A Preadapted Universal Switch Distribution for Testing Hilberg’s Conjecture

Łukasz Dębowski

Abstract

Hilberg’s conjecture states that the mutual information between two adjacent long blocks of text in natural language grows like a power of the block length. The exponent in this hypothesis can be upper bounded using the pointwise mutual information computed for a carefully chosen code. The bound is the better, the lower the compression rate is but there is a requirement that the code be universal. In this paper, we introduce two novel universal codes, called the plain switch distribution and the preadapted switch distribution. The advantage of the switch distributions is that they both achieve a low compression rate and are guaranteed to be universal. Using the switch distributions we obtain that the exponent in Hilberg’s conjecture is ≤ 0.83, which improves over the previous bound ≤ 0.94 obtained using the Lempel-Ziv code.

Keywords: universal coding, natural language, Hilberg’s conjecture

---

Ł. Dębowski is with the Institute of Computer Science, Polish Academy of Sciences, ul. Jana Kazimierza 5, 01-248 Warszawa, Poland (e-mail: ldebowski@ipipan.waw.pl).
I Introduction

Hilberg’s conjecture is a hypothesis concerning natural language which states that the mutual information between two adjacent long blocks of text grows very fast, namely as a power of the block length \([1, 2, 3, 4, 5, 6, 7]\). There are two important information-theoretic results concerning this conjecture. On the one hand, Hilberg’s hypothesis can be linked with the idea that texts in natural language refer to large amounts of randomly accessed knowledge in a repetitive way \([8, 9]\). On the other hand, Hilberg’s hypothesis can be linked with the fact that the number of distinct words in a text grows as a power of the text length \([10, 8]\), the fact known as Herdan’s or Heaps’ law \([11, 12]\). These two results make Hilberg’s conjecture interesting and worth direct empirical testing.

In this paper we want to improve on the method for testing Hilberg’s conjecture proposed in \([13]\). Let us introduce some formal notation to accompany our reasoning. Consider a probability space \((\Omega, \mathcal{F}, Q)\) with \(\Omega = \{0, 1, \ldots, D-1\}\), random variables \(X_k : \Omega \ni (x_i)_{i \in \mathbb{Z}} \mapsto x_k \in \{0, 1, \ldots, D-1\}\), and distribution \(Q\) which is stationary on \((X_i)_{i \in \mathbb{Z}}\) but not necessarily ergodic. Blocks of symbols or variables are denoted as \(X_{m:n} = (X_i)_{n \leq i \leq m}\) with \(X_{m:n}\) being the empty block for \(m < n\). Moreover, for a random variable \(X\) we introduce a random variable \(Q(X)\) which takes value \(Q(X = x)\) for \(X = x\). The pointwise entropy of variable \(X\) is the random variable \(H^Q(X) = -\log Q(X)\) whereas the pointwise mutual information between \(X\) and \(Y\) is

\[
I^Q(X; Y) = -\log Q(X) - \log Q(Y) + \log Q(X, Y).
\]

Having this in mind, Hilberg’s conjecture states that

\[
I^Q(X_{1:n}^n; X_{2:n+1}^{2n}) \propto n^\beta, \quad \beta \in (0, 1).
\]

Hilberg \([1]\) supposed that \(\beta \approx 0.5\) holds for texts in English but his estimate was very rough, based on the results of Shannon’s experiment \([14]\). It is an interesting open question how much the exponent \(\beta\) varies across different texts and whether it is possibly a text-independent language universal.

Before we try to answer how to estimate \(\beta\), it is important to note that the discussion of Hilberg’s hypothesis is intimately connected with the question whether the natural language production is nonergodic and what is its right ergodic decomposition. According to the ergodic decomposition theorems \([15, 16, 8]\), any stationary measure \(Q\) equals the expectation \(E_Q F\), where \(F = Q(\cdot|I)\) is the random ergodic measure for measure \(Q\) and \(I\) is the shift-invariant algebra. There also exist stationary nonergodic measures \(Q\), called Santa Fe processes, for which the mutual information \(I^Q(X_{1:n}^n; X_{2:n+1}^{2n})\) grows according to a power law but its main contribution comes from identifying the random ergodic measure \(F\) given the block \(X_{1:n}^n\) \([8, 9]\). In that case mutual information \(I^F(X_{1:n}^n; X_{2:n+1}^{2n})\) for the random ergodic measure \(F\) itself is negligibly small. The Santa Fe processes are not irrelevant for our discussion. In their original construction they were intended as some idealized models for the transmission of knowledge in natural language. Thus when estimating the exponent in Hilberg’s conjecture we have first to decide whether we do it for measure \(Q\) or for measure \(F\). The methods
for estimating $I^Q(X^n_1; X^{2n}_{n+1})$ and $I^F(X^n_1; X^{2n}_{n+1})$ are very different. We suppose that estimating the mutual information for the nonergodic measure $Q$ is closer to the original intention of Hilberg although some philosophical problem remains ‘how many’ ergodic components we actually admit (e.g., do we assume that $Q$ models texts in a given register of a particular language or in any register of any natural language).

In contrast to the random ergodic measure $F$, the possibly nonergodic measure $Q$ is not identifiable given a single realization $(X_i)_{i \in \mathbb{Z}}$. Despite that, it is somewhat baffling that Hilberg’s exponent $\beta$, a property of measure $Q$, can be partly learned from a single realization $(X_i)_{i \in \mathbb{Z}}$ if the growth of mutual information is uniform in $Q$. As we shall see, the later condition seems a reasonable assumption according to the empirical data. Under this important condition, determining the exponent $\beta$ is a problem similar in flavor to finding the entropy of an empirical process or the Kolmogorov complexity of a string. Namely, we can find an upper bound of $\beta$ but we cannot evaluate a nontrivial lower bound.

The help comes from universal coding. Here we say that a distribution $P$ is weakly universal if for every stationary distribution $Q$ we have

$$
\lim_{n \to \infty} \frac{1}{n} E_Q H^P(X^n_1) = h_Q,
$$

where the entropy rate $h_Q$ is

$$
h_Q := \lim_{n \to \infty} \frac{1}{n} E_Q H^Q(X^n_1) = \inf_{k \in \mathbb{N}} E_Q \left[ -\log Q(X_{k+1} | X^n_1) \right].
$$

On the other hand, the distribution $P$ is called strongly universal if for every stationary ergodic distribution $Q$ we have $Q$-almost surely

$$
\limsup_{n \to \infty} \frac{1}{n} H^P(X^n_1) \leq h_Q.
$$

Strongly universal distributions are weakly universal under mild conditions [17]. Dębowski [13] proposed to investigate the empirical law of form

$$
I^P(X^n_1; X^{2n}_{n+1}) \propto n^\gamma, \quad \gamma \in (0, 1),
$$

where $P$ is a weakly universal distribution. Relationship (5), which can be called the codewise Hilberg conjecture, has been checked experimentally for the Lempel-Ziv code on a sample of 10 texts in English and it holds surprisingly uniformly with $\gamma = 0.94$ [13]. The same estimate $\gamma = 0.94$ has been obtained for 21 other texts in German and French (work under review).

Are laws (5) and (7) related? In fact, if they hold uniformly for large $n$ then exponents $\beta$ and $\gamma$ can be linked. To see it, the following lemma is helpful.

**Lemma 1** ([8]) Consider a function $G : \mathbb{N} \to \mathbb{R}$ such that $\lim_k G(k)/k = 0$ and $G(n) \geq 0$ for all but finitely many $n$. For infinitely many $n$, we have $2G(n) - G(2n) \geq 0$.

If $P$ is weakly universal, the above statement is satisfied for Kullback-Leibler divergence

$$
G(n) = E_Q \left[ H^P(X^n_1) - H^Q(X^n_1) \right].
$$

2
Hence we obtain that

\[ E_Q I^P(X_1^n; X_{n+1}^{2n}) \geq E_Q I^Q(X_1^n; X_{n+1}^{2n}) \]  

holds for infinitely many \( n \). Thus if relationships (3) and (7) hold uniformly for large \( n \) then

\[ \gamma \geq \beta. \]  

In other words, the smaller \( \gamma \) we observe for a text (or methodologically better, for a large sample of different texts), the better bound it gives for \( \beta \). It can be also easily shown that the bound is the tighter, the smaller compression rate \( H^P(X_1^n)/n \) is, with the sole provision that distribution \( P \) be weakly universal. Results of our experiment suggest that this requirement is essential.

Thus the question of bounding Hilberg’s exponent \( \beta \) boils down, if we assume the uniform information growth in (3), to finding appropriate universal distributions. Many methods have been proposed for compression of texts in natural language, e.g.: Lempel-Ziv (LZ) code [18], \( n \)-gram models [19, 20, 21], prediction by partial match (PPM) [22], context tree weighting (CTW) [23], probabilistic suffix trees (PST) [24], grammar-based codes [25], PAQ codes [26], and switch distributions [27]. These compression schemes can be divided into two classes: (a) preadapted distributions, which are trained on large corpora and achieve low compression rate—as low as 0.88 bpc (bits per character) for WinRK 3.1.2\(^1\) and (b) adaptive distributions, which are not pre-trained and achieve larger compression rate but are proven to be universal. Whereas the distributions proposed so far belong either to class (a) or (b), for testing Hilberg’s conjecture, we need a distribution that would combine the advantages of classes (a) and (b), namely low compression rate and universality.

In this paper we propose and investigate two novel universal distributions, one of which is not preadapted and the other is preadapted. The point of our departure is a modification of the switch distributions proposed in [27]. The idea of a switch distribution is to use a mixture of adaptive Markov chains of varying orders but, at each data point, the probabilities are partly transferred among different orders. In this way, lower order Markov chains are used to compress the data exclusively until enough information is gathered to predict new outcomes with higher order chains. This leads to much better compression than while the weights are fixed [27]. If we tamper a bit with the definition of the switch distribution we obtain a universal compression scheme. This scheme will be called the plain switch distribution. It is not preadapted yet. The preadapted switch distribution is obtained by initializing the Markov chains with frequencies coming from a large corpus and letting them gradually adapt to the compressed source. It will be shown that both switch distributions are universal. For the considered input text they achieve almost the same ultimate compression rate 2.21 bpc, approximately twice smaller than for the LZ code. This figure is not so favorable as for the WinRK 3.1.2 but we have a guarantee that the switch distributions are universal.

Once we have constructed the universal switch distributions, we can use them for testing the codewise Hilberg conjecture (7). In the previous paper [13], the LZ code was used for a sample of texts in English which yielded \( \gamma = 0.94 \).

\(^1\)http://www.maximumcompression.com/data/text.php
Here using the plain switch distribution we obtain a slightly tighter bound \( \gamma = 0.83 \). Surprisingly, the preadapted switch distribution yields almost the same compression rate for long blocks as the plain switch distribution and does not give a tighter bound for \( \gamma \). Differences in the estimates of \( \gamma \) may also stem from differences in data representation. In \[13\] the alphabet of \( D = 27 \) symbols was used. Here we use \( D = 256 \) and obtain \( \gamma = 0.89 \) for the LZ code. It is important to underline that all estimates of \( \gamma \) can be only obtained using universal distributions. As we show, if a nonuniversal distribution is used, the pointwise mutual information can be very low despite a good-looking compression rate. To a certain extent this also applies to the preadapted switch distribution, where the pointwise mutual information is low for short blocks.

The organization of the paper is as follows. In Section II, we present the plain switch distribution. In Section III, we discuss the preadapted switch distribution. In Section IV, we test Hilberg’s conjecture using the introduced distributions.

II The plain switch distribution

The frequency of substring \( w_k^1 \in \{0, 1, \ldots, D - 1\}^k \) in string \( z^n_1 \in \{0, 1, \ldots, D - 1\}^n \) will be denoted as

\[
c(u^k_1 | z^n_1) = \sum_{i=0}^{n-k} 1\{w_i^k = z_{i+1}^1\}.
\]

(11)

The plain switch distribution is defined as follows:

**Definition 1 (plain switch distribution)** Define conditional probabilities

\[
B(x_{n+1}|x^n_1, -1) = D^{-1}
\]

(12)

Let coefficients \( p_n \in (0, 1) \), where \( n = 1, 2, \ldots \), satisfy \( \prod_{n=0}^{\infty} p_n > 0 \). Put also \( q_n = 1 - p_n \). We define the partial switch distribution \( P(x^n_1, k) \) by conditions

\[
P(x_{1}, -1) = p_0 B(x_1| -1),
\]

(13)

\[
P(x_{1}, 0) = q_0 B(x_1|0),
\]

(14)

\[
P(x^n_1, k) = 0 \text{ for } k < -1 \text{ or } k \geq n,
\]

(15)

\[
P(x^n_1, k) = [p_n P(x^n_1, k) + q_n P(x^n_1, k - 1)] B(x_{n+1}|x^n_1, k)
\]

for \( n \geq 1 \) and \(-1 \leq k \leq n\). 

(16)

The total probability for block \( x^n_1 \) according to the switch distribution is

\[
P(x^n_1) = \sum_{k=-1}^{n-1} P(x^n_1, k).
\]

(17)

The scheme of computing \( P(x^n_1) \) is depicted in Figure \[4\].
Remark 1: Condition $\prod_{n=0}^{\infty} p_n > 0$ holds for instance if we fix

$$p_n = \exp\left[-(n + 1)^{-\alpha}\right], \quad \alpha > 1.$$ (18)

Value $\alpha$ is a parameter.

Remark 2: Probability $B(x_{n+1}^n | x_1^n, k)$ defines an adaptive $k$-th order Markov model. Probability $P(x_1^n, k)$ represents the mass of the adaptive $k$-th order Markov model modified by communication with models of lower orders. The motivation for this communication, carried out in formula (16), is that lower order Markov models should be solely used for compression until enough data are collected to predict new outcomes with higher order Markov models, cf., the ‘catch-up phenomenon’ described in [27]. Distribution $P(x_1^n)$ is a special case of the general scheme of switch distributions considered by [27] to overcome the ‘catch-up phenomenon’. In contrast to the models discussed in [27], the switch distribution considered here is universal and still can be efficiently computed.

Now we will show that the switch distribution (17) is both strongly and weakly universal. First we need this simple fact:

**Theorem 1** Introduce notation

$$B(x_1^n | x_{i-k}^{i-1}, k) = \prod_{i=l}^{n} B(x_i | x_{i-1}^{i-1}, k).$$ (19)

The switch distribution satisfies the following:
i) there exists a constant $\delta_1 > 0$ such that for all $n \geq 1$ we have

$$P(x_1^n) \geq \delta_1 D^{-n}, \quad (20)$$

ii) for each $k \geq 0$ there exists a constant $\delta_k > 0$ such that for all $n \geq k + 1$ we have

$$P(x_1^n) \geq \delta_k B(x_{k+1}^n|x_1^k,k). \quad (21)$$

Proof: For $n \geq 1$ we have

$$P(x_1^n) \geq \left( \prod_{i=0}^{n-1} p_i B(x_{i+1}|x_1^i, -1) \right) \geq \delta_1 P(x_1^n) - 1. \quad (22)$$

where $\delta_1 = \prod_{i=0}^{\infty} p_i > 0$. Thus we have claim (i). On the other hand, for $k \geq 0$ and $n \geq k + 1$ we obtain

$$P(x_1^n) \geq \left( \prod_{i=0}^{k} q_i B(x_{i+1}|x_1^i, i) \right) \left( \prod_{i=k+1}^{n-1} p_i B(x_{i+1}|x_1^i, k) \right)
= \left( \prod_{i=0}^{k} q_i \right) D^{-k} \left( \prod_{i=k+1}^{n-1} p_i \right) B(x_{k+1}^n|x_1^k,k)
\geq \delta_k B(x_{k+1}^n|x_1^k,k), \quad (23)$$

where

$$\delta_k = \left( \prod_{i=0}^{k} q_i \right) D^{-k} \left( \prod_{i=k+1}^{\infty} p_i \right) > 0. \quad (24)$$

Hence the claim (ii) follows. □

Combining Theorem 1(ii) with the ergodic theorem we obtain the proof of universality.

Theorem 2 The switch distribution is strongly and weakly universal.

Proof: Let $Q$ be a stationary ergodic distribution. Since the alphabet of $X_1$ is finite, by the ergodic theorem differences $B(X_1^n|X_1^{n-1}, k) - Q(X_1^n|X_1^{n-1})$ converge to 0 $Q$-almost surely. Hence

$$\lim_{n \to \infty} \frac{1}{n} \left[ - \log B(X_1^n|X_1^k,k) \right] = \lim_{n \to \infty} \frac{1}{n} \left[ - \prod_{i=k+1}^{n} \log Q(X_i|X_{i-1}^{i-k-1}) \right]. \quad (25)$$

Applying the ergodic theorem again, we obtain

$$\lim_{n \to \infty} \frac{1}{n} \left[ - \prod_{i=k+1}^{n} \log Q(X_i|X_{i-1}^{i-k-1}) \right] = E_Q \left[ - \log Q(X_{k+1}|X_k^k) \right]. \quad (26)$$
Now we combine these facts with Theorem 3(ii), which yields
\[
\limsup_{n \to \infty} \frac{1}{n} \left[ -\log P(X^n_n) \right] \leq \inf_{k \in \mathbb{N}} \lim_{n \to \infty} \frac{1}{n} \left[ -\log B(X^n_{k+1} | X^n_k, k) \right] = \inf_{k \in \mathbb{N}} \mathbb{E}_Q \left[ -\log Q(X_{k+1} | X^n_{k+1}) \right] = h_Q. \tag{28}
\]
Hence the distribution \( P \) is strongly universal. Moreover, as shown in [17], the claim of Theorem 3(i) and the strong universality are sufficient conditions that distribution \( P \) be weakly universal. \( \square \)

A naive implementation of the switch distribution \( P(x^n_1) \) has the time complexity \( O(n^3) \) for the following reason: There are \( O(n^2) \) calls of \( B(x_{i+1} | x^n_1, k) \) where \( k, l \leq n \) and in a naive implementation each \( B(x_{i+1} | x^n_1, k) \) has time complexity \( O(kl) \). This is, however, a very careless approach and usually we can do much better. Let us denote the maximal length of a substring that appears at least twice in a string \( z^n_1 \) as
\[
L(z^n_1) := \max \left\{ k : \exists w^n_1 : c(w^n_1 | z^n_1) > 1 \right\}. \tag{29}
\]
For brevity, \( L(z^n_1) \) will be called the depth of \( z^n_1 \).

**Theorem 3** The value of the switch distribution \( P(x^n_1) \) can be computed in time \( O(ns) \) where \( s = L(x^n_1) \) is the depth of \( x^n_1 \).

**Remark:** The depth \( L(X^n_1) \) is bounded by \( O(\log n) \) for a large class of processes called finite-energy processes. They can be obtained by dithering ergodic processes with an IID noise [28]. For texts in natural language, an experiment indicates that the depth \( L(x^n_1) \) is of order \( O(\log^\alpha n) \), where \( \alpha < 4 \) [29].

**Proof:** We can knock down the complexity of an individual call of \( B(x_{i+1} | x^n_1, k) \) to a constant if we store the frequencies of substrings tested in formula (12) and we increment them on line. Some further important savings can be done if we know the depth \( s = L(x^n_1) \). The value of \( s \) can be computed in time \( O(n) \) by building the suffix tree of \( x^n_1 \) [30]. Once we have that \( s \), let us observe that
\[
B(x_{i+1} | x^n_1, k) = B(x_{i+1} | x^n_1, s) \tag{30}
\]
holds for all \( k > s \).

Thus we can flush all probabilities \( P(x^n_1, k) \) for \( k > s \) into a dummy variable \( P(x^n_1, \bullet) \). In the following, without affecting the value of \( P(x^n_1) \), the recursion (15)-(16) can be altered to
\[
P(x^n_0, k) = 0 \text{ for } k < -1 \text{ or } k \geq n, \tag{31}
\]
\[
P(x^n_0, \bullet) = 0 \text{ for } n < s + 1, \tag{32}
\]
\[
P(x^{n+1}_0, k) = \left[p_n P(x^n_0, k) + q_n P(x^n_0, k - 1) \right] B(x_{n+1} | x^n_0, k) \tag{33}
\]
for \( n \geq 1 \) and \( -1 \leq k \leq \min(n, s) \),
\[
P(x^{n+1}_0, \bullet) = \left[p_n P(x^n_1, \bullet) + q_n P(x^n_1, s) \right] B(x_{n+1} | x^n_0, s) \text{ for } n \geq s + 1. \tag{34}
\]
The formula for the total probability becomes
\[
P(x^n_1) = \sum_{k=-1}^{s} P(x^n_0, k) + P(x^n_0, \bullet). \tag{35}
\]
Hence the time complexity of \( P(x^n_1) \) is of order \( O(ns) \). \( \square \)
The space complexity of the switch distribution can also be reduced by observing that in order to compute \( B(x_{l+1}^l| x_l^1, k) \) we only need to store the frequencies of substrings \( w \) that appear in \( x_l^1 \) at least twice and the frequencies of their extensions \( wa \), where \( a \in \{0, 1, ..., D - 1\} \). These strings can be also found while building the suffix tree of \( x_l^1 \).

Parameter \( s \) in the algorithm (31)–(35) will be called the depth of the switch distribution. Without a significant change of \( P(x_l^1) \), the depth of the switch distribution can be chosen as much smaller than the depth of string \( x_l^1 \). This fact can be also used for the further speed-up of computation. Fixing the depth, however, leads asymptotically to the \( Q \)-almost sure bound

\[
\lim_{n \to \infty} \frac{1}{n} \left[ - \log P(X_n^1) \right] = \lim_{n \to \infty} \frac{1}{n} \left[ - \log B(X_{n+1}^n|X_1^s, s) \right]
\]

(36)

if \( Q \) is ergodic. Conditional entropy \( E_Q \left[ - \log Q(X_{s+1}^1|X_1^s) \right] \) is greater than \( h_Q \).

### III The preadapted switch distribution

Often we want to predict or compress data \( x_l^1 \) that are generated by a class of complex unknown distributions \( Q \) that partly resemble the empirical distribution of another, much larger data \( y_j^1 \). Such a case arises in particular in the compression of texts in natural language. Then using a universal distribution such as the plain switch distribution need not be the best approach, since this distribution has to learn all frequencies of substrings from the data \( x_l^1 \). A competing approach is to use frequencies of substrings from the larger data \( y_j^1 \). This can yield a better compression rate for finite data \( x_l^1 \). The problem of using a fixed empirical distribution of \( y_j^1 \) is, however, that it is not universal. The source of the problem lies in using non-adaptive substring frequencies. A simple solution for this problem is to initialize the substring frequencies with the frequencies coming from \( y_j^1 \) and let them gradually adapt to \( x_l^1 \). In this way we obtain a preadapted universal compression scheme. One can suppose that this scheme may compress better than both the plain switch distribution and the empirical distribution of \( y_j^1 \).

Let us clarify this idea.

**Definition 2 (fixed switch distribution)** Let \( y_j^1 \) be a fixed sequence, called the training data. Define conditional probabilities \( B(x_{n+1}^n| x_1^n, -1) = D^{-1} \) and

\[
B(x_{n+1}^n| x_1^n, k) = \frac{c(x_{n+1-1-k}^n|y_j^1) + B(x_{n+1}^n| x_1^n, k - 1)}{c(x_{n+1-1-k}^n|y_j^{n-k}) + 1}.
\]

(38)

Using these \( B(x_{n+1}^n| x_1^n, k) \), we define the fixed switch distribution \( P \) via formulae (13)–(17).

For short blocks \( x_l^1 \), the fixed switch distribution can achieve much lower compression rate than the plain switch distribution but is not universal. To obtain a universal distribution which combines the advantages of the fixed switch distribution and the plain switch distribution, we may consider a compromise between expressions (12) and (38). This can be done easily as follows.
Definition 3 (preadapted switch distribution) Let \( y_1 \) be a fixed sequence, called the training data. Define conditional probabilities

\[
B(x_{n+1} | x_1^n, k) = \frac{\mathbb{P}(x_{n+1} = y_1^n x_1^n, k \rightarrow 1)}{\mathbb{P}(x_{n+1} = y_1^n x_1^n, k \rightarrow 1) + 1}.
\]

(39)

Using these \( B(x_{n+1} | x_1^n, k) \), we define the preadapted switch distribution \( P \) via formulae (13)–(17).

As in the plain case, we can show that the preadapted switch distribution is universal and efficiently computable. The proof of universality relies on the observation that the influence of training data \( y_1 \) on the probability of long blocks \( x_1^n \) is asymptotically negligible.

Theorem 4 The preadapted switch distribution is strongly and weakly universal.

Proof: Analogously to the plain switch distribution, the preadapted switch distribution satisfies the analogue of Theorem 1. Having this fact in mind, we can prove the universality. Let \( Q \) be a stationary ergodic distribution. Since the alphabet of \( X_i \) is finite, by the ergodic theorem differences

\[
B(x_{n+1} | x_1^n, k) - Q(x_{n+1} | x_1^n, s)
\]

converge to 0 \( Q \)-almost surely. The further reasoning proceeds like the proof of Theorem 2. □

Theorem 5 The value of the preadapted switch distribution \( P(x_1^n) \) can be computed in time \( O((j + n)s) \) where \( s = L(y_1^n x_1^n) \).

Proof: The complexity of an individual call of \( B(x_{l+1} | x_1^n, k) \) can be reduced to a constant if we record the frequencies of substrings tested in formula (39) and we increment them on line. Initializing these frequencies takes time \( O(js) \). Let us also observe that

\[
B(x_{l+1} | x_1^n, k) = B(x_{l+1} | x_1^n, s)
\]

(40)

holds for all \( k > s \). Thus without affecting the value of \( P(x_1^n) \), the algorithm (15)–(16) can be changed to (31)–(34) and the formula for the total probability becomes (35). Thus the time complexity of \( P(x_1^n) \) is of order \( O((j + n)s) \). □

The space complexity of the preadapted switch distribution can also be reduced by noticing that in order to compute \( B(x_{l+1} | x_1^n, k) \) we only have to record the frequencies of substrings \( w \) that appear in \( y_1^n x_1^n \) at least twice and the frequencies of their extensions \( wa \), where \( a \in \{0, 1, ..., D - 1\} \).

IV Bounding Hilberg’s exponent \( \beta \)

Here we describe a simple experiment that we have performed using the three switch distributions and the Lempel-Ziv code. As the training data we have taken The Complete Memoirs by J. Casanova (6,719,801 characters), and as the compressed text—Gulliver’s Travels by J. Swift (579,438 characters). Both texts were downloaded from the Project Gutenberg. The alphabet size was set

\[\text{http://www.gutenberg.org/}\]
Table 1: Compression rates for the switch distributions and the LZ code.

| n    | LZ        | plain switch | preadapted switch | fixed switch |
|------|-----------|--------------|-------------------|-------------|
| 2    | 7.7459    | 8.3547       | 6.4605            | 7.2124      |
| 4    | 10.2089   | 8.6367       | 6.2363            | 6.2506      |
| 8    | 10.1347   | 8.4657       | 5.847             | 5.5963      |
| 16   | 10.1122   | 8.1227       | 5.8676            | 5.6687      |
| 32   | 9.9482    | 6.9604       | 5.1395            | 5.0712      |
| 64   | 9.8816    | 6.8478       | 4.9319            | 4.8489      |
| 128  | 9.4894    | 6.7355       | 5.254             | 5.8753      |
| 256  | 9.1817    | 5.9023       | 4.9481            | 5.2167      |
| 512  | 8.8427    | 5.2381       | 4.7506            | 4.8141      |
| 1024 | 8.5069    | 4.6831       | 4.414             | 4.626       |
| 2048 | 8.0525    | 4.1411       | 4.0127            | 4.4325      |
| 4096 | 7.7158    | 3.9809       | 3.8476            | 4.4953      |
| 8192 | 7.3084    | 3.6209       | 3.5361            | 4.4023      |
| 16384| 6.9471    | 3.3941       | 3.3238            | 4.3935      |
| 32768| 6.5467    | 3.0459       | 3.0114            | 4.3422      |
| 65536| 6.1909    | 2.7745       | 2.7504            | 4.327       |
| 131072| 5.865   | 2.5342       | 2.5223            | 4.3188      |
| 262144| 5.5665   | 2.3759       | 2.3664            | 4.3142      |
| 524288| 5.2928   | 2.2252       | 2.213             | 4.3126      |

Table 1: Compression rates for the switch distributions and the LZ code.

as \( D = 256 \). The switch distributions were computed using transition probabilities \( p_n \) of form (18) with \( \alpha = 1.001 \) since we observed that the lower the \( \alpha \) is the better compression is achieved. Moreover, we have used algorithm (31)–(35) with fixed depth \( s = 7 \) since more than 99.99% of the probability mass in the observed cases concentrated in \( P(x_n^1, k) \) with \( k \leq 4 \). Hence it was a safe approximation. The Lempel-Ziv code was computed by our own implementation for the ASCII encoding of the text. The results are presented in Tables 1 and 2 and Figures 2 and 3.

In Figure 2 and Table 1, the quality of compression can be compared for the particular distributions. Among the universal schemes, the best compression is given by the preadapted switch distribution followed by the plain switch distribution followed by the LZ code. However, our hope that the preadapted switch distribution will significantly beat the plain switch distribution has not been fully confirmed. Indeed for short blocks the preadapted switch distribution mimics the behavior of the fixed switch distribution and performs much better than the plain switch distribution. Alas, for long blocks the difference between the two universal switch distributions becomes negligible. Ultimately, both universal switch distributions compress the text twice better than the LZ code. For no universal code we can observe the ultimate stabilization of the compression rate. On the other hand, the fixed switch distribution, which is not universal, stabilizes ultimately at the constant rate of 4.31 bpc.

The stabilization of the fixed switch distribution is clearly visible in Figure 3 and Table 2 which concern the pointwise mutual information. Namely, we can see that pointwise mutual information for the fixed switch distribution does not grow, whereas for the other distributions, which are universal, the pointwise mu-
Figure 2: Compression rates for the switch distributions and the LZ code. The solid line is the least square regression $y = 11.51n^{-0.127}$, computed for the plain switch distribution. The dotted line is the least square regression $y = 12.66n^{-0.0625}$, computed for the LZ code.

| $n$ | LZ    | Plain switch | Preadapted switch | Fixed switch |
|-----|-------|--------------|-------------------|--------------|
| 2   | -1.32 | -0.71        | -0.64             | -2.14        |
| 4   | -6.04 | -1.13        | -0.67             | 0.52         |
| 8   | -3.99 | 1.37         | -0.5              | 0.62         |
| 16  | -4.96 | 5.3          | 1.27              | 0.59         |
| 32  | 5.25  | 37.5         | 4.96              | 0.65         |
| 64  | 4.26  | 38.26        | 1.8               | 0.62         |
| 128 | 27.27 | 49.6         | 28.22             | -2.47        |
| 256 | 61.92 | 100.78       | -3.89             | 0.57         |
| 512 | 155.1 | 166.76       | -29.5             | 0.64         |
| 1024| 353.89| 345.54       | 156.62            | 0.64         |
| 2048| 789.56| 668.01       | 441.52            | 0.63         |
| 4096| 1554.31| 954.54     | 786.68            | 0.65         |
| 8192| 3187.28| 2128.53   | 1945.46           | 0.65         |
| 16384| 6119.95| 4017.21    | 3558.77           | 0.64         |
| 32768| 11608.28| 7062.68    | 6551.43           | 0.63         |
| 65536| 22241.83| 13877.79   | 13549.91          | 0.63         |
| 131072| 41621.75| 26852.4   | 25727.25          | 0.64         |
| 262144| 78530.25| 47113.91  | 45313.69          | 0.59         |
| 524288| 142330.87| 81859.85 | 81873.95          | 0.6          |

Table 2: Pointwise mutual information for the switch distributions and the LZ code.
Figure 3: Pointwise mutual information for the switch distributions and the LZ code. The solid line is the least square regression $y = 1.395n^{0.834}$, computed for the plain switch distribution. The dashed line is the least square regression $y = 0.946n^{0.863}$, computed for the preadapted switch distribution. The dotted line is the least square regression $y = 1.209n^{0.887}$, computed for the LZ code.
tual information grows rather fast. The tightest bound for the pointwise mutual information is obtained in the case of the plain switch distribution, which gives the exponent $\gamma = 0.83$ for the codewise Hilberg conjecture \cite{7}. It is surprising that the pointwise mutual information for the two other universal distributions grows almost at the same rate, despite the large difference of compression rates between the universal switch distributions and the LZ code. This indicates that the codewise Hilberg conjecture \cite{7} may be more than an accidental property of a universal code.

The described experiment has a preliminary character and its main motivation was to illustrate the advantages of universal switch distributions. In the future research, which belongs to the field of quantitative linguistics rather than information theory, we are planning to test the codewise Hilberg conjecture \cite{7} for a wide selection of texts and languages. This task requires universal codes with low compression rates. Apart from the switch distributions introduced here, one should seek also for other codes having these properties. We may hope there is some room for improvement.

Acknowledgment

The author wishes to thank Jacek Koronacki and Jan Mielniczuk for a discussion.

References

[1] W. Hilberg, “Der bekannte Grenzwert der redundanzfreien Information in Texten — eine Fehlinterpretation der Shannonschen Experimente?” Frequenz, vol. 44, pp. 243–248, 1990.

[2] W. Ebeling and G. Nicolis, “Entropy of symbolic sequences: the role of correlations,” Europhys. Lett., vol. 14, pp. 191–196, 1991.

[3] ———, “Word frequency and entropy of symbolic sequences: a dynamical perspective,” Chaos Sol. Fract., vol. 2, pp. 635–650, 1992.

[4] W. Ebeling and T. Pöschel, “Entropy and long-range correlations in literary English,” Europhys. Lett., vol. 26, pp. 241–246, 1994.

[5] W. Bialek, I. Nemenman, and N. Tishby, “Predictability, complexity and learning,” Neural Comput., vol. 13, p. 2409, 2001.

[6] ———, “Complexity through nonextensivity,” Physica A, vol. 302, pp. 89–99, 2001.

[7] J. P. Crutchfield and D. P. Feldman, “Regularities unseen, randomness observed: The entropy convergence hierarchy,” Chaos, vol. 15, pp. 25–54, 2003.

[8] Ł. Dębowski, “On the vocabulary of grammar-based codes and the logical consistency of texts,” IEEE Trans. Inform. Theor., vol. 57, pp. 4589–4599, 2011.
[9] ——, “Mixing, ergodic, and nonergodic processes with rapidly growing information between blocks,” *IEEE Trans. Inform. Theor.*, vol. 58, pp. 3392–3401, 2012.

[10] ——, “On Hilberg’s law and its links with Guiraud’s law,” *J. Quantit. Linguist.*, vol. 13, pp. 81–109, 2006.

[11] G. Herdan, *Quantitative Linguistics*. Butterworths, 1964.

[12] H. S. Heaps, *Information Retrieval—Computational and Theoretical Aspects*. Academic Press, 1978.

[13] Ł. Dębowski, “Empirical evidence for Hilberg’s conjecture in single-author texts,” in *Methods and Applications of Quantitative Linguistics—Selected papers of the 8th International Conference on Quantitative Linguistics (QUALICO)*, I. Obradović, E. Kelih, and R. Köhler, Eds., 2013, pp. 143–151.

[14] C. Shannon, “Prediction and entropy of printed English,” *Bell Syst. Tech. J.*, vol. 30, pp. 50–64, 1951.

[15] R. M. Gray and L. D. Davisson, “The ergodic decomposition of stationary discrete random processes,” *IEEE Trans. Inform. Theor.*, vol. 20, pp. 625–636, 1974.

[16] O. Kallenberg, *Foundations of Modern Probability*. Springer, 1997.

[17] T. Weissman, “Not all universal source codes are pointwise universal,” 2004, [http://www.stanford.edu/~tsachy/interest.htm](http://www.stanford.edu/~tsachy/interest.htm).

[18] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *IEEE Trans. Inform. Theor.*, vol. 23, pp. 337–343, 1977.

[19] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, J. C. Lai, and R. L. Mercer, “An estimate of an upper bound for the entropy of english,” *Comput. Linguist.*, vol. 18, pp. 31–40, 1992.

[20] F. Jelinek, *Statistical Methods for Speech Recognition*. The MIT Press, 1997.

[21] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[22] J. G. Cleary and I. H. Witten, “Data compression using adaptive coding and partial string matching,” *IEEE Trans. Comm.*, vol. 32, pp. 396–402, 1984.

[23] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “The context tree weighting method: Basic properties,” *IEEE Trans. Inform. Theor.*, vol. 41, pp. 653–664, 1995.

[24] D. Ron, Y. Singer, and N. Tishby, “The power of amnesia: Learning probabilistic automata with variable memory length,” *Machine Learn.*, vol. 25, pp. 117–149, 1996.
[25] J. C. Kieffer and E. Yang, “Grammar-based codes: A new class of universal lossless source codes,” IEEE Trans. Inform. Theor., vol. 46, pp. 737–754, 2000.

[26] D. S. an Giovanni Motta, Handbook of Data Compression. Springer, 2009.

[27] T. van Erven, P. Grünwald, and S. de Rooij, “Catching up faster in Bayesian model selection and model averaging,” in Advances in Neural Information Processing Systems 20 (NIPS 2007), 2007.

[28] P. C. Shields, “String matching bounds via coding,” Ann. Probab., vol. 25, pp. 329–336, 1997.

[29] Ł. Dębowski, “Maximal lengths of repeat in English prose,” in Synergetic Linguistics. Text and Language as Dynamic System, S. Naumann, P. Grzybek, R. Vulanović, and G. Altmann, Eds. Wien: Praesens Verlag, 2012, pp. 23–30.

[30] E. Ukkonen, “On-line construction of suffix trees,” Algorithmica, vol. 14, pp. 249–260, 1995.