Normal Numbers

Contents

1 Motivation **1**

2 Definitions **1**

3 Finite-State Compressors and Normal Sequences **3**

3.1 Definitions 3

3.2 Normal implies finite-state incompressible 4

3.3 Non-normal implies finite-state compressible 4

4 Champernowne-like Sequences **5**

1 Motivation

2 Definitions

Fix the alphabet $\Sigma = \{0, 1, \ldots, b-1\}$, $b \geq 2$.

Definition 2.0.1. An sequence $X \in \Sigma^\infty$ is said to be *simply normal to the base b* if

$$\lim_{n\to\infty} \frac{N(X[0\ldots n-1], k)}{n} = \frac{1}{b},$$

for any $k \in \Sigma$.

That is, any digit occurs in X with limiting frequency $\frac{1}{b}$.

Definition 2.0.2 (Hardy, Niven-Zuckerman)**.** A sequence $X \in \Sigma^\infty$ is said to be *normal to the base b* if for any finite string $w \in \Sigma^*$, we have

$$\lim_{n\to\infty} \frac{N(X[0\ldots n-1], w)}{n - |w| + 1} = \frac{1}{b^{|w|}}.$$

That is, any pattern w occurs with the right frequency in X.

We now show an equivalence theorem due to Pillai. The proof is a simplification of Pillai's original proof, due to John Maxfield.

Theorem 2.0.3. *A sequence is normal to base b if and only if it is simply normal to bases b^k, $k \in \mathbb{N}^+$.*

Proof. It is easy to see that if X is normal to the base b, then it is simply normal to bases b, b^2, ...

We now show the converse. Let $X \in \Sigma^\infty$ be simply normal in bases b, b^2, ...

Let w be an n-long pattern, where $n = kb - r$, with $k \geq 0$ and $0 \leq r \leq b - 1$. We now consider the frequency of occurrences of w in X.

First, consider the number of occurrences of w in the collection of sequences of length kb. After w is fixed, there are r free positions in the sequence. w itself can occur in positions 0, 1, ..., r. Hence there are $(r+1)b^r$ different occurrences of w in Σ^{kb}.

First, parse the infinite sequence X into disjoint blocks of length $2kb$. Consider occurrences of w which occur properly inside each blocks, and ignoring for now, the occurrences which span across 2 blocks. The frequency is

$$N(w, 1) = \frac{(r+1)b^r}{kb\ b^{kb}},$$

since there are kb digits of X to the base b to correspond to each digit of the base b^{kb}, and X is simply normal to the base b^{kb}.

Now, consider the occurrences which span 2 blocks. (*i.e.* the block length is now $4kb$.) To count these occurrences, imagine that a single block is extended by $n - r$ bits. Then the spanning occurrences will occur in the positions $kb - s$ to $kb - s + n$. This leaves $kb + n - s = 2kb + s$ positions free. Thus the number of patterns is $(n-1)b^{2kb+s}$. The proportion of these occurrences with respect to the total number of patterns is

$$\frac{(n-1)b^{2kb+s}}{kb\ b^{kb}} = \frac{n-1}{2kb^{n+1}}.$$

Thus the total number of occurrences is

$$N(w, 2) = \frac{1}{b^n} - \frac{n-1}{kb^{n+1}} + \frac{n-1}{2kb^{n+1}}$$

In general,

$$N(w,m) = \frac{1}{b^n} - \frac{n-1}{kb^{n+1}} + \frac{n-1}{kb^{n+1}} \left[\sum_{k=1}^{m-1} \frac{1}{2^k} \right] \tag{1}$$

Thus

$$\lim_{m \to \infty} N(w,m) = \frac{1}{b^n}, \text{ where } |w| = n.$$

Now, $N(w,X)$ is $\lim_{m\to\infty} N(w,m)$ (why?). Hence X is normal to the base b. □

3 Finite-State Compressors and Normal Sequences

In his 1948 Ph.D. thesis under the supervision of D. H. Lehmer, D. D. Wall showed that if X is a normal sequence, and L is an infinite arithmetic progression, then the subsequence $\langle X_k \rangle_{k \in L}$ is a normal sequence as well. The proof uses a characterization of normality based on exponential sums, called Weil's criterion.

In 1966, Agofanov showed a remarkable result, generalizing the result of Wall, that if X is a normal sequence, and L is a regular subset of natural numbers (*i.e.* L can be decided by a finite-state automaton), then the subsequence $\langle X_k \rangle_{k \in L}$ is also normal. Unfortunately, the proof is not clear from his paper, where he claims that the result follows from the ergodic theorem for stationary Markov chains.

In an attempt to reproduce Agofanov's result, in 1972, Schnorr and Stimm formally established the connections between finite-state automata and normality. They showed that a sequence is normal if and only if it is incompressible by lossless finite-state compressors. This is a useful tool in the study of normality - to establish that a sequence is normal, it is sufficient to show that no finite-state compressor can compress it. In the next section, we take this approach to show that a specific construction is normal. We now prove the result of Schnorr and Stimm, following the treatment in a 2014 paper by Becher and Heiber.

3.1 Definitions

Definition 3.1.1. A *finite-state compressor* C is a 6-tuple $(Q, \mathcal{A}, \mathcal{B}, q_0, \delta, o)$ where Q is a finite set of states, $\mathcal{A}$ is the finite input alphabet, $\mathcal{B}$ is the finite output alphabet, q_0 is the initial state, $\delta : Q \times \mathcal{A} \to Q$ is the transition function, and $o : Q \times \mathcal{A} \to \mathcal{B}$ is the output function.

Definition 3.1.2. We say that a finite-state compressor C is *lossless* if every pair of (final state, output string) uniquely determines the input string.

Thus losslessness is a criterion which says that we can decompress the string from the output string, and the output state.

3.2 Normal implies finite-state incompressible

3.3 Non-normal implies finite-state compressible

Let $X \in \mathcal{A}^\infty$ be non-normal. We construct a lossless finite-state compressor that compresses X.

Since X is non-normal, it is not simply normal to some base $\mathcal{A}^n$. Hence for some $x \in \mathcal{A}^n$,

$$\lim_{n\to\infty} \frac{N(x, X^n[1 \dots k])}{k} \neq \frac{1}{|\mathcal{A}^n|}. \tag{2}$$

Thus, either the lim sup or the lim inf of the above ratio sequence is different from $\frac{1}{|\mathcal{A}^n|}$.

We define a sequence of positions $(i_k)_{k\in\mathbb{N}}$ relative to this block-size n such that for each $y \in \mathcal{A}^n$, the limiting frequency of y,

$$f_y = \lim_{n\to\infty} \frac{N(x, X^n[1 \dots i_k])}{i_k}$$

exists. The difficulty is to ensure that all alphabets simultaneously have a well-defined limiting frequency in the subsequence of X. In the subsequence, $f_x \neq \frac{1}{|\mathcal{A}^n|}$.

We select the subsequence as follows. First, collect an infinite sequence $(i^(1)_k)_{k\in\mathbb{N}}$ of $\mathbb{N}$ such that f_x is well-defined and different from $\frac{1}{|\mathcal{A}^n|}$. Since either the lim sup or the lim inf of the ratio sequence in (2) is different from $\frac{1}{|\mathcal{A}^n|}$, such a subsequence exists. Then construct a subsequence $(i^(2)_k)_{k\in\mathbb{N}}$ of $(i^(1)_k)_{k\in\mathbb{N}}$ to fix the limiting frequency of some alphabet $y_1 \in \mathcal{A} - \{x\}$. Since the selection of this subsequence will not alter f_x, we now have a subsequence where both f_x and f_{y_1} are fixed. Continue in this manner until we get a subsequence $(i_k)_{k\in\mathbb{N}}$ where all alphabets have a limiting frequency.

Now, we show that X is compressible, by examining it at prefixes with lengths chosen according to the subsequence above. We compress m-length blocks of symbols from the alphabet $\mathcal{A}^m$. Define C_m that maps m-length blocks from $\mathcal{A}^n$ to compressed codewords as follows.

Let C_m be such that for each state $q \in (\mathcal{A}^n)^{<m}$ and input symbol $z \in \mathcal{A}^n$,

$$\delta(q,z) = \begin{cases} qz & \text{if } |q| < m-1 \\ \lambda & \text{if } |q| = m-1. \end{cases}$$

and

$$o(q,z) = \begin{cases} \lambda & \text{if } |q| < m-1 \\ \overline{o}(qz) & \text{if } |q| = m-1, \end{cases}$$

where $\overline{o} : (\mathcal{A}^n)^m \to \mathcal{B}^*$ is a Huffman code into a prefix-free subset of $\mathcal{B}^*$ such that

$$\overline{o}(z) = \left\lceil \sum_{i=1}^{m} -\log_{|\mathcal{B}|} f_{z_i} \right\rceil.$$

Such a coding exists. Thus C_m is lossless.

Now, we have

$$\begin{aligned} \rho_{C_m}(X^n) &= \liminf_{k\to\infty} \frac{|C_m(X^n[1\ldots k])|}{k \log_{|\mathcal{B}|}(|\mathcal{A}^n|)} \\ &\le \lim_{k\to\infty} \frac{|C_m(X^n[1\ldots i_k])|}{i_k \log_{|\mathcal{B}|}(|\mathcal{A}^n|)} \\ &\le \lim_{k\to\infty} \frac{(i_k)/m + \sum_{y\in\mathcal{A}^n} N(y, X^n[1\ldots i_k])(-\log_{|\mathcal{B}|} f_y)}{i_k \log_{|\mathcal{B}|}(|\mathcal{A}^n|)} \\ &\le \frac{1}{m \log_{|\mathcal{B}|}(|\mathcal{A}^n|)} + \frac{\sum_{y\in\mathcal{A}^n} f_y(-\log_{|\mathcal{B}|} f_y)}{\log_{|\mathcal{B}|}(|\mathcal{A}^n|)} \end{aligned}$$

Since $f_x \neq |\mathcal{A}|^{-n}$, using the fact that the uniform distribution on $\mathcal{A}^n$ is the unique distribution with maximal entropy, we know that

$$\frac{\sum_{y\in\mathcal{A}^n} f_y(-\log_{|\mathcal{B}|} f_y)}{\log_{|\mathcal{B}|}(|\mathcal{A}^n|)} < 1.$$

Hence for large m, $\rho_{C_m}(X) < 1$.

4 Champernowne-like Sequences

A simple example of a normal sequence in base 2 is

$$S = 0.0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \ldots$$

i.e. the concatenation of the binary strings in the standard enumeration order. This is related to the Champernowne constant, which in base 10 is

$$.1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, \ldots$$

i.e. the concatenation of the base-10 representations of the positive integers. Champernowne showed that this constant is normal. We show that the related sequence S is normal to the base 2.

It suffices to show that no lossless finite-state compressor with the binary input alphabet can compress S, since this implies that S is normal to the base 2.

Let

$$S[1 \ldots n] = S[1 \ \ldots \ n_1], S[n_1 + 1 \ \ldots \ n_2], \ldots, S[n_{c-1} + 1 \ \ldots \ n_c]$$

be a parsing of S into distinct substrings. Let $\lfloor \log_2 n \rfloor = \ell$. Then

$$c = \sum_{j=1}^{\ell} 2^j + \tau = 2^{\ell+1} - 1 + \tau,$$

where $\tau \in [0, 2^{\ell+1})$ is the number of distinct substrings having length $\ell + 1$. We know that n may be written as

$$n = \sum_{j=1}^{\ell} j2^j + (\ell + 1)\tau = (\ell - 1)2^{\ell+1} + (\ell + 1)\tau,$$

whence we get

$$\begin{aligned} c \log c &= (2^{\ell+1} - 1 + \tau) \log \left(2^{\ell+1} - 1 + \tau\right) \\ &\geq 2^{\ell+1}(\ell + 1) + (\ell + 1)\tau - O(\ell) \\ &\geq n - O(\ell) \end{aligned} \tag{3}$$

Let $C = (Q, A, B, q_0, \delta, o)$ be a lossless finite-state compressor, with s states. Note that B contains λ as a symbol. Let

$$Y[1 \ldots n] = Y[1 \ \ldots \ n_1], Y[n_1 + 1 \ \ldots \ n_2], \ldots, Y[n_{c-1} + 1 \ \ldots \ n_c]$$

be the corresponding output substrings that C emits on S, where some characters may be λ. Let $L(Y[1 \ldots n])$ denote the number of non-λ characters in $Y[1 \ldots n]$. Even though the input phrases are distinct, the output

phrases may not be distinct. However, we know that for any output string w, the triple (start state, end state, w) uniquely determines the input substring from which w was produced.

Thus each output substring corresponds to at most s^2 input strings. It follows that the number of different input phrases which correspond to j-length outputs is $s^2 2^j$.

We prove that

$$L(Y[1 \dots n]) \geq n - o(n),$$

adapting a proof idea by Lempel and Ziv in their paper on universal compression.

The key idea in the proof is to group together output phrases of the same length[1]. Let c_j denote the number of output phrases of the same length, and k be the length of the longest output phrase (k may not be ℓ, the length of the longest input phrase). Then

$$L(Y[1 \dots n]) = \sum_{j=1}^{k} c_j j. \tag{4}$$

We also have $\sum_{j=1}^{k} c_j = c$, the number of distinct input phrases.

We want a lower bound on $L(Y[1 \dots n])$. Hence we can assume that all strings of shorter lengths appear, *i.e.* letting $c_j = 2^j$ for $j < k$, while ensuring that $\sum_{j=1}^{k} c_j = c$. This adjustment can only lower $L(Y[1 \dots n])$, hence suffices for the lower bound. We have

$$L(Y[1 \dots n]) \;\geq\; \sum_{j=1}^{k-1} 2^j s^2 j + \left[c - \sum_{j=1}^{k-1} c_j \right] (k+1) s^2, \tag{5}$$

since $c_j \leq 2^j s^2$. The last term is a lower bound on the contribution of $k+1$ length output strings.

We proceed as follows.

1. The first term is $(k-1)2^{k+1}s^2$.
2. We then estimate the second term.
3. Finally, we express k in terms of n to get the estimate on $L(Y[1 \dots n])$ in terms of s and n.

[1] ignoring the λs

To estimate the second term, let $c = qs^2 + r$, where $0 \leq r < s^2$ and $q = \sum_{j=0}^{k} 2^j + R_k$, where $0 \leq R_k < 2^{k+1}$. Then

$$c = \left(\sum_{j=0}^{k} 2^j + R_k \right) s^2 + r = \sum_{j=0}^{k} 2^j s^2 + R_k s^2 + r. \tag{6}$$

Substituting this in the second term in (5), we get the following bound.

$$\left[c - \sum_{j=1}^{k-1} c_j \right] (k+1)s^2 = (R_k s^2 + r)(k+1)s^2 = (k+1)R_k s^4 + (k+1)rs^2 \tag{7}$$

Using this estimate in (5) yields

$$L(Y[1 \ldots n]) \geq (k-1)2^{k+1}s^2 + (k+1)R_k s^4 + (k+1)rs^2, \tag{8}$$

where $0 \leq R_k < 2^{k+1}$ and $0 \leq r < s^2$.

Set

$t = \frac{c - (\sum_{j=0}^{k} j2^j)}{s^2} = R_k - 1 - (r/s^2)$.

Substituting in (8), we get

$$\begin{aligned} L(Y[1 \ldots n]) &\geq (k-1)s^2[2^{k+1} + t] + s^2(k+3+2t) \\ &\geq (k-1)s^2[c + s^2] + 2s^2(t+2). \end{aligned}$$

We now try to express k in terms of s and c so that the final expression becomes independent of the number of output phrases. From the equation that $c = s^2(2^{k+1} + t)$, the following holds.

$$\begin{aligned} k - 1 &= \log \frac{c - s^2 t}{s^2} - 2 \\ &= \log \frac{c + s^2}{4s^2} - \log \left[1 + \frac{(t+1)s^2}{c - s^2 t} \right] \quad \text{[multiplying and dividing by } c + s^2\text{]} \end{aligned}$$

which when substituted in the inequality above yields

$$L(Y[1 \ldots n]) \geq (c + s^2) \left(\log \frac{c + s^2}{4s^2} + \rho \right). \tag{9}$$

where $\rho > 0$.

By the relation (3), we get

$$L(Y[1 \dots n]) \geq n - O(\ell) - c \log(4s^2) \geq n - O(\ell) - 2c,$$

assuming that $s \geq 1$. Since $c = O(\log n)$, and $\ell = O(\log n)$, we get $L(Y[1 \dots n-1]) \geq n - O(\log n)$. This proves the result.