The Annals of Probability
1997, Vol. 25, No. 1, 329–336

STRING MATCHING BOUNDS VIA CODING[1]

BY PAUL C. SHIELDS

University of Toledo

It is known that the length $L(x_1^n)$ of the longest block appearing at least twice in a randomly chosen sample path of length n drawn from an i.i.d. process is asymptotically almost surely equal to $C \log n$, where the constant C depends on the process. A simple coding argument will be used to show that for a class of processes called the finite energy processes, $L(x_1^n)$ is almost surely upper bounded by $C \log n$, where C is a constant. While the coding technique does not yield the exact constant C, it does show clearly what is needed to obtain $\log n$ bounds.

1. Introduction. Let A denote a finite set and let x_m^n denote the sequence $x_m, x_{m+1}, \ldots, x_n$, where each $x_1 \in A$. Define $L(x_1^n)$ to be the length of the longest block that appears at least twice in x_1^n; that is, $L(x_1^n)$ is the largest integer $L \leq n$ for which there are integers $0 \leq s < t \leq n - L$ such that

$$x_{s+1}^{s+L} = x_{t+1}^{t+L}.$$

The asymptotic behavior of $L(x_1^n)$ for sequences drawn from a stationary, finite-alphabet ergodic process is of interest in DNA modeling. Known results include the following.

1. For any i.i.d. or mixing Markov process there is a constant C, which depends on the process, such that
$$\lim_{n \to \infty} \frac{L(x_1^n)}{\log n} = C \quad \text{almost surely.}$$
2. For any ergodic process with entropy rate H,
$$\liminf_{n \to \infty} \frac{L(x_1^n)}{\log n} \geq \frac{1}{H} \quad \text{almost surely.}$$
3. For any positive function $n \mapsto \lambda(n)$ which is $o(n)$, there is a mixing process and an increasing sequence $\{n_j\}$ such that
$$\lim_{j \to \infty} \frac{L(x_1^{n_j})}{\lambda(n_j)} = +\infty \quad \text{almost surely.}$$

Positive results for i.i.d. and mixing Markov processes are discussed, for example, in [1]. The fact that $1/H$ is lower bound follows easily from results in [5]. The fact that $L(x_1^n)$ can grow at any $o(n)$ rate was established in [10].

Received May 1995; revised May 1996.
[1]Partially supported by NSF Grant DMS-90-24240 and MTA-NSF Project 37.
AMS 1991 *subject classifications.* Primary 60G17; secondary 94A24.
Key words and phrases. String matching, prefix codes.

There is a simple connection between repeated blocks and coding, namely, the second time a block occurs it can be described merely by giving its length and telling where it started earlier. This idea is the basis for several coding procedures, including the well-known Lempel–Ziv coding algorithm used in many standard data compression packages. The main purpose of this paper is to show how this coding idea, in combination with an asymptotic lower bound on code length due to Barron, yields an asymptotic lower bound on the probability a repeated block, conditioned on the past of its second occurrence. This bound immediately shows that $L(x_1^n)/\log n$ is almost surely bounded for processes that have suitable exponential upper bounds on conditional probabilities, thus extending the i.i.d. and Markov upper bound results to a much wider class of processes.

To state the principal result and the consequent string matching bounds precisely, the following notation and terminology will be used. A process μ with alphabet A is a Borel probability measure on the space A^∞ of infinite sequences drawn from A. For each $t \geq 1$, $L \geq 1$, and x_1^{t+L}, let

$$\mu\big(x_{t+1}^{t+L} | x_1^t\big) = \frac{\mu\big(\{y \in A^\infty\colon y_1^{t+L} = x_1^{t+L}\}\big)}{\mu\big(\{y \in A^\infty\colon y_1^t = x_1^t\}\big)}$$

denote the conditional measure on the next L steps, given the first t steps. A sequence x_1^n is said to have a repeated L-block at position $t+1$ if $t \leq n - L$ and there is an s such that $0 \leq s < t$ and $x_{s+1}^{s+L} = x_{t+1}^{t+L}$.

THEOREM 1. *Let μ be stationary process with alphabet A. For each n, let $t(n)$ and $L(n)$ be integer-valued mappings from A^n to $[1, n]$ such that x_1^n has a repeated $L(n)$-block at position $t(n)+1$. Then*

$$\mu\big(x_{t(n)+1}^{t(n)+L(n)} \mid x_1^{t(n)}\big) \geq \frac{1}{n^5} \quad \textit{eventually a.s.} \tag{1}$$

This theorem immediately gives $\log n$ string matching bounds for finite energy processes. An ergodic process μ has finite energy if there are constants $c < 1$ and K such that

$$\mu\big(x_{t+1}^{t+L} \mid x_1^t\big) \leq Kc^L, \tag{2}$$

for all $t \geq 1$, for all $L \geq 1$, and for all x_1^{t+L} of positive measure.

THEOREM 2. *If μ is a finite energy A-valued process, then there is a constant C such that $L(x_1^n) \leq C \log n$, eventually almost surely.*

PROOF. Let $L(x_1^n)$ be the length of the longest string that appears twice in x_1^n and assume $L(x_1^n) \to \infty$. The repeated block bound (1), together with the finite energy bound (2), yields

$$\limsup_{n\to\infty} \frac{L(x_1^n)}{\log n} \leq \frac{5}{-\log c},$$

almost surely, which establishes the theorem. □

Theorem 2 extends previous upper bounding results, since i.i.d. processes and mixing Markov processes have finite energy. Furthermore, functions of mixing Markov chains have finite energy, hence have log n string matching bounds. (Functions of Markov chains are also called hidden Markov chains or finite state processes.) Another type of finite energy process is obtained by adding i.i.d. noise to a given ergodic process, for example, the binary process defined by

$$X_n = Y_n + Z_n \pmod 2$$

where $\{Y_n\}$ is an arbitrary binary ergodic process and $\{Z_n\}$ is binary i.i.d. and independent of $\{Y_n\}$. (Adding noise is often called "dithering.")

REMARK 1. The question of when $L(x_1^n)/\log n$ is almost surely bounded is connected to an entropy estimation algorithm proposed by Grassberger [3], and described as follows. For each infinite sequence $x = \{x_1, x_2, \ldots\}$ and positive integer i, let $x(i) = \{x_i, x_{i+1}, \ldots\}$, the sequence obtained by omitting the first $i - 1$ terms of x. For $i \in [1, n]$, let $L_i^n = L_i^n(x)$ denote the length of the shortest prefix of $x(i)$ that is not a prefix of $x(j)$ for $j \in [1, n]$, $j \neq i$. Grassberger suggested that for any ergodic process of entropy H,

$$\frac{1}{n \log n} \sum_{i=1}^{n} L_i^n(x) \to \frac{1}{H} \quad \text{almost surely.} \tag{3}$$

As shown in [9], there are ergodic processes for which (3) is false, even for convergence in probability, but a trimmed mean result is true; namely, eventually almost surely, εn of the numbers $L_i^n(x)/\log n$ are within ε of $(1/H)$, for any given $\varepsilon > 0$. (An error in the proof of this result is corrected in the Appendix of the present paper.) Consequently, the limit result (3) is true for any ergodic process for which $L(x_1^n)/\log n$ is almost surely bounded.

REMARK 2. After submission of this paper, it was learned that Kontoyiannis and Suhov [4], using a different method in combination with the results of [9], obtained the conclusion of Theorem 2 for processes satisfying the condition that there be an $\alpha > 0$ and an $r \geq 1$, such that the essential infimum of $\mu(x_r \mid x_{-\infty}^0)$ is at least α. Their result is easily derived from Theorem 1 by using their idea to upper bound the left-hand side of (1).

2. The coding method. A code, or more precisely, a faithful n-code, is a one-to-one function C_n from A^n into variable length binary sequences, called code words. Let $l(x_1^n)$ denote the length of the code word assigned to x_1^n. Two facts from coding theory will be used. The first, due to Shannon, is that for any given probability distribution μ on A^n, there is a faithful n-code with length function $l(x_1^n) = \lceil -\log \mu(x_1^n) \rceil$, where $\lceil \cdot \rceil$ denotes the upper integer function. In fact, such codes can be constructed with the prefix property, namely, so that no code word is a prefix of any other code word. Such codes are called prefix codes. Any prefix n-code with length function $\lceil -\log \mu(x_1^n) \rceil$ will be called a Shannon code with respect to μ.

The second coding fact, due to Barron, is that for no process is there a sequence of prefix n-codes whose lengths are shorter than Shannon code lengths by more than $2 \log n$, infinitely often on a set of positive measure.

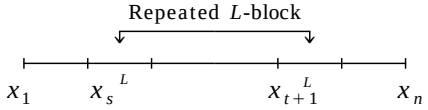
LEMMA 1 (Barron's lemma). *Let μ be a process with alphabet A, and for each n, let C_n be a prefix n-code with length function $l_n(x_1^n)$. For almost every $x \in A^\infty$, there is an $N = N(x)$ such that*

$$l_n(x_1^n) + \log \mu(x_1^n) \geq -2 \log n, \quad n \geq N.$$

Barron's lemma was proved in his thesis [2], but only published much later in [8]. Its proof, along with a sketch of the proof of the existence of Shannon codes, will be given after it is shown how Theorem 1 follows from Barron's lemma.

PROOF OF THEOREM 1. The basic coding idea is to code the second occurrence of a repeated block by telling where it starts, its length, and where it occurred earlier, then use Shannon codes on the parts preceding and following this second occurrence, and compare the total length of this code with the Shannon code length for x_1^n.

To illustrate the coding idea, suppose a block of length L occurs at position s, and then again at position $t + 1 > s$, as indicated in the following figure.



The sequence x_1^n is encoded by first encoding the indices s, t and L. Since these integers are all bounded by n, each can be specified using $\lceil \log n \rceil$ bits. Next, the initial segment x_1^t is encoded using the Shannon code for the measure $\mu(x_1^t)$ and the final segment x_{t+L+1}^n is encoded using the Shannon code for the conditional measure $\mu(x_{t+L+1}^n \mid x_1^{t+L})$. These require $\lceil -\log \mu(x_1^t) \rceil$ and $\lceil -\log \mu(x_{t+L+1}^n \mid x_1^{t+L}) \rceil$ bits, respectively. The code is the concatenation of these five codes, in the order given. In particular, total code length is

$$3 \log n + \log \frac{1}{\mu(x_1^t)} + \log \frac{1}{\mu(x_{t+L+1}^n \mid x_1^{t+L})} \tag{4}$$

bits. Here, as later, the fact that these logarithms should be rounded up to integers is ignored, as only asymptotic results will be of interest.

To see that the code is a prefix code, consider how the decoder, who knows the process μ and n, operates. The first $3 \log n$ bits determine s, t and L. Since t is now known, and Shannon codes are prefix codes known to the decoder, the decoder can recognize that it has a code word after reading the next $-\log \mu(x_1^t)$ bits and decode to get x_1^t. Since s and L are now known, the decoder knows that $x_{t+1+j} = x_{s+j}$, for $1 \leq j \leq L$. (This can be done recursively if it happens that $[s, s + L - 1]$ and $[t + 1, t + L]$ overlap.) Since $t + L + 1$ and x_1^{t+L} are now known, the decoder can determine x_{t+L+1}^n from the final block of bits.

To compare code length with the length of the Shannon code with respect to $\mu(x_1^n)$, first factor $\mu(x_1^n)$ and take the logarithm to obtain

$$\log \mu(x_1^n) = \log \mu(x_1^t) + \log \mu(x_{t+1}^{t+L} \mid x_1^t) + \log \mu(x_{t+L+1}^n \mid x_1^{t+L}).$$

Adding this to the code length (4) produces

$$3 \log n + \log \mu(x_{t+1}^{t+L} \mid x_1^t)$$

and hence Barron's lemma gives

$$3 \log n + \log \mu(x_{t+1}^{t+L} \mid x_1^t) \geq -2 \log n,$$

eventually almost surely, which is just the conclusion, (1), of Theorem 1 expressed in logarithmic form. □

The key to Barron's lemma is an inequality, known as the Kraft inequality, which holds for any prefix code. Recall that a prefix n-code is mapping C from A^n into variable length binary words such that if $C(x_1^n)$ is a prefix of $C(y_1^n)$ then $x_1^n = y_1^n$. The Kraft inequality asserts that

$$\sum_{x_1^n} 2^{-l(x_1^n)} \leq 1. \tag{5}$$

One simple way to prove this inequality is to associate with each x_1^n the dyadic subinterval of the unit interval $[0, 1]$ of length $2^{-l(x_1^n)}$ whose left endpoint has dyadic expansion $C(x_1^n)$. The prefix property implies that these subintervals are disjoint, which implies (5).

By the way, it should be noted here that a kind of partial converse of the Kraft inequality holds, namely, if $l(x_1^n)$ is a positive integer-valued function such that $\sum 2^{-l(x_1^n)} \leq 1$, then each x_1^n can be associated with a dyadic subinterval of the unit interval of length $2^{-l(x_1^n)}$ such that distinct n-sequences correspond to disjoint subintervals. (This is an easy exercise.) A prefix code is then obtained by defining $C(x_1^n)$ to be the dyadic expansion of the left endpoint of the interval associated with x_1^n. In particular, $l(x_1^n) = \lceil -\log \mu(x_1^n) \rceil$ clearly satisfies (5), so there is a prefix code with this length function; in other words, Shannon codes exist.

The standard information theory proofs of these prefix code results first identify the code words of a prefix code with the leaves of a labeled binary tree, then note that (5) holds if and only if $l(\cdot)$ is the depth function of such a tree. See [11], Section 1.7.3.

PROOF OF BARRON'S LEMMA. Fix a sequence $\{c_n\}$ of positive numbers and define

$$B_n = \{x_1^n : l_n(x_1^n) + \log \mu(x_1^n) \leq -c_n\}.$$

By using the relation $l_n(x_1^n) = \log 2^{l_n(x_1^n)}$, the set B_n can be expressed as

$$B_n = \{x_1^n : \mu(x_1^n) \leq 2^{-l_n(x_1^n)} 2^{-c_n}\},$$

and hence

$$\mu(B_n) = \sum_{x_1^n \in B_n} \mu(x_1^n) \leq 2^{-c_n} \sum_{x_1^n \in B_n} 2^{-l_n(x_1^n)}.$$

Since C_n is assumed to be a prefix code, the Kraft inequality (5) implies that $\sum_{x_1^n} 2^{-l_n(x_1^n)} \leq 1$, so that $\mu(B_n) \leq 2^{-c_n}$. With $c_n = 2 \log n$ the sum $\sum \mu(B_n)$ is finite, and hence Barron's lemma follows from the Borel–Cantelli lemma.

REMARK 3. The proof of Barron's lemma actually yields the stronger result that $l(x_1^n) + \log \mu(x_1^n) \geq -c_n$, eventually almost surely, provided $\sum 2^{-c_n} < \infty$, which is the form given by Barron. The prefix code assumption is not really necessary, for headers of asymptotically negligible length can always be added to an invertible code to make it into a prefix code, ([11], Section 1.7.3). It should also be noted that finiteness of the alphabet is not needed, for Barron's lemma and the consequent Theorems 1 and 2 are valid in the countable alphabet case.

APPENDIX

Correction of prior proof. In [9], two errors, one minor and one more serious, were made in the proof of Lemma 3, which is stated here in the following equivalent form.

LEMMA 3 [9]. *If u is ergodic with positive entropy H and if $\varepsilon > 0$, then eventually almost surely there are at most εn indices $i \in [1, n]$ for which $L_i^n(x) > (1 + \varepsilon)\log n/H$.*

The minor error was an incorrect definition of the Ornstein–Weiss return-time function; see equation (4) on page 406. This author did not allow for repetition of a k-block within the first k steps. This is easily repaired by using the correct form of the Ornstein–Weiss return-time function [6]. The more serious error was that the argument starting with the last two lines of page 406 is only valid for the case when $i < j$; in other words, both forward and backward return-time functions are needed.

The proof can be corrected as follows. The forward and backward return-time functions are defined, respectively, by

$$F_k(x) = \min\{m \geq 1: x_{m+1}^{m+k} = x_1^k\},$$

$$B_k(x) = \min\{m \geq 1: x_{-m-k+1}^{-m} = x_{-k+1}^{0}\}.$$

The Ornstein–Weiss recurrence-time theorem yields the forward result,

$$\lim_{k \to \infty} \frac{1}{k} \log F_k(x) = H \quad \text{almost surely.}$$

Likewise, when applied to the reversed process it yields the backward result,

$$\lim_{k \to \infty} \frac{1}{k} \log B_k(x) = H \quad \text{almost surely,}$$

since the reversed process is ergodic and has the same entropy as the original process. These limit results imply that there is a positive integer K and two

sets F and B, each of measure at most $\varepsilon n/4$, such that if $k \geq K$, then

$$\log F_k(x) \geq kH(1+\varepsilon/4)^{-1}, \qquad x \notin F,$$

and

$$\log B_k(x) \geq kH(1+\varepsilon/4)^{-1}, \qquad x \notin B.$$

The ergodic theorem implies that for almost every x there is an integer $N(x)$ such that if $n \geq N(x)$ then $T^{i-1}x \in F$, for at most $\varepsilon n/3$ indices $i \in [1, n]$, and $T^{i-1}x \in B$, for at most $\varepsilon n/3$ indices $i \in [1, n]$. Fix such an x and assume $n \geq N(x)$.

Let k be the least integer that exceeds $(1+\varepsilon/2)\log n/H$. By making n larger, if necessary, it can be assumed that $k \geq K$. Suppose

$$L_i^n = L_i^n(x) > (1+\varepsilon)\log n/H.$$

By making n larger, if necessary, it can be assumed that $L_i^n - 1 \geq k$, and hence, by the definition of L_i^n there is an index $j \in [1, n]$, such that $j \neq i$ and $x_i^{i+k-1} = x_j^{j+k-1}$. The two cases $i < j$ and $i > j$ will be considered separately.

CASE 1. $i < j$. In this case, the k-block starting at i recurs in the future within the next n steps; that is, $F_k(T^{i-1}x) \leq n$, so that

$$\frac{1}{k}\log F_k(T^{i-1}x) \leq \frac{\log n}{k} < H(1+\varepsilon/4)^{-1},$$

which means that $T^{i-1}x \in F$.

CASE 2. $j < i$. If $i + k - 1 \leq n$, this means that $B_k(T^{i+k-2}x) \leq n$, which means that $T^{i+k-2}x \in B$. Otherwise $i + k - 1 > n$; that is, i is within k of n.

In summary, if $L_i^n(x) > (1+\varepsilon)\log n/H$ then $T^{i-1}x \in F$, $T^{i+k-2}x \in B$, or $i + k - 1 > n$. By making n larger, if necessary, it can be supposed that $k \leq \varepsilon n/3$. Thus, there can be at most εn indices $i \in [1, n]$ for which $L_i^n > (1+\varepsilon)(\log n)/H$. This establishes Lemma 3.

REMARK 4. The incorrect definition of the return-time function was noted by Quas [7], as part of his extension of this author's results to the countable alphabet case. The need for both forward and backward return-time functions was discovered by this author while fitting this material into [11], Section II.5.

REFERENCES

[1] ARRATIA, R. and WATERMAN, M. (1989). The Erdős–Rényi strong law for pattern matching with a given proportion of mismatches. *Ann. Probab.* **17** 1152–1169.

[2] BARRON, A. (1985). Logically smooth density estimation. Ph.D. dissertation, Dept. Electrical Engineering, Stanford Univ.

[3] GRASSBERGER, P. (1989). Estimating the information content of symbol sequences and efficient codes. *IEEE Trans. Inform. Theory* **35** 669–675.

[4] KONTOYIANNIS, I. and SUHOV, Y. M. (1993). Prefixes and the entropy rate for long-range sources. In *Probability, Statistics, and Optimization* (F. P. Kelly, ed.). Wiley, New York.
[5] ORNSTEIN, D. and WEISS, B. (1990). How sampling reveals a process. *Ann. Probab.* **18** 905–930.
[6] ORNSTEIN, D. and WEISS, B. (1993). Entropy and data compression. *IEEE Trans. Inform. Theory* **IT-39** 78–83.
[7] QUAS, A. (1995). An entropy estimator for a class of infinite alphabet processes. Preprint.
[8] SHIELDS, P. (1990). Universal almost sure data compression using Markov types. *Problems Control Inform. Theory* **19** 269–277.
[9] SHIELDS, P. (1992). Entropy and prefixes. *Ann. Probab.* **20** 403–409.
[10] SHIELDS, P. (1992). String matching—the general ergodic case. *Ann. Probab.* **20** 1199–1203.
[11] SHIELDS, P. (1996). *The Ergodic Theory of Discrete Sample Paths. AMS Graduate Studies in Mathematics* **13**. Amer. Math. Soc., Providence, RI.

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF TOLEDO
TOLEDO, OHIO 43606
E-MAIL: pshield2@uoft02.utoledo.edu