\alpha)$ denotes the number of subintervals $I^{(n)}$ with the Hölder exponent value of α.

At stage n, $\frac{n!}{i!(n-i)!}(= N^{(n)}(\alpha))$ subintervals have the same mass of $m_0^{n-i}m_1^i$. Therefore,

$$\begin{aligned}\alpha^{(n)} &= -(\log_2 m_0^{n-i}m_1^i)/n\\ &= -(i/n)\log_2 m_1 - (1-i/n)\log_2 m_0 \qquad (3)\\ &= (i/n)\alpha_{max} + (1-i/n)\alpha_{min}, \qquad (4)\end{aligned}$$

where $\alpha_{min} = -\log_2 m_0 = -\log_2 p$ and $\alpha_{max} = -\log_2 m_1 = -\log_2(1-p)$. According to the Stirling's formula,

$$\frac{n!}{i!(n-i)!} \sim (2^{-n})^{-E(i/n)}, \qquad (5)$$

where $E(i/n) = -(i/n)\log_2(i/n) - (1-i/n)\log_2(1-i/n)$. Combining above equations, we can find

$$\begin{aligned}f(\alpha) &= -\frac{\alpha_{max}-\alpha}{\alpha_{max}-\alpha_{min}}\log_2\left(\frac{\alpha_{max}-\alpha}{\alpha_{max}-\alpha_{min}}\right) \qquad (6)\\ &\quad -\frac{\alpha-\alpha_{min}}{\alpha_{max}-\alpha_{min}}\log_2\left(\frac{\alpha-\alpha_{min}}{\alpha_{max}-\alpha_{min}}\right),\end{aligned}$$