

# Data Discovery and Anomaly Detection Using Atypicality: Theory

Anders Høst-Madsen, *Fellow, IEEE*, Elyas Sabeti, *Member, IEEE*, Chad Walton

## Abstract

A central question in the era of 'big data' is what to do with the enormous amount of information. One possibility is to characterize it through statistics, e.g., averages, or classify it using machine learning, in order to understand the general structure of the overall data. The perspective in this paper is the opposite, namely that most of the value in the information in some applications is in the parts that deviate from the average, that are unusual, atypical. We define what we mean by 'atypical' in an axiomatic way as data that can be encoded with fewer bits in itself rather than using the code for the typical data. We show that this definition has good theoretical properties. We then develop an implementation based on universal source coding, and apply this to a number of real world data sets.

## Index Terms

Big Data, atypicality, minimum description length, data discovery, anomaly.

## I. INTRODUCTION

One characteristic of the information age is the exponential growth of information, and the ready availability of this information through networks, including the internet – “Big Data.” The question is what to do with this enormous amount of information. One possibility is to characterize it through statistics – think averages. The perspective in this paper is the opposite, namely that most of the value in the information is in the parts that deviate from the average, that are unusual, atypical. The rest is just background noise. Take art: the truly valuable paintings are those that are rare and atypical. The same could be true for scientific research and entrepreneurship. Take online collections of photos, such as Flickr.com. Most of the photos are rather pedestrian snapshots and not of interest to a wider audience. The photos that of interest are those that are unique. Flickr has a collection of photos rated for 'interestingness,' and one can notice that those photos are indeed very different from typical photos. They are atypical.

The aim of our approach is to extract such 'rare interesting' data out of big data sets. The central question is what 'interesting' means. A first thought is to focus on the 'rare' part. That is, interesting data is something that

A. Høst-Madsen and E. Sabeti are with the Department of Electrical Engineering, University of Hawaii Manoa, Honolulu, HI 96822 (e-mail: {ahm,sabeti}@hawaii.edu). C. Walton is with the Department of Surgery, University of Hawaii, Honolulu, HI, 96813. Email: cwalton@hawaii.edu. This work was supported in part by NSF grants CCF 1017823, 1017775, and 1434600. The paper was presented in part at IEEE Information Theory Workshop 2013, Seville.

is unlikely based on prior knowledge of typical data or examples of typical data, i.e., training. This is the way an outlier is usually defined. Unlikeliness could be measured in terms of likelihood, in terms of codelength [1], [2] – called ‘surprise’ in [3] – or according to some distance measure. This is also the most common principle in anomaly detection [4]. However, perhaps being unlikely is not sufficient for something to be ‘interesting.’ In many cases, outliers are junk that are eliminated not to contaminate the typical data. What makes something interesting is maybe that it has a new unusual structure in itself that is quite different from the structure of the data we have already seen. Return to the example of paintings: what make masterworks interesting is not just that they are different than other paintings, but that they have some ‘structure’ that is intriguing. Or take another example. Many scientific discoveries, like the theory of relativity and quantum mechanics, began with experiments that did not fit with prevailing theories. The experiments were outliers or anomalies. What made them truly interesting was that it was possible to find a new theory to explain the data, be it relativity or quantum mechanics. This is the principle we pursue: finding data that have better alternative explanations than those that fit the typical data.

Something being unlikely is not even necessary for the data to be ‘interesting.’ Suppose the typical data is iid uniform  $\{0,1\}$ . Then any sequence of bits are equally likely. Therefore, a sequence consisting of purely 1, 111111... is in no way ‘surprising.’ Yet, it should catch our interest.

When we look for new interesting data, a characteristic is that we do not know what we are looking for. We are looking for “unknown unknowns” [5]. Instead of looking at specific statistics of data, we need to use a universal approach. This is provided by information theory.

This idea of finding alternative explanations for data rather than measuring some kind of difference from typical data is what separates our method from usual approaches in outlier detection and anomaly detection. As far as we can determine from reading hundreds of papers, our approach has not been explored previously. Obviously, information theory and coding have been used in anomaly detection, data mining, and knowledge discovery before, and we will discuss how this compares to our approach later. Our methodology also has connections to tests for randomness, e.g., the run length test and [6], [7], but our aim is different.

### *A. Applications*

Atypicality is relevant in large number of various applications. We will list a few applications here.

**ECG.** For electrocardiogram (ECG) recordings there are patterns in heart rate variability that are known to indicate possible heart disease [8], [9], [10], [11]. With modern technology it is possible for an individual to wear and unobtrusive heart rate monitor 24/7. If atypical patterns occur, it could be indicative of disease, and the individual or a doctor could be notified. But perhaps a more important application is to medical research. One can analyze a large collection of ECG recordings and look for individuals with atypical patterns. This can then potentially be used to develop new diagnostic tools.

**Genomics.** Another example of application is interpretation of large collections of genomics data. Given that all mammals have essentially the same set of genes, there must exist some significant differences that distinguish the obvious distinct attributes between species, as well as more subtle differences within a species. Although the

genome has been mined by exhaustive studies applying a panoply of approaches, regions once thought to be “uninteresting” have recently come under increased study for their potential role in defined morphological and physiological differences between individuals [12]. Applying an atypical evaluation tool to genomic data from individuals of known pathophysiological/morphological irregularities may provide valuable insight to the genetic mechanisms underlying the condition.

**Ocean Monitoring.** In passive acoustic monitoring (PAM) [13] of oceans, one or more hydrophones is towed behind a ship or deployed in a fixed bottom-mounted or suspended array in order to record vocalizations of marine mammals. One major focus is to detect, and perhaps count, rare or endangered species. It would be highly interesting to scan the data for any unusual patterns, which can then be further examined by a researcher.

**Plant Monitoring.** In for example nuclear plants, atypical monitoring data may be indicative of something about to go wrong.

**Computer Networks.** Atypical network traffic could be indicative of a cyberattack. This is already being used through anomaly detection [14]. However, an abstract atypicality approach can be used to find more subtle attacks – the unknown unknowns.

**Airport Security.** Already software is being used to flag suspicious flyers, likely based on past attacks. Atypical detection could be used to find innovative attackers.

**Stock Market.** Atypicality could be used to detect insider trading. It could also be used by investors to find unusual stocks to invest in, promising outstanding returns – or ruin.

**Astronomy.** Atypicality can be used to scan huge databases for new kinds of cosmological phenomena.

**Credit Card Fraud.** Unusual spending patterns could be indicative of fraud. This is already used by credit card companies, but obviously in a simple, and annoying way, as anyone who’s credit card has been blocked on an overseas trip can testify to.

**Gambling.** Casinos are constantly fighting fraudsters. This is a game of cat and mouse. Fraudsters constantly find new ways to trick the casinos (one such inventor was Shannon himself). Therefore, an abstract atypicality approach may be the best solution to catch new ways of fraud.

## B. Notation

We use  $x$  to denote a sequence in general, and  $x^l$  when we need to make the length explicit;  $x_i$  denotes a single sample of the sequence. We use capital letters  $X_i$  to denote random variables rather than specific outcomes. Finally  $\mathcal{X}$  denotes a subsequence. All logarithms are to base 2 unless otherwise indicated.

## II. ATYPICALITY

Our starting point is the in theory of randomness developed by Kolmogorov and Martin-Löf [15], [7], [16]. Kolmogorov divides (infinite) sequences into ‘typical’ and ‘special.’ The typical sequences are those that we can call random, that is, they satisfy all laws of probability. They can be characterized through Kolmogorov complexity. A sequence of bits  $\{x_n, n = 1, \dots, \infty\}$  is random (i.e. iid uniform) if the Kolmogorov complexity of

the sequence satisfies  $K(x_1, \dots, x_n) \geq n - c$  for some constant  $c$  and for all  $n$  [15]. The sequence is incompressible if  $K(x_1, \dots, x_n | n) \geq n$  for all  $n$ , and a finite sequence is algorithmically random if  $K(x_1, \dots, x_n | n) \geq n$  [16]. In terms of coding, an iid random sequence is also incompressible, or, put another way, the best coder is the identity function. Let us assume we draw sequences  $x^n$  from an iid uniform distribution. The optimum coder is the identity function, and the code length is  $n$ . Now suppose that for one of these sequences we can find a (universal) coder so that the code length is less than  $n$ ; while not directly equivalent, one could state this as  $K(x_1, \dots, x_n | n) < n$ . With an interpretation of Kolmogorov's terms, this would not be a 'typical' sequence, but a 'special' sequence. We will instead call such sequences 'atypical.' Considering general distributions and general (finite) alphabets instead of iid uniform distributions, we can state this in the following general principle

**Definition 1.** *A sequence is atypical if it can be described (coded) with fewer bits in itself rather than using the (optimum) code for typical sequences.*

This definition is central to our approach to the atypicality problem.

In the definition, the “(optimum) code for typical sequences,” is quite specific, following the principles in for example [16]. We assume prefix free codes. Within that class the coding could be done using Huffman codes, Shannon codes, Shannon-Fano-Elias codes, arithmetic coding etc. We care only about the code length, and among these the variation in length is within a few bits, so that the code length for typical encoding can be quite accurately calculated.

On the other hand, “described (coded) with fewer bits in itself” is less precise. In principle one could use Kolmogorov complexity, but Kolmogorov complexity is not calculable and it is only given except for a constant, and comparison with code length therefore is not an “apples-to-apples” comparison. Rather, some type of universal source coder should be used. This can be given a quite precise meaning in the class of finite state machine sources, [17] and following work, and is strongly related to minimum description length (MDL) [18], [19], [20], [17]. What is essential is that we adhere to *strict decodability* at the decoder. The decoder only sees a stream of bits, and from this it should be able to accurately reconstruct the source sequence. So, for example, if a sequence is atypical, there must be a type of “header” telling the decoder to use a universal decoder rather than the typical decoder. Or, if atypical sequences can be encoded in multiple ways, the decoder must be informed through the sequence of bits which encoder was used. One could argue that such things are irrelevant for for example anomaly detection, since we are not actually encoding sequences. The problem is that if such terms are omitted, it is far too easy to encode a sequence “in itself.” This is like choosing a more complex model to fit data, without accounting for the model complexity in itself, which is exactly what MDL sets out to solve, although also in this case actual encoding is not done. We therefore try to account for all factors needed to describe data, and we believe this is one of the key strengths of the approach.

A major difference between atypical data and anomalous data is that atypicality is an axiomatic property of data, defined by Definition 1 based on Kolmogorov-Martin-Löf randomness. On the other hand, as far as we know, an

anomaly is not something that can be strictly defined. Usually, we think of an anomaly as something caused by an outside phenomenon: an intruder in a computer network, a heart failure, a gambler playing tricks. This influences how we think of performance. If a detector fails to give an indication of an anomaly, we have a miss (or type II error), but if it gives an indication when such things are not happening we have a false alarm (type I error). Atypicality, on the other hand, is purely a property of data. Ideally, there are therefore no misses or false alarms: data is atypical or not. Here is what we mean. If there is an anomaly that expresses itself through the observed data, that must mean that there is some structure in the data, and in theory a source coder would discover and exploit such structure and reduce code length. Thus, if the data is not atypical that means there is simply no way to detect the anomaly through the observations – again in theory. We therefore cannot really call that a miss. On the other hand, suppose that in a casino a gambler has a long sequence of wins. This could be due to fraud, but it could also be simply due to randomness. But casino security would be interested in either case for further scrutiny. Thus, the reason for the atypicality does not really matter, the atypicality itself matters. Still, to distinguish the two cases we call a sequence *intrinsically* atypical if it is atypical according to Definition 1 while being generated from the typical probability model, while it is *extrinsically* atypical if it is in fact generated by any other probability law.

Definition 1 has two parts that work in concert, and we can write it simplified as  $C_t(x) - C_a(x) > 0$  where  $C_t$  is the typical code length and  $C_a$  the atypical code length. The typical code length  $C_t(x)$  is simply an expression of the likelihood of seeing a particular sequence. If  $C_t(x)$  is large it means that the given sequence is unlikely to happen, and detecting sequences by  $C_t(x) > \tau$  would catch many outliers. As an extreme example, if a sequence is impossible according to the typical distribution,  $C_t(x) = \infty$ , and it would always be caught. But it would not work universally. If, as we started out with, typical sequences are iid uniform, any sequence is equally likely and  $C_t(x) > \tau$  would not catch any sequences. In this case, if a test sequence has some structure, it is possible that  $C_a(x) < C_t(x)$ , and such sequences would be caught by atypicality; thus calculating  $C_a(x)$  is essential. Calculating  $C_t(x)$  is also essential. Suppose that we instead use  $E[C_t] - C_a(x)$ , where  $E[C_t]$  is the code length used to encode typical sequences “on average,” essentially the entropy rate. Again, this will catch some sequences: if a test sequence has more or less structure than typical sequences,  $E[C_t] - C_a(x) \neq 0$ . But again, it will omit very obvious examples: if as test sequence we use a typical sequence with 0 and 1 swapped,  $E[C_t] \approx C_a(x)$ , while on the other hand  $C_t(x) > C_a(x)$ . And impossible sequences with  $C_t(x) = \infty$  would not be caught with absolute certainty. Now, to declare something an outlier, we have to find a coder with  $C_a(x) < C_t(x)$ . It is not sufficient that  $C_t(x)$  is large, i.e., that the sequence is unlikely to happen. However, we can always use the trivial coder that transmits data uncoded. If the sequence is unlikely to happen according to the typical distribution, then it is likely that  $C_t(x) > (\text{length of } x)$ .

Thus, it can be seen that the two parts work in concert to catch sequences. Each part might catch some sequences, but to catch all “anomalies,” both parts have to be used.

Another point of view is the following. Suppose again the typical model is binary uniform iid. We look at a collection of sequences, and now we want to find the *most* atypical sequences, i.e., the most “interesting” sequences. Without a specification of what “interesting” is, it seems reasonable to choose those sequences that have the most

structure, and again this can reasonably be measured by how much the sequence can be compressed. This is what Rissanen [17] calls “useful information,”  $U(x) = n - C_a(x)$ . But again, we need to take into account the typical model if it is not uniform iid. For example, if typical sequences have much structure, then sequence with little structure might be more interesting. We therefore end up with that  $C_t(x) - C_a(x)$  is a reasonable measure of how interesting sequences might be.

#### A. Alternative approaches

While, as argued in the introduction, and outlined above, what we are aiming for is not anomaly detection in the traditional sense, there are still many similarities. And certainly information theory and universal source coding has been used previously in anomaly detection, e.g., [4], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. The approaches have mostly been heuristic. A more fundamental and systematic approach is Information Distance defined in [31]. Without being able to claim that this applies to *all* of the perhaps hundreds of papers, we think the various approaches can be summarized as using universal source coding as a type of distance measure, whether it satisfies strict mathematical metric properties as in [31] or is more heuristic. On the other hand, our methodology in Definition 1 cannot be classified as a distance measure in a traditional sense. We are instead trying to find alternative explanations for data. We will comment on how our approach contrasts with a few other approaches.

While the similarity distance developed in [31] is not directly applicable to the problem we consider, we can to some extent adapt it, which is useful for contrast. The similarity distance is

$$d = \frac{\min\{K(y|x^*), K(x|y^*)\}}{\max\{K(x), K(y)\}}$$

Instead of being given the typical distribution, we can imagine that we are given a very long typical sequence  $x$  which is used for “training.” In that case

$$d = \frac{K(x|y^*)}{K(x)} = \frac{K(x, y) - K(y)}{K(x)}$$

within a certain approximation. Suppose, as was our starting point above, that the typical distribution is binary iid uniform. If  $y$  is also binary iid uniform, within a constant  $K(x, y) = K(x) + K(y)$ , and  $d = 1$ . But if  $y$  is drawn from some other distribution,  $x$  cannot help describing  $x$  either, and still  $d = 1$ . That makes sense: two completely random sequences are not similar, whether they are from the same distribution or not. Thus, similarity distance cannot be used for ‘anomaly’ detection as we have have defined it: looking for ‘special’ sequences in the words of Kolmogorov. This is not a problem of the similarity metric; it does exactly what it is designed for, which is really deterministic similarity between sequences, appropriate for classification. The reason similarity distance still gives results for anomaly detection [32] is actually that universal source coders approximate Kolmogorov complexity poorly.

Heuristic methods using for anomaly detection using universal source coding [4], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [1], [2] are mostly based on comparing code length. Let  $C(x)$  be the code length to encode the sequence  $x$  with a universal source coder. Let  $x$  be a training string and  $y$  a test sequence. We can

then compare  $\frac{C(x)}{|x|}$  with  $\frac{C(y)}{|y|}$  (which could be seen as a measure of entropy rate) or compare  $C(xy)$  with  $C(x)$  to detect change. The issue with this is that there are many completely dissimilar sources that have the same entropy rate. As an example, let the data be binary iid with the original source having  $P(X = 1) = \frac{1}{3}$  and the new source  $P(X = 1) = \frac{2}{3}$ . Then the optimum code for the original source and the optimum code for the new source have the same length. On the other hand, atypicality will immediately distinguish such sequences.

### III. BINARY IID CASE

In order to clarify ideas, at first we consider a very simple model. The typical model is iid binary with  $P(X_n = 1) = p$ . The alternative model class also binary iid but with  $P(X_n = 1) = \theta$ , where  $\theta$  is unknown. We want to decide if a given sequence  $x^l$  is typical or atypical. This can be stated as the hypothesis test problem

$$H_0 : \theta = p$$

$$H_1 : \theta \neq p$$

This problem does not have an UMP (universal most powerful) test. However, a common approach to solving this type of problem is the GLRT (generalized likelihood ratio test) [33]. Let

$$P(b) = P(X_n = b)$$

$$\hat{P}(b) = \frac{N(b|x^l)}{l}$$

where  $l$  is the sequence length and  $N(b|x^l)$  is the number of  $x_n = b \in \{0, 1\}$ . The GLRT is

$$\begin{aligned} L &= \log \frac{\prod_{b=0}^1 \hat{P}(b)^{N(b|x^l)}}{\prod_{b=0}^1 P(b)^{N(b|x^l)}} \\ &= \sum_{b=0}^1 N(b|x^l) \log \frac{1}{l} N(b|x^l) - \sum_{b=0}^1 N(b|x^l) \log P(b) \\ &= l \sum_{b=0}^1 \hat{P}(b) \log \frac{1}{l} N(b|x^l) - l \sum_{b=0}^1 \hat{P}(b) \log P(b) \\ &= lD(\hat{p}||p) \end{aligned}$$

$$\phi(x^l) = \begin{cases} 1 & L > t \\ 0 & L \leq t \end{cases} \quad (1)$$

Where  $D(\hat{p}||p) = \sum_{b=0}^1 \hat{P}(b) \log \frac{\hat{P}(b)}{P(b)}$  is the relative entropy [16] and  $t$  some threshold. While the GLRT is a heuristic principle, it satisfies some optimality properties, and in this case it is equal to the invariant UMP test [34], which can be considered an optimum solution under certain constraints. Thus, it is reasonable to take this as the optimum solution for this problem, and we do not need to appeal to Kolmogorov or information theory to solve the problem.

The complications start if we consider sequences of variable length  $l$ . The test (1) depends on the sequence length. We need to choose a threshold  $t(l)$  as a function of  $l$ , which will then result in a false alarm probability  $P_{FA}(t(l))$

and detection probability  $P_D(t(l))$ . There is no obvious argument for how to choose  $t(l)$  from a hypothesis testing point of view; we could choose  $t$  independent of  $l$ , but that is just another arbitrary choice.

We will consider this problem in the context of Definition 1. In order to do so, we need to model the problem from a coding point of view. We assume we have an (infinite) sequence of sequences of variable length  $l_i$ , and these need to be encoded. We need to encode each bit, and also to encode whenever a new sequence starts. For typical encoding of the bits we can use a Shannon code, Huffman code, arithmetic coding etc. The code length for a sequence of length  $l$  is

$$\begin{aligned} L_t &= N(1|x^l) \log \frac{1}{p} + N(0|x^l) \log \frac{1}{1-p} \\ &= l \left( \hat{p} \log \frac{1}{p} + (1 - \hat{p}) \log \frac{1}{1-p} \right) \end{aligned} \quad (2)$$

except for a small constant factor; here  $\hat{p} = \hat{P}(1) = \frac{1}{l} \sum x_i$ . We also need to encode where a sequence ends and a new one starts. For simplicity let us for now assume lengths are geometrically distributed. We can then model the problem as one with three source symbols '0', '1' and ',' with an iid distribution with  $P(',') = \epsilon$ ,  $P('0') = p - \frac{\epsilon}{2}$ ,  $P('1') = (1 - p) - \frac{\epsilon}{2}$ . If we assume  $\epsilon$  is small, the expression (2) is still valid for the content part, and to each sequence is added a constant  $-\log \epsilon$  to encode separators. To decide if a sequence is atypical according to Definition 1, we can use the universal source coder from [16]: the source encodes first the number of ones  $k$ ; then it enumerates the sequences with  $k$  ones, and transmits the index of the given sequence. For analysis it is important to have a simple expression for the code length. We can therefore use  $L_a = lH(\hat{p}) + \frac{1}{2} \log l$ . This is an approximation which is good for reasonably large  $l$  and it also reaches the lower bound in [17], [35]. The source-coder also needs to inform the decoder that the following is an atypical sequence (so that it knows to use the atypical decoder rather than the typical encoder), and where it ends. For the former we can use a ',' to indicate the start of an atypical sequence rather than the ',' for typical sequences. If the probability that a sequence is atypical is  $\delta \ll 1$ ,  $P(',') = \delta\epsilon$  and  $P(',') = (1 - \delta)\epsilon \approx \epsilon$ . The code length for a ',' now is  $-\log \epsilon - \log \delta$ . To mark the end of the atypical sequence we could again insert a ',' or a ',,'. But the code for either is based on the distribution of lengths of *typical* sequences, which we assume known, whereas we would have no knowledge of the length of atypical sequences. Instead it seems more reasonable to encode the length of the specific atypical sequence. As argued in [18], [36] this can be done with  $\log^* l + \log c$ , where  $c$  is a constant and

$$\log^*(l) = \log l + \log \log l + \log \log \log l + \dots \quad (3)$$

where the sum continues as long as the argument to the log is positive. To summarize we have

$$\begin{aligned} L_t &= l \left( \hat{p} \log \frac{1}{p} + (1 - \hat{p}) \log \frac{1}{1-p} \right) - \log \epsilon \\ L_a &= lH(\hat{p}) + \frac{1}{2} \log l + \log^* l + \log c - \log \epsilon - \log \delta \\ &\approx lH(\hat{p}) + \frac{3}{2} \log l - \log \epsilon + \tau \\ \tau &= -\log \delta + \log c \end{aligned} \quad (4)$$



The criterion for a sequence to be atypical is  $L_a < L_t$ , which easily seen to be equivalent to

$$D(\hat{p}||p) > \frac{\tau + \frac{3}{2} \log l}{l} \quad (5)$$

If the lengths are fixed, this reduces to (1). But if the lengths are variable, (5) provides a threshold as a function of  $l$ . The term  $\frac{3}{2} \log l$  ensures that  $\lim_{l \rightarrow \infty} P_{FA}(l) = 0$ , which seems reasonable. If instead  $D(\hat{p}||p) > \frac{\tau}{l}$  is used, it is easy to see that  $\lim_{l \rightarrow \infty} P_{FA}(l) > 0$ . Except for this property, the term  $\frac{3}{2} \log l$  might seem arbitrary, e.g., why  $\frac{3}{2}$ ? But it is based on solid theory, and as will be seen later it has several important theoretical properties.

We will examine the criterion (5) in more detail. The inequality (5) gives two thresholds for  $\hat{p}$ ,

$$\hat{p} > p_+$$

$$\hat{p} < p_-$$

Where  $0 < p_- < p < p_+ < 1$ . It is impossible to find explicit expressions for  $p_{\pm}$ , but it is clear that

$$p_{\pm} \rightarrow p \quad \text{as } l \rightarrow \infty.$$

Therefore, for  $l$  large, we can replace  $D(\hat{p}||p)$  with a series expansion. We then end up with the more explicit criterion

$$\begin{aligned} \frac{(p - \hat{p})^2}{pq \ln 4} &> \frac{1}{l} \left( \tau + \frac{3}{2} \log l \right) \\ |\hat{p} - p| &> \Delta\tau \doteq \sqrt{\frac{pq \ln 4}{l}} \sqrt{\tau + \frac{3}{2} \log l} \end{aligned} \quad (6)$$

In the following we will use this as it is considerably simpler to analyze. We can also write this as

$$\left| \frac{\sum_{i=1}^l x_i - p}{\sqrt{pql}} \right| > \sqrt{2\tau \ln 2 + 3 \ln l} \quad (7)$$

Now, if not for the term  $3 \ln l$ , this would be a central limit type of statement, and the probability that a sequence is classified as (intrinsically) atypical would be

$$P_A \approx 2Q\left(\sqrt{2 \ln 2 \tau}\right) \quad (8)$$

independent of  $l$ . Our main interest is exactly the the dependency on  $l$ , which is given by the following Theorem

**Theorem 2.** Consider an iid  $\{0, 1\}$ -sequence. Let  $P_A(l)$  be the probability that a sequence of length  $l$  is classified as intrinsically atypical according to (6). Then  $P_A(l)$  is bounded by

$$P_A(l) \leq 2^{-\tau+1} \frac{1}{l^{3/2}} K(l, \tau) \quad (9)$$

$$\forall \tau : \lim_{l \rightarrow \infty} K(l, \tau) = 1$$

For  $p = \frac{1}{2}$  this can be strengthened to

$$P_A(l) \leq 2^{-\tau+1} \frac{1}{l^{3/2}} \quad (10)$$

These bounds are tight in the sense that

$$\lim_{l \rightarrow \infty} \frac{\ln P_A(l)}{-\frac{3}{2} \ln l} = 1 \quad (11)$$

*Proof:* The Chernoff bound (e.g., [37]) states

$$\begin{aligned} P_A(l) &= P(|\hat{p} - p| > \Delta\tau) = 2P\left(\sum_{i=1}^l X_i \geq lp + b\right) \\ &\leq 2 \inf_{s>0} \{e^{-lsp-sb} M_X(s)^l\} \end{aligned}$$

Where (as usual,  $q = 1 - p$ )

$$b = \sqrt{lpq \ln 4} \sqrt{\tau + \frac{3}{2} \log l}$$

and  $M_S(s)$  is the moment generating function of  $X_i$ , which for a Bernoulli random variable is

$$M_X(s) = pe^s + q$$

Then

$$\frac{1}{2} P_A(l) \leq \inf_{s>0} \left\{ \exp(-s(pl + b)) (pe^s + q)^l \right\}$$

Minimizing over  $s$  gives

$$\frac{1}{2} P_A(l) \leq \left( \frac{lq}{lq - b} \right)^l \left( \frac{q(lp + b)}{p(lq - b)} \right)^{-lp - b}$$

or

$$\begin{aligned} \ln \frac{1}{2} P_A(l) &\leq l \ln \left( \frac{lq}{lq - b} \right) + (-lp - b) \ln \left( \frac{q(lp + b)}{p(lq - b)} \right) \\ &= l \ln \left( 1 + \frac{b}{lq - b} \right) + (-lp - b) \ln \left( 1 + \frac{b}{p(lq - b)} \right) \\ &\leq \frac{b^2 (3l^2 q^2 p + lb(7p^2 - 6p - 3) + b^2(6p + 3))}{6p^2(b - lq)^3} \\ &\leq -\frac{b^2}{2lpq} + O(1) \frac{b^3}{l^2} \\ &= -\tau \ln 2 - \frac{3}{2} \ln l + O(1) \frac{\ln^{3/2} l}{\sqrt{l}} \tau^{3/2}, \end{aligned} \quad (12)$$

where we have used  $x - \frac{x^2}{2} \leq \ln(1 + x) \leq x - \frac{x^2}{2} + \frac{x^3}{3}$  for  $x \geq 0$ . The equation (12) directly leads to (9).

For  $p = \frac{1}{2}$  Hoeffding's inequality [38] gives the bound

$$\begin{aligned} P_A(l) &\leq 2 \exp\left(-2 \frac{b^2}{l}\right) \\ &= 2 \exp\left(-\ln 2 \left(\tau + \frac{3}{2} \log l\right)\right) \end{aligned} \quad (13)$$

for  $p = \frac{1}{2}$  this is tighter than (12).

For the lower bound we use moderate deviations from [39]. Define  $\tilde{X}_i = \frac{X_i - p}{\sqrt{pq}}$ . We can then rewrite (7) as

$$\left| \frac{\sum_{i=1}^l \tilde{X}_i}{\sqrt{l(2\tau \ln 2 + 3 \ln l)}} \right| > 1$$

We define  $a_l = \frac{1}{2\tau \ln 2 + 3 \ln l}$ , which satisfies  $\lim_{l \rightarrow \infty} a_l = 0$ ,  $\lim_{l \rightarrow \infty} la_l = \infty$ . Using this as  $a_l$  in [39, Theorem 3.7.1] gives

$$\begin{aligned} & \liminf_{l \rightarrow \infty} \frac{1}{2\tau \ln 2 + 3 \ln l} \ln P \left( \left| \frac{\sum_{i=1}^l \tilde{X}_i}{\sqrt{l(2\tau \ln 2 + 3 \ln l)}} \right| > 1 \right) \\ &= \liminf_{l \rightarrow \infty} \frac{1}{3 \ln l} \ln P \left( \left| \frac{\sum_{i=1}^l \tilde{X}_i}{\sqrt{l(2\tau \ln 2 + 3 \ln l)}} \right| > 1 \right) \\ &\geq -\frac{1}{2} \end{aligned}$$

Together with the upper bound, this gives (11). ■

Figure 1 compares the upper bound with simulations.

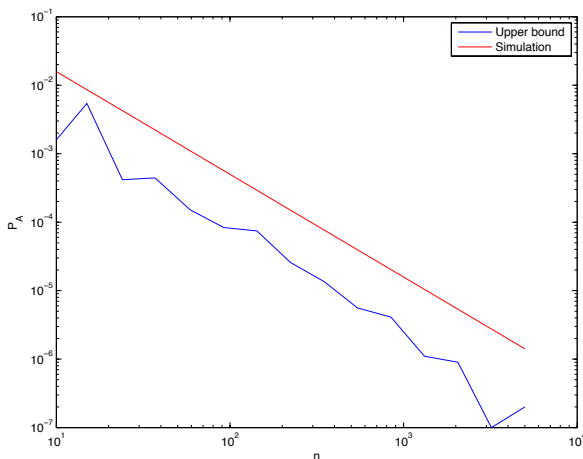


Fig. 1. Simulated  $P_A$  and the Upper bound for  $\tau = 1, p = 0.3$ .

We can also bound the miss probability for *extrinsically* atypical sequences as follows

**Theorem 3.** *Suppose that the typical sequence is iid  $\{0, 1\}$ -sequence with  $P(X_n = 1) = p$ . Let the test sequence be iid with  $P(X_n = 1) = p_a$ . The probability that the test sequence is missed according to criterion (6) is upper*

bounded by

$$P_M(l) \leq 2^{-\tau} \frac{1}{l^{3/2}} \left( \frac{q_a p}{p_a q} \right)^{\sqrt{lpq(2\tau \ln 2 + 3 \ln l)}} \times \left( \frac{q_a^{p-1} q^{p+1}}{p_a^p p^p} \right)^{-l} K(l, \tau) \quad (14)$$

$$\forall \tau : \lim_{l \rightarrow \infty} K(l, \tau) = 1$$

*Proof:* We may assume that  $p_a < p$ . Similarly to the proof of Theorem 2 the Chernoff bound is

$$P_M(l) \leq \inf_{s>0} \left\{ \exp(-s(pl - b)) (p_a e^s + q_a)^l \right\}$$

Minimizing over  $s$  gives

$$P_M(l) \leq \left( \frac{lq_a}{lq + b} \right)^l \left( \frac{q_a(lp - b)}{p_a(lq + b)} \right)^{-lp+b}$$

or

$$\begin{aligned} \ln P_M(l) &\leq l \ln \left( \frac{lq_a}{lq + b} \right) + (-lp + b) \ln \left( \frac{q_a(lp - b)}{p_a(lq + b)} \right) \\ &\leq l \ln \left( \frac{q_a}{q} \right) - lp \ln \left( \frac{q_a}{p_a} \right) - lp \ln \left( \frac{q}{p} \right) \\ &\quad + b \left( \ln \left( \frac{p}{q} \right) + \ln \left( \frac{q_a}{p_a} \right) \right) - \frac{b^2}{2lpq} + O \left( \frac{b^3}{l^2} \right) \end{aligned}$$

using series expansions. ■

#### A. Hypothesis testing interpretation

The solution (5) may seem arbitrary, but it has a nice interpretation in terms of hypothesis testing [40]. Return to the solution (1). That solution gives a test for a given  $l$ . However, the problem is that it does not reconcile tests for different  $l$ . One way to solve that issue is to consider  $l$  a random variable, i.e., introducing a prior distribution in the Bayesian sense. Let the prior distribution of  $l$  be  $P_L(l)$ . The equation (1) now becomes

$$\begin{aligned} L &= \log \frac{\prod_{b=0}^1 \hat{P}(b)^{N(b|x^l)} P_L(l)}{\prod_{b=0}^1 P(b)^{N(b|x^l)} P_L(0)} \\ &= l \sum_{b=0}^1 \hat{P}(b) \log \frac{1}{l} N(b|x^l) \\ &\quad - l \sum_{b=0}^1 \hat{P}(b) \log P(b) + \log P_L(l) - \log P_L(0) \\ &= lD(\hat{p}||p) + \log P_L(l) - \log P_L(0) \end{aligned}$$

The hypothesis test now is

$$D(\hat{p}||p) > \frac{\tau + \log P_L(0) - \log P_L(l)}{l} \quad (15)$$

Of course, the problem is that we don't know  $P(l)$ . Still, compare that with (5) without the approximations,

$$D(\hat{p}\|p) > \frac{\tau + \frac{1}{2} \log l + c + \log^* l}{l} \quad (16)$$

To the term  $c + \log^* l$  corresponds a distribution on the integers, namely  $Q(l)$  in [18, (3.6)]. Except for the term  $\frac{1}{2} \log l$ , the equations (15) and (16) are identical if we use the prior distribution  $P_L(l) = Q(l)$ . Rissanen [18] argues that the distribution  $Q(l)$  is the most reasonable distribution on the integers when we have really no prior knowledge, mainly from a coding point of view. This therefore seems a reasonable distribution for  $P(l)$ . What about the term  $\frac{1}{2} \log l$ ? The model for the non-null hypothesis has one unknown parameter,  $p$ , so that it is more complex than the null hypothesis. We have to account for this additional complexity. Our goal is to find an explanation for atypical sequences among a large class of explanations, not just the distribution of zeros and ones. If there is no ‘‘penalty’’ for finding a complex explanation, any data can be explained, and all data will be atypical. This is Occam’s razor [16]. The ‘‘penalty’’ for one unknown parameter as argued by Rissanen is exactly  $\frac{1}{2} \log l$ . We therefore have the following explanation for (5),

**Fact 4.** *The criterion (5) can be understood as a hypothesis test with prior distribution  $Q(l)$  [18] and penalty  $\frac{1}{2} \log l$  for the unknown parameter.*

Seen in this light, Theorem 2 is not surprising. In (5) we have replaced  $\frac{1}{2} \log l + \log^* l$  with  $\frac{3}{2} \log l$ , which implicitly corresponds to the prior distribution  $P_L(l) \sim l^{-3/2}$ , which is exactly the distribution seen in (9).

### B. Atypical subsequences

One problem where we believe our approach excels is in finding atypical *subsequences* of long sequences. The difficulty in find atypical subsequences is that we may have short subsequences that deviate much from the typical model, and long subsequences that deviate little. How do we choose among these? Definition 1 gives a precise answer. For the formal problem statement, consider a sequence  $\{x_n, n = -\infty, \dots, \infty\}$  from a finite alphabet  $\mathcal{A}$  (where in this section  $\mathcal{A} = \{0, 1\}$ ). The sequence is generated according to a probability law  $\mathcal{P}$ , which is known. In this sequence is embedded (infrequent) finite subsequences  $\mathcal{X}_i = \{x_n, n = n_i, \dots, n_i + l_i - 1\}$  from the finite alphabet  $\mathcal{A}$ , which are generated by an alternative probability law  $\tilde{\mathcal{P}}_\theta$ . The probability law  $\tilde{\mathcal{P}}_\theta$  is unknown, but it might be known to be from a certain class of probability distributions, for example parametrized by the parameter  $\theta$ . Each subsequence  $\mathcal{X}_i$  may be drawn from a different probability law. The problem we consider is to isolate these subsequences, which we call atypical subsequences. In this section, as above, we will assume both  $\mathcal{P}$  and  $\tilde{\mathcal{P}}_\theta$  are binary iid.

The solution is very similar to the one for variable length sequences above. The atypical subsequences are encoded with the universal source coder from [16] with a code length  $L_a = lH(\hat{p}) + \frac{1}{2} \log l$ . The start of the sequence is encoded with an extra symbol ‘.’ which has a code length  $-\log P(‘.’)$  and the length is encoded in  $\log^* l$  bits. In conclusion we end up with exactly the same criterion as (5), repeated here

$$D(\hat{p}\|p) > \frac{\tau + \frac{3}{2} \log l}{l} \quad (17)$$

The only difference is that  $\tau$  has a slight different meaning.

For the subsequence problem, a central question is what the probability is that a given sample  $x_n$  is part of an (intrinsically) atypical subsequence. Notice that there are infinitely many subsequences that can contain  $x_n$ , and each of these have a probability of being atypical given by Theorem 2.

We can obtain an upper bound as follows. Let us say that  $X_n$  has been determined to be part of an atypical sequence  $\mathcal{X}_i$ . It is clear that the sequence  $\mathcal{X}_i$  must also be atypical according to (17). Therefore, we can upper bound the probability  $P_A(X_n)$  that  $X_n$  is part of an atypical sequence with the probability of the event (17), using the approximate criterion (6),

$$\exists n_1 \leq n < n_1 + l : \left| \frac{\sum_{i=n_1}^{n_1+l-1} X_i - p}{\sqrt{pql}} \right| > \sqrt{2\tau \ln 2 + 3 \ln l}$$

We can rewrite this as

$$\begin{aligned} \exists n_1 \leq n < n_1 + l : \left| \frac{\sum_{i=n_1}^{n_1+l-1} X_i - p}{\sqrt{pql}} \right| &> \sqrt{2\tau \ln 2 + 3 \ln l} \\ \exists n_1 \leq n < n_1 + l : \left| \sum_{i=n_1}^{n_1+l-1} X_i - p \right| &> \sqrt{lpq \ln 2(2\tau + 3 \log l)} \end{aligned}$$

We could upper bound this with a union bound using Theorem 2. However, it is quickly seen that this does not converge. The problem is that the events in the union bound are highly dependent, so we need a slightly more refined approach; this results in the following Theorem

**Theorem 5.** *Consider the case  $p = \frac{1}{2}$ . The probability  $P_A(X_n)$  that a given sample  $X_n$  is part of an atypical subsequence is upper bounded by*

$$P_A(X_n) \leq (K_1\sqrt{\tau} + K_2)2^{-\tau} \quad (18)$$

for some constants  $K_1, K_2$ .

*Proof:* Without loss of generality we can assume  $n = 0$ . For some  $l_0 > 0$  let  $\mathcal{I}_{l_0}$  be the set of subsequences containing  $X_0$  of length  $l \leq l_0$ . For  $i \in \mathcal{I}_{l_0}$  let  $l(i)$  be the length of the subinterval. From Theorem 2 we know that  $P_A(i) \leq 2^{-\tau+1} \frac{1}{l^{3/2}} K(l, \tau)$  and therefore

$$\sum_{i \in \mathcal{I}_{l_0}} P_A(i) \leq K2^{-\tau}$$

for some constant  $K$ . This argument does not work if we allow arbitrarily long subsequences, because the sum is divergent. However, we can write

$$P_A(X_0) \leq \sum_{i \in \mathcal{I}_{l_0}} P_A(i) \leq K2^{-\tau} + P_{A,l_0}(X_0)$$

where  $P_{A,l_0}(X_0)$  is the probability that  $X_0$  is in an atypical subsequence of at least length  $l_0$ . The proof will be to bound  $P_{A,l_0}(X_0)$ .

Define the following events

$$\begin{aligned}\bar{A}(n_1, l) &= \left\{ \sum_{i=n_1}^{n_1+l-1} X_i - p > \sqrt{lpq \ln 2(2\tau + 3 \log l)} \right\} \\ \underline{A}(n_1, l) &= \left\{ \sum_{i=n_1}^{n_1+l-1} X_i - p < -\sqrt{lpq \ln 2(2\tau + 3 \log l)} \right\}\end{aligned}$$

For  $p = \frac{1}{2}$  we can rewrite

$$\begin{aligned}& \sum_{i=n_1}^{n_1+l-1} X_i - p > \sqrt{lpq \ln 2(2\tau + 3 \log l)} \\ &= \sum_{i=n_1}^{n_1+l-1} (2X_i - 1) > \sqrt{l \ln 2(2\tau + 3 \log l)}\end{aligned}\tag{19}$$

For ease of notation define

$$v(l) = \left\lceil \sqrt{l \ln 2(2\tau + 3 \log l)} \right\rceil$$

Then using the union bound we can write

$$\begin{aligned}P_A(X_0) &\leq \sum_{n_1=-\infty}^{-1} P \left( \bigcup_{l=-n_1+1}^{\infty} \bar{A}(n_1, l) \right) \\ &\quad + \sum_{n_1=-\infty}^{-1} P \left( \bigcup_{l=-n_1+1}^{\infty} \underline{A}(n_1, l) \right) \\ &\leq 2 \sum_{n_1=-\infty}^{-1} P \left( \bigcup_{l=-n_1+1}^{\infty} \bar{A}(n_1, l) \right) \\ &= 2 \sum_{n_1=-\infty}^{-1} 1 - P \left( \bigcap_{l=-n_1+1}^{\infty} \bar{A}^c(n_1, l) \right) \\ &= 2 \sum_{n_1=-\infty}^{-1} \left( 1 - \prod_{l=-n_1+1}^{\infty} P \left( \bar{A}^c(n_1, l) \mid \bigcap_{\ell=-n_1+1}^{l-1} \bar{A}^c(n_1, \ell) \right) \right)\end{aligned}$$

where we have excluded the length one sequence consisting of  $X_0$  itself. Now consider  $P \left( \bar{A}^c(n_1, l) \mid \bigcap_{\ell=-n_1+1}^{l-1} \bar{A}^c(n_1, \ell) \right) = 1 - P \left( \bar{A}(n_1, l) \mid \bigcap_{\ell=-n_1+1}^{l-1} \bar{A}^c(n_1, \ell) \right)$ .

We can think of  $S_l = \sum_{i=n_1}^{n_1+l-1} (2X_i - 1)$  as a simple random [41], and we will use this to upper bound the probability  $P \left( \bar{A}(n_1, l) \mid \bigcap_{\ell=-n_1+1}^{l-1} \bar{A}^c(n_1, \ell) \right)$ . This probability can be interpreted as the probability that the random walk passes  $v(l)$  given that it was below  $v(\ell)$  at times  $-n_1 < \ell < l$ . But since the random walk can increase by at most one, and since the threshold is increasing with  $l$ , that means that at time  $l$  we must have  $S_l = v(l)$ . Furthermore, it is easy to see that the probability is upper bounded by the probability that  $S_l = v(l)$ .

given that the random walk is below  $v(l)$  at times  $-n_1 < \ell < l$ . Thus

$$\begin{aligned}
& P\left(\bar{A}(n_1, l) \mid \bigcap_{\ell=-n_1+1}^{l-1} \bar{A}^c(n_1, \ell)\right) \\
& \leq P(S_l = v(l) \mid S_\ell < v(l), -n_1 < \ell < l) \\
& = \frac{P(S_l = v(l), S_\ell < v(l), -n_1 < \ell < l)}{P(S_\ell < v(l), -n_1 < \ell < l)} \\
& \leq \frac{P(S_l = v(l), S_\ell < v(l), -n_1 < \ell < l)}{P(S_\ell < v(l), 0 \leq \ell < l)} \tag{20}
\end{aligned}$$

The denominator can be interpreted as the probability that the maximum of the random walk stays below  $v(l)$ , which by Theorem 2 can be expressed by

$$\begin{aligned}
P_D(l) &= P(S_\ell < v(l), 0 \leq \ell < l) \\
&= 1 - 2P(S_{l-1} \geq v(l+1) - 1) - P(S_{l-1} = v(l) - 1) \\
&\geq 1 - 2^{-\tau+c} l^{-3/2} \\
&\geq \frac{1}{2}
\end{aligned}$$

for  $\tau$  and  $l$  sufficiently large, and where  $c$  is some constant. Since, as discussed at the start of the proof, we can assume that  $l \geq l_0$ , we can choose  $l_0$  large enough that this is satisfied; furthermore, since  $P_N(l)$  is increasing in  $\tau$ , we can choose  $l_0$  independent of  $\tau$  as long as  $\tau$  is sufficiently large.

We will next upper bound the numerator in (20). This is the probability that we have a path that has stayed below  $v(l)$  at steps  $-n_1 < \ell < l$ , but then at step  $l$  hits  $v(l)$ . We will count such paths. We divide them into two groups that we count separately. The first group are all paths that start at zero and hit  $v(l)$  *first time* after  $l$  steps. The second group is more easily described in reverse time. Those are paths that start at  $v(l)$  at step  $l$ , then stay below  $v(l)$  until time  $\tilde{n} < 0$ , when they hit  $v(l)$  again, and finally hit 0 at time  $n_1$ . According to [41, Section 3.10] we can count all these paths by

$$N = \frac{v(l)}{l} N_l(0, v(l)) + \sum_{t=v(l)}^{-n_1+1} \frac{1}{l-t-1} N_{l-t-1}(1, 0) N_t(0, v(l)) \tag{21}$$

Where  $N_n(a, b)$  are the number of length  $n$  paths between  $a$  and  $b$ .

We need to upper bound the probability  $P(S_n = k)$  that a path starting a 0 hits  $k$  after  $n$  steps. We use [41,



Section 3.10] and [16, 13.2] to get

$$\begin{aligned}
P(S_n = k) &= N_n(0, k) 2^{-n} \\
&= \binom{n}{\frac{1}{2}(n+k)} 2^{-n} \\
&\leq \sqrt{\frac{n}{\pi \frac{1}{4}(n+k)(n-k)}} 2^{nH\left(\frac{\frac{1}{2}(n+k)}{n}\right)} 2^{-n} \\
&= \sqrt{\frac{4n}{\pi(n^2 - k^2)}} 2^{nH\left(\frac{\frac{1}{2}(n+k)}{n}\right)} 2^{-n}
\end{aligned}$$

We can bound the power of the exponent to 2 as follows

$$\begin{aligned}
&nH\left(\frac{\frac{1}{2}(n+k)}{n}\right) - n \\
&= n\left(H\left(\frac{1}{2} + \frac{1}{2}\frac{k}{n}\right) - 1\right) \\
&\leq -\frac{2}{\ln 2} n \left(\frac{1}{2}\frac{k}{n}\right)^2 \\
&= -\frac{1}{2\ln 2} \frac{k^2}{n}
\end{aligned} \tag{22}$$

Thus,

$$P(S_n = k) \leq \frac{2}{\sqrt{\pi n}} e_2\left(-\frac{1}{2\ln 2} \frac{k^2}{n}\right)$$

where  $e_2(x) = 2^x$ .

We will use this to bound the probability of set of paths in the second term in (21). We can bound

$$\begin{aligned}
P_2(n_1, l) &= \sum_{t=v(l)}^{-n_1+1} \frac{1}{l-t-1} N_{l-t-1}(1, 0) N_t(0, v(l)) 2^{-(l-1)} \\
&\leq \sum_{t=v(l)}^{-n_1+1} \frac{1}{l-t-1} \frac{2}{\sqrt{\pi(l-t-1)}} \\
&\quad \times e_2\left(-\frac{1}{2\ln 2} \frac{1}{l-t-1}\right) N_t(0, v(l)) 2^{-t} \\
&\leq \frac{4}{\pi(l+n_1-2)^{3/2}} \sum_{t=v(l)}^{-n_1+1} P(S_t = v(l))
\end{aligned}$$

Here the sum  $\sum_{t=v(l)}^{-n_1+1} P(S_t = v(l))$  when looked at in reverse time can be interpreted as the probability of a path starting at  $v(l)$  hits zero before time  $-n_1 + 1$ . We can write this as (See [41, Section 3.10])

$$\begin{aligned}
\sum_{t=v(l)}^{-n_1+1} P(S_t = v(l)) &= P(M_{-n_1+1} \geq v(l)) \\
&\leq 2P(S_{-n_1+1} \geq v(l))
\end{aligned}$$

We can use the proof of Theorem 2, specifically (13) to bound this by

$$P(M_{-n_1+1} \geq v(l)) \leq \exp\left(-\frac{2v(l)^2}{-n_1+1}\right)$$

Then

$$P_2(n_1, l) \leq \frac{K}{(l+n_1-2)^{3/2}} \exp\left(-\frac{v(l)^2}{2(-n_1+1)}\right) \sqrt{-n_1+1}$$

We will next bound the probability of the paths in the first term in (21). We have

$$\begin{aligned} P(S_l = v(l)) &\leq \sqrt{\frac{4l}{\pi(l^2 - v(l)^2)}} e_2 \left(-\frac{1}{2\ln 2} \frac{v(l)^2}{l}\right) \\ &\quad \sqrt{\frac{4}{\pi l \left(1 - \frac{v(l)^2}{l^2}\right)}} e_2 \left(-\frac{1}{2\ln 2} \frac{v(l)^2}{l}\right) \\ &= \frac{2}{\sqrt{\pi \left(1 - \frac{v(l)^2}{l^2}\right)}} 2^{-\tau} l^{-2} \end{aligned}$$

and

$$\begin{aligned} P_1(l) &= \frac{v(l)}{l} P(S_l = v(l)) \\ &\leq \sqrt{\frac{\ln 2(2\tau + 3\log l)}{l}} \frac{2}{\sqrt{\pi \left(1 - \frac{v(l)^2}{l^2}\right)}} 2^{-\tau} l^{-2} \\ &\leq \sqrt{\frac{8\tau \ln 2}{\pi \left(1 - \frac{v(l)^2}{l^2}\right)}} 2^{-\tau} l^{-5/2} \\ &\quad + \sqrt{\frac{12 \ln l}{\pi \left(1 - \frac{v(l)^2}{l^2}\right)}} \sqrt{\frac{4}{\pi}} 2^{-\tau} l^{-5/2} \end{aligned}$$

Thus

$$\begin{aligned} &\ln \left( \prod_{l=-n_1+1}^{\infty} P \left( \overline{A}^c(n_1, l) \mid \bigcap_{\ell=-n_1+1}^{l-1} \overline{A}^c(n_1, \ell) \right) \right) \\ &\geq \sum_{l=-n_1+1}^{\infty} \ln \left( 1 - \frac{P_1(l) - P_2(n_1, l)}{P_D(l)} \right) \\ &\geq K \sum_{l=-n_1+1}^{\infty} -P_1(l) - P_2(n_1, l) \\ &\doteq S(-n_1, \tau) \end{aligned}$$

and

$$\begin{aligned}
P_A(X_0) &\leq 2 \sum_{n_1=-\infty}^{-1} 1 - e^{S(-n_1, \tau)} \\
&\leq 2K \sum_{n_1=-\infty}^{-1} \sum_{l=-n_1+1}^{\infty} P_1(l) \\
&\quad + 2K \sum_{n_1=-\infty}^{-1} \sum_{l=-n_1+1}^{\infty} P_2(n_1, l)
\end{aligned} \tag{23}$$

where  $K > 0$  is some constant.

First we evaluate the sum of  $P_1(l)$ . The term  $\frac{v(l)^2}{l^2}$  is decreasing in  $l$ , so for sufficiently large  $l_1$ ,  $\frac{v(l)^2}{l^2} \leq \frac{1}{2}$ . We can evaluate the sum separately for  $l_0 \leq l \leq l_1$  and for  $l > l_1$ . Convergence depends only on the latter tail. The threshold  $l_1$  is increasing with  $\tau$ . If for example we put  $l_1 = 8\tau \ln 2$ , i.e., proportional to  $\tau$ , we have  $\frac{v(l)^2}{l^2} \leq \frac{1}{2}$  for  $\tau > 10$ . Therefore

$$\sum_{l=l_0}^{l_1} P_1(l) \leq K\tau 2^{-\tau}$$

For  $l > l_1$  we can write

$$P_1(l) = \sqrt{\frac{16\tau \ln 2}{\pi}} 2^{-\tau} l^{-5/2} + \sqrt{\frac{24 \ln l}{\pi}} \sqrt{\frac{4}{\pi}} 2^{-\tau} l^{-5/2}$$

Then (for  $-n_1 + 1 > l_1$ )

$$\begin{aligned}
&K \sum_{l=-n_1+1}^{\infty} P_1(l) \\
&\leq k_1 2^{-\tau} \sqrt{\frac{\ln(n-n_1)}{(-n_1)^3}} \\
&\quad + k_2 2^{-\tau} \operatorname{erfc} \left( \sqrt{\frac{3}{2} \ln(-n_1)} \right) \\
&\quad + k_3 2^{-\tau} \sqrt{\frac{\tau}{(-n_1)^3}}
\end{aligned} \tag{24}$$

where  $k_i > 0$  are some constants and where we have used

$$\begin{aligned}
\sum_{l=k}^{\infty} l^{-5/2} &\leq \int_{k-1}^{\infty} x^{-5/2} dx = \frac{2}{3(k-1)^{3/2}} \\
\sum_{l=k}^{\infty} \sqrt{\ln l} l^{-5/2} &\leq \int_{k-1}^{\infty} \sqrt{\ln x} x^{-5/2} dx \\
&= \frac{1}{9} \left( \sqrt{6\pi} \operatorname{erfc} \left[ \sqrt{\frac{3}{2} \ln k - 1} \right] + \frac{6\sqrt{\ln k - 1}}{(k-1)^{3/2}} \right)
\end{aligned}$$

as it can be verified that all three sums, when (24) is inserted in (23), are convergent, using  $\sum_{k=1}^{\infty} f(k) \leq f(1) + \int_1^{\infty} f(x) dx$ .

We bound the second sum in (23),

$$\begin{aligned} & \sum_{n_1=-\infty}^{-1} \sum_{l=-n_1+1}^{\infty} P_2(n_1, l) \\ &= \sum_{n_1=-\infty}^{-1} \sum_{l=-n_1+1}^{\infty} \frac{8\sqrt{-n_1+1}}{\pi(l+n_1-2)^{3/2}} \exp\left(-\frac{v(l)^2}{2(-n_1+1)}\right) \end{aligned}$$

We can ignore the small constants and write

$$\begin{aligned} P &= \sum_{n_1=-\infty}^{-1} \sum_{l=-n_1+1}^{\infty} \frac{8}{\pi(l+n_1)^{3/2}} \exp\left(-\frac{v(l)^2}{2(-n_1)}\right) \sqrt{-n_1} \\ &= \sum_{n_1=-\infty}^{-1} \sum_{l=-n_1+1}^{\infty} \frac{8\sqrt{-n_1}}{\pi(l+n_1)^{3/2}} 2^{-\frac{l\tau}{n_1}} l^{-\frac{3l}{n_1}} \\ &\leq \int_1^{\infty} \int_t^{\infty} \frac{8\sqrt{t}}{\pi(l-t)^{3/2}} 2^{-\frac{l\tau}{t}} l^{-\frac{3l}{t}} dl dt \\ &= \int_1^{\infty} \int_1^{\infty} \frac{8\sqrt{t}}{\pi t^{3/2}(\tilde{l}-1)^{3/2}} 2^{-\tilde{l}\tau} \tilde{l}^{-3\tilde{l}} t^{-3\tilde{l}} t d\tilde{l} dt \\ &= \int_1^{\infty} \int_1^{\infty} \frac{8}{\pi(\tilde{l}-1)^{3/2}} 2^{-\tilde{l}\tau} \tilde{l}^{-3\tilde{l}} t^{-3\tilde{l}} d\tilde{l} dt \\ &= \int_1^{\infty} \left( \int_1^{\infty} t^{-3\tilde{l}} dt \right) \frac{8}{\pi(\tilde{l}-1)^{3/2}} 2^{-\tilde{l}\tau} \tilde{l}^{-3\tilde{l}} d\tilde{l} \\ &= \int_1^{\infty} \frac{1}{3\tilde{l}-1} \frac{8}{\pi(\tilde{l}-1)^{3/2}} 2^{-\tilde{l}\tau} \tilde{l}^{-3\tilde{l}} d\tilde{l} \\ &= 2^{-\tau} \int_1^{\infty} \frac{1}{3\tilde{l}-1} \frac{8}{\pi(\tilde{l}-1)^{3/2}} 2^{-(\tilde{l}-1)\tau} \tilde{l}^{-3\tilde{l}} d\tilde{l} \end{aligned}$$

The remaining integral is clearly convergent, and decreasing in  $\tau$ . Therefore  $P \leq K2^{-\tau}$  ■

There are two important implications of Theorem 5. First is that for  $\tau$  sufficiently large,  $P_A(X_n) < 1$ , and in fact  $P_A(X_n)$  can be made arbitrarily small for large enough  $\tau$ . This is an important theoretical validation of Definition 1 and the resulting criterion (5) and (6). If the theory had resulted in  $P_A(X_n) = 1$  then everything would be atypical, and atypicality would be meaningless. That this is not trivially satisfied is shown by Proposition 6 just below. What that Proposition says is that if in the above equation instead of  $\frac{3}{2} \log l$  we had had  $\frac{1}{2} \log l$ , then everything would have been atypical. Now,  $\frac{1}{2} \log l$  corresponds to “forgetting” that the length of an atypical sequence also needs to be encoded for the resulting sequence to be decodable. Thus, it is the strict adherence to decodability that has lead to a meaningful criterion. So, although decodability at first seems unrelated to detection, it turns out to be of crucial importance. Similarly, at first the term  $\frac{3}{2} \log l$  may have seen arbitrary. However, this is just (within a margin) sufficient to ensure that not everything becomes atypical.

The second important implication of Theorem 5 is that it validates the meaning of  $\tau$ . The way we introduced  $\tau$  was as the number of bits needed to encode the fact that an atypical sequence starts, and therefore we should put  $\tau = -\log P(\text{atypical sequence starts})$ . Theorem 5 confirms that  $\tau$  has the desired meaning for purely random sequences. And the reason this is not trivial is that  $\tau$  was chosen from the probability of an atypical *sequence*, while Theorem 5 gives the probability of a *sample* being atypical.

**Proposition 6.** Consider the case  $p = \frac{1}{2}$ . Suppose instead of (7) we use the criterion

$$\left| \frac{\sum_{i=1}^l X_i - p}{\sqrt{\frac{1}{4}l}} \right| > \sqrt{2\tau \ln 2 + \alpha \ln l} \quad (25)$$

(with  $\alpha = 3$  giving (7)). Then if  $\alpha \leq 1$ , the probability that a given sample  $X_n$  is part of an atypical subsequence is  $P_A(X_n) = 1$ .

*Proof:* We can assume that  $n = 0$ . We will continue with the random walk framework from the proof of Theorem 5. Define the event

$$\begin{aligned} \bar{A}_l &= \left\{ \sum_{i=-l+1}^0 (2X_i - 1) > \sqrt{l \ln 2(2\tau + \alpha \log l)} \right\} \\ \underline{A}_l &= \left\{ \sum_{i=-l+1}^0 (2X_i - 1) < \sqrt{l \ln 2(2\tau + \alpha \log l)} \right\} \end{aligned}$$

and

$$v(l) = \left\lceil \sqrt{l \ln 2(2\tau + \alpha \log l)} \right\rceil$$

Then

$$P_A(X_0) \geq P \left( \bigcup_{l=0}^{\infty} \bar{A}_l \cup \bigcup_{l=0}^{\infty} \underline{A}_l \right)$$

Namely, we declare that  $X_0$  is atypical if it is the endpoint of an atypical sequence  $\{x[-l], x[-l+1], \dots, x[0]\}$  for some  $l$ . Clearly,  $X_0$  could be the start or midpoint of an atypical sequence, so this a rather loose lower bound.

Now we can write

$$\begin{aligned} &P \left( \bigcup_{l=0}^{\infty} \bar{A}_l \cup \bigcup_{l=0}^{\infty} \underline{A}_l \right) \\ &= 1 - P \left( \bigcap_{l=0}^{\infty} \bar{A}_l^c \cap \bigcap_{l=0}^{\infty} \underline{A}_l^c \right) \\ &= 1 - \prod_{l=0}^{\infty} P \left( \bar{A}_l^c \cap \underline{A}_l^c \left| \bigcap_{k=0}^{l-1} \bar{A}_k^c \cap \bigcap_{k=0}^{l-1} \underline{A}_k^c \right. \right) \\ &= 1 - \prod_{l=0}^{\infty} \left[ 1 - P \left( \bar{A}_l \cup \underline{A}_l \left| \bigcap_{k=0}^{l-1} \bar{A}_k^c \cap \bigcap_{k=0}^{l-1} \underline{A}_k^c \right. \right) \right] \end{aligned}$$

Consider the probability  $P \left( \bar{A}_l \cup \underline{A}_l \left| \bigcap_{k=0}^{l-1} \bar{A}_k^c \cap \bigcap_{k=0}^{l-1} \underline{A}_k^c \right. \right)$ . The only way the conditional event can happen is if

$S_{l-1} = v(l) - 1$  and  $X_l = 1$  or  $S_{l-1} = -v(l) + 1$  and  $X_l = -1$ . Here we have

$$\begin{aligned}
& P(S_{l-1} = v(l) - 1) \\
&= N_{l-1}(0, v(l) - 1) 2^{-l+1} \\
&= \binom{l-1}{\frac{1}{2}(l+v(l)-2)} 2^{-l+1} \\
&\geq \sqrt{\frac{l-1}{2(l+v(l)-2)(l-v(l))}} 2^{(l-1)H\left(\frac{\frac{1}{2}(l+v(l)-2)}{l-1}\right)} 2^{-l+1} \\
&= \sqrt{\frac{l-1}{2((l-1)^2 - (v(l)-1)^2)}} 2^{(l-1)H\left(\frac{\frac{1}{2}(l+v(l)-2)}{l-1}\right)} 2^{-l+1} \\
&\geq \sqrt{\frac{1}{2l}} 2^{(l-1)H\left(\frac{\frac{1}{2}(l+v(l)-2)}{l-1}\right)} 2^{-l+1}
\end{aligned}$$

Here

$$\begin{aligned}
& (l-1)H\left(\frac{\frac{1}{2}(l+v(l)-2)}{l-1}\right) - l + 1 \\
&= (l-1) \left( H\left(\frac{1}{2} + \frac{1}{2} \frac{v(l)-1}{l-1}\right) - 1 \right) \\
&\geq -\frac{2}{\ln 2} (l-1) \left( \frac{1}{2} \frac{v(l)-1}{l-1} \right)^2 + (l-1) o\left(\left(\frac{v(l)-1}{l-1}\right)^3\right) \\
&= -\frac{1}{2\ln 2} \frac{(v(l)-1)^2}{l-1} + \frac{(v(l)-1)^3}{(l-1)^2} \epsilon \left(\frac{1}{l}\right) \\
&\geq -\frac{1}{2\ln 2} \left( \frac{v(l)^2}{l} + \frac{v(l)^2}{(l-1)^2} + \frac{1}{l-1} \right) + \frac{v(l)^3}{l^2} \epsilon \left(\frac{1}{l}\right) \\
&= -\frac{1}{2\ln 2} \frac{v(l)^2}{l} + \frac{v(l)^3}{l^2} \epsilon \left(\frac{1}{l}\right) \\
&\geq -\frac{1}{2\ln 2} \frac{v(l)^2}{l} - \frac{v(l)^3}{l^2}
\end{aligned}$$

Where the last inequality is only true for  $l$  sufficiently large, as for some  $l_0$  we have  $\forall l > l_0 : |\epsilon(l^{-1})| < 1$ . Then

$$\begin{aligned}
P(S_{l-1} = v(l) - 1) &\geq \sqrt{\frac{1}{2l}} 2^{-\tau} l^{-\frac{\alpha}{2}} 2^{-\frac{v(l)^3}{l^2}} \\
&= \sqrt{\frac{1}{2}} 2^{-\tau} l^{-\frac{\alpha+1}{2}} 2^{-\frac{v(l)^3}{l^2}}
\end{aligned}$$

And

$$\begin{aligned}
& \ln \left( 1 - P \left( \bigcup_{l=0}^{\infty} \bar{A}_l \cup \bigcup_{l=0}^{\infty} \underline{A}_l \right) \right) \\
&\leq \sum_{l=1}^{\infty} \ln \left( 1 - \sqrt{\frac{1}{2}} 2^{-\tau} l^{-\frac{\alpha+1}{2}} 2^{-\frac{v(l)^3}{l^2}} \right) \\
&\leq \sum_{l=0}^{\infty} -\sqrt{\frac{1}{2}} 2^{-\tau} l^{-\frac{\alpha+1}{2}} 2^{-\frac{v(l)^3}{l^2}}
\end{aligned}$$

Here  $\lim_{l \rightarrow \infty} \frac{v(l)^3}{l^2} = 0$ . So, for example, for  $l$  sufficiently large,  $\frac{v(l)^3}{l^2} \leq 1$ . Then

$$\begin{aligned} & \ln \left( 1 - P \left( \bigcup_{l=0}^{\infty} \bar{A}_l \cup \bigcup_{l=0}^{\infty} A_l \right) \right) \\ & \leq \sum_{l=l_0}^{\infty} -\frac{1}{2} \sqrt{\frac{1}{2}} 2^{-\tau l} l^{-\frac{\alpha+1}{2}} \end{aligned}$$

This is divergent for  $\alpha \leq 1$  proving that  $P \left( \bigcup_{l=0}^{\infty} \bar{A}_l \cup \bigcup_{l=0}^{\infty} A_l \right) = 1$ . ■

Theorem 5 states that for  $\alpha = \frac{3}{2}$   $P_A(X_n) < 1$  (convergence), while Proposition 6 shows that for  $\alpha = \frac{1}{2}$   $P_A(X_n) = 1$  (divergence). There is a gap between those values of  $\alpha$  that is hard to fill in theoretically. We have therefore tested it out numerically, see Fig. 2. Of course, testing convergence numerically is not quite well-posed. Still the figure indicates that the phase transitions between divergence and convergence happens right around  $\alpha = 1$ .

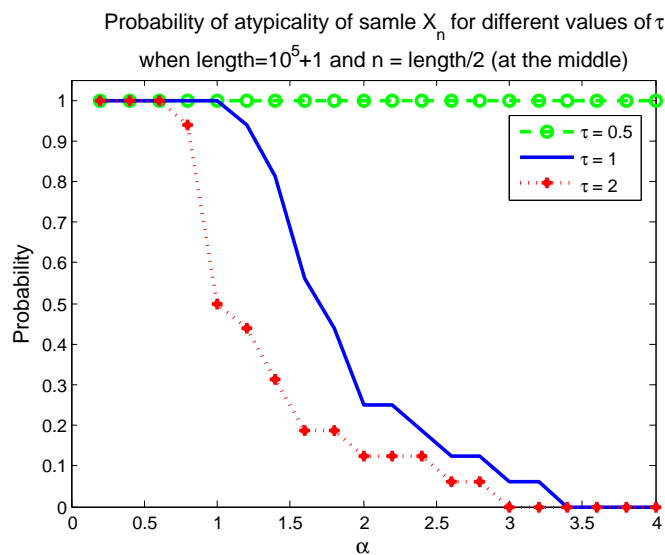


Fig. 2. Transition between divergence and convergence as a function of  $\alpha$

### C. Recursive coding

Instead of using Definition 1 directly, we could approach the problem as follows. First the sequence is encoded with the typical code. Now, if the distribution of the sequence is in agreement with the typical code, the results should be a sequence of iid binary bits with  $P(X_i = 1) = \frac{1}{2}$  [16], i.e., a purely random sequence; and this sequence cannot be further encoded. We can now try if we can further encode the sequence with a (universal) code. If so, we categorize the sequence as atypical. Let  $l^*$  be the length of the sequence after typical coding. In (4) the typical and atypical codelengths are therefore

$$\begin{aligned} L_t &= l^* - \log \epsilon \\ L_a &= l^* H(\hat{p}^*) + \frac{3}{2} \log l^* - \log \epsilon + \tau^* \end{aligned} \quad (26)$$

Here  $\hat{p}^*$  is the estimated  $p$  for the *encoded* sequence. Now

$$l^* = l \left( \hat{p} \log \frac{1}{p} + (1 - \hat{p}) \log \frac{1}{1-p} \right) \quad (27)$$

$$l^* H(\hat{p}^*) \sim l H(\hat{p}) \quad (28)$$

$$\log l^* = \log l + \log \left( \hat{p} \log \frac{1}{p} + (1 - \hat{p}) \log \frac{1}{1-p} \right) \quad (29)$$

$$\sim \log l \quad (30)$$

The argument for (28) is as follows (without doing detailed calculations): if we encode a sequence with a “wrong” code and then later re-encode with the “correct” code (for the induced statistic), the result is the same as originally encoding with the correct code. Thus the criterion (4) and (26) are *approximately* equivalent. We can state this as follows

**Proposition 7.** *Definition 1 can be applied to encoded sequences instead of the original data.*

This of course ignores all integer constraints, block boundaries etc. But the importance of this statement is that it is sometimes easier to operate on (partially) encoded sequences simply because the amount of data has already been reduced, and the problem has been standardized: as such, we do not need to know the typical codebook or even the model of typical data since everything under the typical model has been reduced to a stream of iid binary digits, and atypicality algorithms can therefore be applied to data streams without knowledge of what is the original data. It also means that theoretical results such as Theorem 5 where we assume typical data is iid uniform has general applicability.

However, first encoding the sequence and then doing atypicality detection also has disadvantages in a practical, finite length setting. Atypical subsequences become embedded in typical sequences in unpredictable ways. For example, it could be difficult to determine where exactly an atypical sequence starts and ends. Our practical implementation therefore uses Definition 1 directly.

#### IV. GENERAL CASE

Return to the problem considered at the start of Section III where we are given a sequence  $x$  of fixed length  $l$  and we need to determine if it is atypical. In the iid case this is a simple hypothesis test problem and the solution is given by (1). In the general case we would like find to alternative explanations from a large abstract class of models. The issue is that it is often possible to fit an alternative model very well to the data if we just allow complex enough models – the well known Occam’s razor problem [16]. Rissanen’s MDL [42], [18], [19] is a solution to this problem. Therefore, in the general case, even for fixed length sequences, the problem is not a straightforward hypothesis test problem, and we have to resort to information theory.



### A. Finite State Machines

One possible class of models in the general case is the class of finite state machines (FSM). Rissanen [17] defines the complexity of a sequence  $x^l$  in the class of FSM sources by

$$I(x^l) = \min\{-\log \hat{P}(x^l|f_j) + \log^* j + c\} \quad (31)$$

where  $f_1, f_2, \dots$  is a sequence of state machines, and where we have used  $\hat{P}(x^l|f_j)$  to emphasize that the probability is estimated. Rissanen uses Laplace's estimator, but the KT-estimator [43], [16] could also be used. Except for integer constraints, this is a valid descriptive length, and can therefore be used in Definition 1. This is a natural extension of the iid case considered in Section III. As opposed to Kolmogorov complexity, this complexity could actually be calculated, although with high complexity. Because of the complexity, it is mostly useful for theoretical considerations, and one result is the following generalization of Theorem 2

**Theorem 8.** *Assume that the typical distribution is iid uniform. If the atypical descriptive length is given by (31) with a maximum number of states independent of  $l$ , the probability of an intrinsically atypical sequence  $P_A(l)$  satisfies*

$$\lim_{l \rightarrow \infty} \frac{\ln P_A(l)}{-\frac{3}{2} \ln l} = 1 \quad (32)$$

*Proof:* Since we consider all state machines with the number of states up to a certain maximum, this must also include the state machine with a single state. This is equivalent to the iid model in Section III, and we therefore get the lower bound in (32). The proof will be to upper bound the probability. As in Section III we use  $\log l + \tau$  bits to indicate beginning and end of atypical sequences. The probability that a sequence  $x^l$  is atypical therefore is

$$\begin{aligned} P_A(l) &= P(I(x^l) + \log l + \tau > l) \\ &= P\left(\bigcup_{f_j} -\log \hat{P}(x^l|f_j) + \log^* j + c + \log l + \tau > l\right) \\ &\leq \bigcup_{f_j} P(-\log \hat{P}(x^l|f_j) + \log l + \tau > l) \end{aligned}$$

We will prove that  $P(-\log \hat{P}(x^l|f_j) + \log l + \tau > l) \leq K_j l^{-(k+2)/2}$  for constants  $K_j$  and  $k$  the number of states in the state machine, and since the slowest decay dominates, we get the upper bound for (32).

For a fixed state machine  $f$  the code length according to [17, (3.6)] is

$$L(x^l|f) = \sum_s \log \left( \frac{n_s(x^l)}{n_{0|s}(x^l)} \right) + \sum_s \log(n_s(x^l) + 1)$$

where  $n_s(x^l)$  denotes the number of occurrences of state  $s$  in  $x^l$  and  $n_{0|s}(x^l)$  the number of times the next symbols

is 0 at this state. Further, from [16, 13.2]

$$\begin{aligned}
L(x^l|f) &\geq \sum_s n_s(x^l) H\left(\frac{n_{0|s}(x^l)}{n_s(x^l)}\right) - \frac{1}{2} \log n_s(x^l) \\
&\quad - \frac{1}{2} \log\left(8 \frac{n_{0|s}(x^l)}{n_s(x^l)} \frac{n_{1|s}(x^l)}{n_s(x^l)}\right) \\
&\quad + \log(n_s(x^l) + 1)
\end{aligned} \tag{33}$$

We want to upper bound the probability of the event  $L(x^l|f) + \log l + \tau < l$ . We can write

$$\log(n_s(x^l) + 1) - \frac{1}{2} \log(n_s(x^l)) = \frac{1}{2} \log l + \log\left(\frac{1}{l} \frac{(n_s(x^l) + 1)^2}{n_s(x^l)}\right).$$

Let

$$r(x^l) = \sum_s n_s(x^l) \left( H\left(\frac{n_{0|s}(x^l)}{n_s(x^l)}\right) - 1 \right)$$

and let  $R(x^l)$  be the remaining small terms in (33) dependent on  $x^l$ ,

$$\begin{aligned}
R(x^l) &= \sum_s -\frac{1}{2} \log\left(8 \frac{n_{0|s}(x^l)}{n_s(x^l)} \frac{n_{1|s}(x^l)}{n_s(x^l)}\right) \\
&\quad + \log\left(\frac{1}{l} \frac{(n_s(x^l) + 1)^2}{n_s(x^l)}\right).
\end{aligned}$$

Then we have to upper bound (notice that  $\sum_s n_s(x^l) = l$ ),

$$P\left(-r(x^l) - R(x^l) \geq \tau + \frac{k+2}{2} \log l\right)$$

The Chernoff bound is

$$\begin{aligned}
&P\left(-r(x^l) - R(x^l) \geq \tau + \frac{k+2}{2} \log l\right) \\
&\leq \exp\left(-t\left(\tau + \frac{k+2}{2} \log l\right)\right) M(t)
\end{aligned}$$

or

$$\begin{aligned}
&\ln P\left(-r(x^l) - R(x^l) \geq \tau + \frac{k+2}{2} \log l\right) \\
&\leq -t\left(\tau + \frac{k+2}{2} \log l\right) + \ln M(t)
\end{aligned}$$

where

$$M(t) = E\left[\exp(-t(r(x^l) + R(x^l)))\right]$$

In order to get a valid bound, we need to show that  $M(t) < K < \infty$  independent of  $l$  for  $t < \ln 2$ . Now it's easy to see that  $\exp(-tR(x^l)) \leq K < \infty$  for all  $t$  and  $l$ . So, we have to show

$$E\left[\exp(-t(r(x^l)))\right] \leq K < \infty$$

We have to show that this is true for all state machines in the class of finite state machines with  $k$  states, which can be done by showing

$$\max_{\text{FSM with } k \text{ states}} E [\exp(-t(r(x^l)))] \leq K$$

It turns out it is easier to prove this if we expand the class over which we take the maximum, and clearly expanding the class does not decrease the maximum. A FSM with  $k$  states is a function  $f(x^l) \in \{1, \dots, k\}$  that satisfies that if  $f(x^m) = f(\tilde{x}^m) = s$  then  $f(x^{mb}) = f(\tilde{x}^{mb})$  for any bit  $b$  [17], i.e., if the FSM is in state  $s$  after  $m$  steps, the next state transition is only dependent on the next bit, not how it got to state  $s$ . We extend the class by dispensing with this requirement. We can then describe the 'program' we run as follows. Based on  $x^m$  we choose a state  $s_m \in \{1, \dots, k\}$  *without* having any knowledge about  $x_{m+1}$ , except that it is independent and uniformly distributed (by the assumption on the typical distribution). We can think of this slightly differently. The program puts  $x_{m+1}$  into bucket  $s \in \{1, \dots, k\}$  and updates  $n_s(m)$  and  $n_{i|s}(m)$ , in order to maximize  $E [\exp(-t(r(x^l)))]$ . It does so based on past data  $x^m$ . Now, as opposed to the state machine setup, the choice of  $s_m$  in no way restricts the choices of states (or buckets)  $s_n$ ,  $n > m$ . Since the program has no knowledge of  $x_{m+1}$  the program cannot optimize  $s_m$  based on the *values* of  $x^m$ . Rather, it is sufficient to look at  $n_s(m)$ . It is now easy to see that the worst case is obtained if the bits are distributed evenly in the states. Thus, the worst case of  $r(x^l)$  is

$$r(x^l) = \sum_s \frac{l}{k} \left( H \left( \frac{n_{0|s}(x^l)}{l/k} \right) - 1 \right)$$

where the  $n_{0|s}(x^l)$  are independent of  $s$ . Thus, the problem is reduced to the case of a single state, which is showing that

$$E \left[ \exp \left( tl \left( 1 - H \left( \frac{k}{l} \right) \right) \right) \right] \leq K < \infty \quad (34)$$

Here we have

$$\begin{aligned} & E \left[ \exp \left( tl \left( 1 - H \left( \frac{k}{l} \right) \right) \right) \right] \\ &= \sum_{k=0}^l 2^{\frac{t}{\ln 2} l (1 - H(\frac{k}{l}))} \binom{l}{k} 2^{-l} \\ &= 1 + 2 \sum_{k=1}^{l/2} 2^{\frac{t}{\ln 2} l (1 - H(\frac{k}{l}))} \binom{l}{k} 2^{-l} \\ &\leq 1 + 2 \sum_{k=1}^{l/2} 2^{\frac{t}{\ln 2} l (1 - H(\frac{k}{l}))} 2^{-l} 2^{lH(\frac{k}{l})} \sqrt{\frac{l}{\pi k(l-k)}} \\ &= 1 + 2 \sum_{k=1}^{l/2} 2^{(1 - \frac{t}{\ln 2}) l (1 - H(\frac{k}{l}))} \sqrt{\frac{l}{\pi k(l-k)}} \\ &\leq 1 + 2 \sum_{k=1}^{l/2} 2^{-(1 - \frac{t}{\ln 2}) \frac{2}{\ln 2} l (\frac{k}{l} - \frac{1}{2})^2} \sqrt{\frac{l}{\pi k(l-k)}} \end{aligned}$$

where we have used [16, 13.2] and (22). The sum is actually decreasing as a function of  $l$ , but this seems hard to prove. Instead we upper bound the sum by

$$\begin{aligned} & \sum_{k=1}^{l/2} 2^{-\left(1-\frac{t}{\ln 2}\right)\frac{2}{\ln 2}l\left(\frac{k}{l}-\frac{1}{2}\right)^2} \sqrt{\frac{l}{\pi k(l-k)}} \\ & \leq \int_1^{l/2} 2^{-\left(1-\frac{t}{\ln 2}\right)\frac{2}{\ln 2}l\left(\frac{k}{l}-\frac{1}{2}\right)^2} \sqrt{\frac{l}{\pi k(l-k)}} dk \end{aligned}$$

Here we can upper bound  $\sqrt{\frac{l}{\pi k(l-k)}} \leq \frac{4k\left(\sqrt{\frac{1}{l}}-1\right)+2}{\sqrt{\pi}}$  for  $1 \leq k \leq \frac{l}{2}$ . Then

$$\begin{aligned} & \int_1^{l/2} 2^{-\left(1-\frac{t}{\ln 2}\right)\frac{2}{\ln 2}l\left(\frac{k}{l}-\frac{1}{2}\right)^2} \sqrt{\frac{l}{\pi k(l-k)}} dk \\ & \leq \int_{1/l}^{1/2} 2^{-\left(1-\frac{t}{\ln 2}\right)\frac{2}{\ln 2}l\left(x-\frac{1}{2}\right)^2} \frac{4x\left(\sqrt{\frac{1}{l}}-1\right)+2}{\sqrt{\pi}} l dx \\ & \leq \int_{-\infty}^{1/2} 2^{-\left(1-\frac{t}{\ln 2}\right)\frac{2}{\ln 2}l\left(x-\frac{1}{2}\right)^2} \frac{4x\left(\sqrt{\frac{1}{l}}-1\right)+2}{\sqrt{\pi}} l dx \\ & = \frac{K_1}{\sqrt{l}} + K_2 \end{aligned}$$

for some constants  $K_1, K_2$ , using Gaussian moments. This proves (34). ■

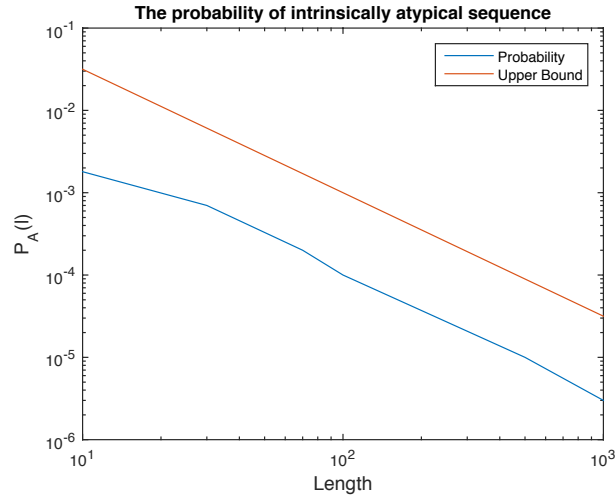


Fig. 3. Probability of an intrinsically atypical sequence. The typical distribution is iid uniform, and for detection of atypical sequences the CTW algorithm has been used (Section IV-B).

While the Theorem is for the typical model iid uniform, as outlined in Section III-C in principle it also applies to general sources, since we can first encode and then look for atypical sequences.

The theorem shows that looking for more complex explanations for data does not essentially increase the probability of intrinsically atypical sequences. Fig. 3 (compare with Fig. 1) confirms this experimentally. The

atypical detection is based on CTW, which as explained in Section IV-B below, is a good approximation of FSM modeling. On the other hand, if one of the FSM models do in fact fit the data, the chance of detecting the sequence is greatly increased, although hard to quantify. If we think of intrinsically atypical sequences as false alarms, this shows the power of the methodology.

Since FSM sources has the same  $P_A(l)$  as in the iid case, it seems reasonable to conjecture that Theorem 5 is still valid, that is  $P_A(X_n) < 1$  for sufficiently large  $\tau$ , which is clearly an essential theoretical property of atypicality. However, as Theorem 5 does not follow directly from Theorem 2, to verify the conjecture requires a formal proof which we do not have at present.

### B. Atypical Encoding

In terms of coding, Definition 1 can be stated in the following form

$$C(x|\mathcal{P}) - C(x) > 0$$

Here  $C(x|\mathcal{P})$  is the code length of  $x$  encoded with the optimum coder according to the typical law, and  $C(x)$  is  $x$  encoded 'in itself.' As argued in Section III, we need to put a 'header' in atypical sequences to inform the encoder that an atypical encoder is used. We can therefore write  $C(x) = \tau + \tilde{C}(x)$ , where  $\tau$  is the number of bits for the 'header,' and  $\tilde{C}(x)$  is the number of bits used for encoding the data itself. For encoding the data itself an obvious solution is to use a universal source coder. There are many approaches to universal source coding: Lempel-Ziv [16], [44], [45], Burrows-Wheeler transform [46], partial predictive mapping (PPM) [47], [48], or T-complexity [49], [6], [50], [51], [52], [53], [54], and anyone of them could be applied to the problem considered in this paper. The idea of atypicality is not linked to any particular coding strategy. In fact a coding strategy does not need to be decided. We could try several source coders and choose the the one giving the shortest code length; or they could even be combined as in [55]. However, to control complexity, we choose a single source coder. The most popular and simplest approach to source coding is perhaps Lempel-Ziv [16], [44], [45]. The issue with this is that while it is optimum in the sense that  $\limsup_{l \rightarrow \infty} \frac{C(x^l)}{l} = H(X)$  wp 1, the convergence is very slow. According to [56]  $E \left[ \frac{C(x^l)}{l} \right] - H(X) \sim \frac{1}{\log l}$  while  $\text{var} \left[ \frac{C(x^l)}{l} \right] \sim \frac{1}{l}$ . Thus, Lempel-Ziv is poor for short sequences, which is exactly what we are interested in for atypicality.

We have therefore chosen to use the Context Tree Weighing (CTW) algorithm [43]. The CTW approach has some advantages in our setup: it is a natural extension of the simple example considered in Section III, it allows estimation of code length without actually encoding, there is flexibility in how to estimate probabilities. Importantly, it can be seen as a practical implementation of the FSM based descriptive length used in Section IV-A.

### C. Typical Encoding and Training

In Definition 1 and the example in Section III we have assumed that the typical model of data is exactly known. If that is the case, typical encoding is straightforward, using for example arithmetic coding – notice that we just need codelength, which can be calculated for arithmetic coding without actually encoding. However, in many cases

the typical model is not known exactly. In the simplest case, the typical model is from a small class parametrized with a few parameters (e.g., binary iid with  $P(X = 1)$  unknown). In that case, finding the typical model is a simple parameter estimation problem, and we will not discuss this further. We will focus on the case where the typical model is not given by a specific model. In that case, it seems obvious to also use universal source coding for typical data. However, this is not straightforward if we want to stay faithful to the idea of Definition 1 as we will argue below.

Let  $\mathcal{P}$  be the typical model and  $\hat{\mathcal{P}}$  be the estimated typical model. When no specific model is given, in principle  $\hat{\mathcal{P}}$  could be given by an estimate of the various joint probability mass functions. However, a more useful approach is to estimate the conditional probabilities  $p(x_n = 1|x_{n-1}, x_{n-2}, \dots, x_{n-N})$ , where  $s = x_{n-1}, x_{n-2}, \dots, x_{n-N}$  is called the context. If the source has finite memory these probabilities characterize the source, and otherwise they could give a good approximate model. The issue is that there are  $2^N$  possible contexts, so for  $N$  even moderately large the amount of training data required to even observe every context is large, and to get good estimates for every context it is even larger. Realistically, therefore not every probability  $p(x_n = 1|s)$  can be estimated. This is an issue universal source coding is designed to deal with, and we therefore turn to universal source coding.

Let us assume we are given a single long sequence  $t$  for training – rather than a model  $\mathcal{P}$  – and based on this we need to encode a sequence  $x$ . Let us denote this coder as  $C(x|t)$ . To understand what this means, we have to realize that when  $x$  is encoded according to  $C(x|\mathcal{P})$  with a known  $\mathcal{P}$ , the coding probabilities are *fixed*; they are not affected by  $x$ . As discussed in Section II, this is an important part of Definition 1 that reacts to ‘outliers,’ data that does not fit the typical model. If the coding probabilities for  $C(x|\mathcal{P})$  were allowed to depend on  $x$ , in extreme case we would always have  $C(x) = C(x|\mathcal{P})$ .

The issue with universal source coders is that they often easily adapts to new types of data, a desirable property of a good universal source coder, but problematic in light of the above discussion. We therefore need to ‘freeze’ the source coder, for example by not updating the dictionary. However, because the training data is likely incomplete as discussed above, the freezing should not be too hard. The resulting encoder  $C(x|t)$  is not a universal source coder, but rather a training based fixed source coder, and the implementation can be quite different from a universal source coder, requiring careful consideration.

We will suggest one algorithm based on the principle of the CTW algorithm. This naturally complements using the CTW algorithm for atypical encoding, but could also be used with other atypical encoders. The following discussion requires a good knowledge of the CTW algorithm, which is most easily obtained from [57].

The algorithm is based on estimating  $P(x_n = 1|s)$  for contexts  $s$ . The estimate for a given  $s$  is done with the KT-estimator [43], [16]  $P(x_n = 1|s) = \frac{b_s + 1/2}{a_s + b_s + 1}$ , where  $a_s$  and  $b_s$  are the number of 0s and 1s respectively seen with context  $s$  in the training data  $t$ , but unaffected by the test sequence  $x$ . As discussed above, the complication is that not every context  $s$  might be seen and that in particular long contexts  $s$  are rarely seen so that the estimates  $P(X_n = 1|s)$  might be more accurate for shorter contexts. We solve that with the weighting idea of the CTW algorithm; the weights can be thought of as a prior distribution on different models. We can summarize this as follows. For every context  $s$ , the subsequence associated with  $s$  could either be memoryless or it could have memory

[57]. We call the former model  $M_1$  and the latter  $M_2$ . The CTW algorithm uses a prior distribution, weights, on the models  $P(M_1) = P(M_2) = \frac{1}{2}$ . Our basic idea is to weigh with  $P(M_1|t)$  and  $P(M_2|t)$  instead of  $\frac{1}{2}$ .

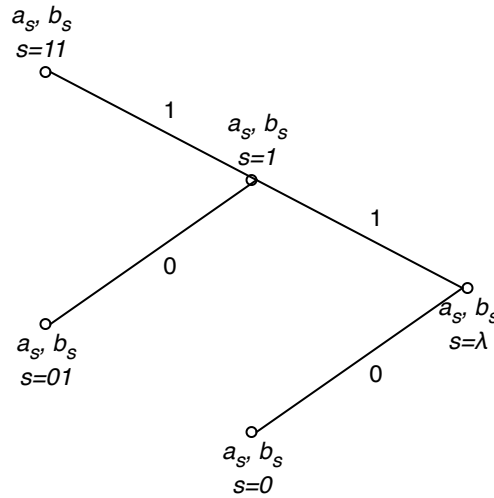


Fig. 4. Example context tree.

The algorithm is best described through an example. We have trained the algorithm with  $t$ , resulting in the context tree seen in Fig. 4. Now suppose we want to find a coding distribution  $P(1|010)$  for the actual data. We begin at the root and calculate

$$P(1|010, t) = P(1|010, M_1)P(M_1|t) + P(1|010, M_2)P(M_2|t)$$

$$P(M_i|t) = \frac{P(t|M_i)P(M_i)}{P(t)} = \frac{P(t|M_i)}{P(t|M_1) + P(t|M_2)}$$

$$P(t|M_1) = P_e(a_s, b_s)$$

$$P(t|M_2) = P_w^{0s}(t)P_w^{1s}(t)$$

$$s = \lambda \quad (\text{empty context})$$

under model  $M_1$  the data is memoryless, so

$$P(1|010, M_1) = P(1|M_1) = \frac{b_s + 1/2}{a_s + b_s + 1}; \quad s = \lambda$$

To find  $P(1|010, M_2)$  we look in the 0-node of the context tree. Here we calculate similarly

$$\begin{aligned}
 P(1|010, t) &= P(1|010, M_1)P(M_1|t) + P(1|010, M_2)P(M_2|t) \\
 P(M_i|t) &= \frac{P(t|M_i)P(M_i)}{P(t)} = \frac{P(t|M_i)}{P(t|M_1) + P(t|M_2)} \\
 P(t|M_1) &= P_e(a_s, b_s) \\
 P(t|M_2) &= P_w^{0s}(t)P_w^{1s}(t) \\
 s &= 0
 \end{aligned}$$

and again under model  $M_1$  the data is iid, so

$$P(1|10, M_1) = P(1|0, M_1) = \frac{b_s + 1/2}{a_s + b_s + 1}; \quad s = 0$$

and so on. From the context tree it is seen that the context 10 has not been seen before; then the context 010 has not been seen either. Then we have

$$P(1|010, M_2) = \frac{1}{2}.$$

No more look-up is needed, and the calculation has completed.

The algorithm can be implemented as follows. We run the standard CTW algorithm on the training data. To freeze, in each node corresponding to the context  $s$  we can pre-compute  $P(M_1|t)$  and  $P(1|s, M_1)$ . This is the only data that needs to be stored. While the algorithm is described from the root and up, implementation is simpler (no recursion) from the top and down to the root.

Often the source coder might be trained with many separate sequences, rather than one long sequence. This is not an issue, but care has to be taken with the startup for each sequence. The original CTW paper [43] assumes that a context of length at least  $D$  is available prior to the start of the sequence, which is not true in practice. The paper [58] solves this by introducing an indeterminate context. A context may start with an indeterminate context, but at most once. With multiple training sequences this could happen more than once. A better approach is therefore to use the start of each training sequence purely as a context (i.e., not code it). This wastes some training bits, but if the sequences are long the loss is minor. A different case would be if we need to find short atypical sequences rather than subsequences. In that case a more careful treatment of start of sequences would be needed.

Freezing the encoder is essential in implementing atypicality. A simulation confirming this is shown in Fig. 5. The CTW algorithm is trained with a three-state Markov chain with transition probability  $[\begin{smallmatrix} .05 & .95 & 0 \\ 0 & .05 & .95 \\ .05 & & \end{smallmatrix}]$  while generating  $\{0,1\}$  according to  $[0 \ 1 \ x; x \ 1 \ 0; 1 \ x \ 0]$ , so this Markov chain mostly generates the following pattern:  $[1 \ 0 \ 1]$ . The test sequence is generated by another three-state Markov chain with the same transition probability but generating  $\{0,1\}$  according to  $[0 \ 1 \ x; x \ 1 \ 0; 0 \ x \ 1]$ , i.e., generating mostly  $[1 \ 0 \ 0]$  as pattern. With the non-frozen algorithm the code length difference between typical and atypical encoding is so small that it can easily be missed, although the difference in the patterns themselves in the raw data is clearly visible to the naked eye. The reason the non-frozen algorithm does not work is that it quickly learns the new  $[1 \ 0 \ 0]$  pattern. Any good



source coder would do that including LZ. This is advantageous to source coding, but in this case it means missing a very obvious atypical pattern.

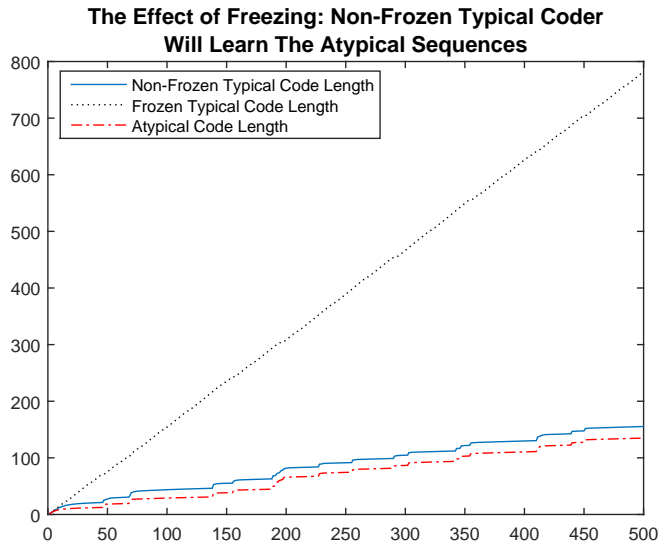


Fig. 5. The importance of freezing source coding when testing for atypicality.

If a very large amount of data is used for training, the complexity can become very high, mainly in terms of memory. Namely, all contexts might be observed, and the context tree completely filled out. For example, suppose the typical data is actually iid. That means that every string  $x_1, \dots, x_N$  is seen with equal probability. For the CTW algorithm that means that every node of the context tree will be filled out, and the number of nodes with depth  $D$  is  $2^{D+1}$ . For dictionary based algorithms, it means that the dictionary size becomes huge. What is needed is some algorithm that not only estimates the unknown parameters, but also the model (e.g., iid). One way could be to trim the context tree (or dictionary), but we have not looked into this in detail.

#### D. Atypical subsequences

For finding atypical subsequences of a long sequences, the same basic setup as in the previous sections can be used. Let  $\mathcal{X}(n, l) = (x_n, x_{n+1}, \dots, x_{n+l-1})$  be a subsequence of  $\{x_n, n = 0, \dots, \infty\}$  that we want to test for atypicality. As in Section III-B the start of a sequence needs to be encoded as well as the length. Additionally the code length is minimized over the maximum depth  $D$  of the context tree. The atypical code length is then given by

$$L_A(\mathcal{X}(n, l)) = \min_D (-\log P_w^\lambda(D) + \log^* D) + \log^* l$$

except for the  $\tau$ . Here  $P_w^\lambda(D)$  denotes the probability at the root of the context tree [43] of depth  $D$ . Since we are also interested in finding short sequences, how the encoding is initialized is of importance, and for atypical coding we therefore use the algorithm in Section II of [58].

For typical coding we use either a known fixed model and Shannon codes, or the algorithm in Section IV-C when the model is not known; when we encode  $\mathcal{X}(n, l)$  we use  $x_{n-D}, \dots, x_{n-1}$  as context for  $x_n$  (we can assume  $n > D$ ). Equivalently, we can encode the total sequence  $\{x_n, n = 0, \dots, \infty\}$  (with the algorithm from Section IV-C); let  $L(n)$  be the codelength for the sequence  $x_0, \dots, x_n$ . Then we can put  $L_T(\mathcal{X}(n, l)) = L(n+l-1) - L(n)$ .

We need to test every subsequence of every length, that is, we need to test subsequences  $\mathcal{X}(n, l)$  for every value of  $n$  and  $l$ . For atypical coding this means that a new CTW algorithm needs to be started at every sample time. So, if the maximum sequence length is  $L$ ,  $L$  separate CTW trees need to be maintained at any time. These are completely independent, so they can be run on parallel processors.

The result is that for every bit of the data we calculate

$$\Delta L(n) = L_A(\mathcal{X}(n, l)) - \min_l L_T(\mathcal{X}(n, l)) \quad (35)$$

and we can state the atypicality criterion as  $\Delta L(n) < -\tau$ . The advantage of stating it like this is that we do not need to choose  $\tau$  prior to running the algorithm. We can sort according to  $\Delta L(n)$ , and first examine the data with smallest  $\Delta L(n)$ , which should be the *most* atypical parts of data. Thus, the algorithm is really parameter free, which is one advantage of our approach. In many anomaly detection algorithms, there are multiple parameters that need to be adjusted.

This implementation clearly is quite complex, but still feasible to implement for medium sized data sets due to the speed of the source coder. In order to process larger data sets, a faster approximate search algorithm is needed, and we are working on such algorithms, but leave that as a topic of later papers.

## V. EXPERIMENTAL RESULTS

In order to verify the performance of our algorithm, we used three different experiments. In the first we evaluated randomness of sources, in accordance with our starting point of Kolmogorov-Martin-Löf randomness. In the second, we looked for infection in human DNA, and in the third we looked for arrhythmia in ECG.

A word about presentation of the results. For the outcome of our method we plot  $\Delta L(n)$  given by (35). At the same time, we would like to illustrate the raw data. The source in all cases is a stream of bits  $x[n] \in \{0, 1\}$ . We convert this to  $y[n] \in \{-1, 1\}$  (i.e.  $y[n] = (-1)^{x[n]-1}$ ), and then plot  $S[N] = \sum_{n=1}^N y[n]$ ; we call this the random walk representation. In our experience, this allows one to quickly assess if there is any obvious pattern in data. If the data is random, the results will look like a typical random walk: both small fluctuations and large fluctuations.

All experimental data and software used is available at <http://itdata.hostmadsen.com>.

### A. Coin Tosses

In this experiment the typical data is iid binary random. As source of typical data we used experimental coin tosses from [59]. This data consists of 40,000 tosses by two Berkeley undergraduates of a fair coin and the result has 20,217 heads ( $X_i = 1$ ), so  $\Pr\{X_i = 1\} = 0.505425 \approx \frac{1}{2}$ . Therefore we can consider it as a real binary IID experiment, indeed it is an example of pure random data. In our experiments with this data, we examine the randomness of other types of data.

One type of data one might *think* is purely random are word length changes in a text. In the first experiment, we generate binary data using consecutive word length comparison of part of M. B. Synge’s “On the shores of the great sea” in the following manner: If the next word is longer than the current word, 1 is assigned to the binary data, otherwise 0. In the case of two consecutive words with same length, a random 0 or 1 is generated (with a *good* random number generator). We then insert this data in the coin toss data. Since we assume coin tosses data is IID, there is no need to train the CTW and the the code length of the IID case (2) can be used for typical coding. Fig. 6 illustrates the result of the algorithm on the mixed data. Thus, word length changes are not iid random. Perhaps this is because word length is bounded from above and below, so that there are limits to how long runs of 0s or 1s are possible.

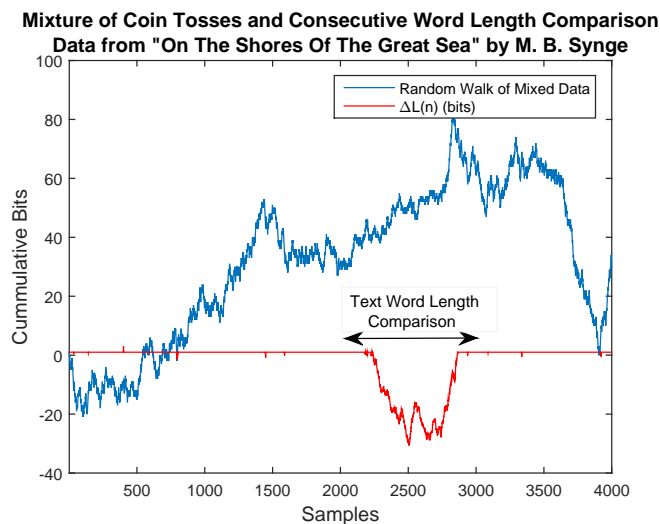


Fig. 6. Random walk of mixed coin tosses and consecutive word comparison.

In second experiment, we generated random data with the infamous RANDU random number generator. This was a random number generator that was widely used until it was discovered that it has some clear deviation from randomness. RANDU generates random numbers in the interval  $[0, 2^{31} - 1]$ , so each number needs 31 bits for binary representation. But instead of using 31 bits for each number, we sum up all the 31 bits and compare it with 15.5 to generate either 0 or 1. Then this data is inserted in the part of coin tosses data. Fig. 7 shows that the most atypical segment is where we have inserted data from RANDU random number generator.

In the third experiment, we generate binary data using consecutive heart rate comparison of part of normal sinus rhythm downloaded from MIT- BIH database [60] in the same way as for the text. Fig. 7 represents the result of the algorithm on the mixed data. As can be seen the atypicality measure shows a huge difference between iid randomness and randomness of consecutive heart beats. We don’t know why this is the case.

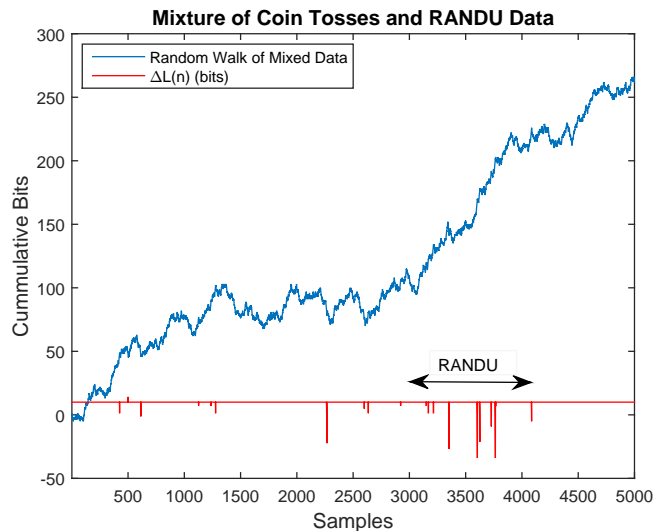


Fig. 7. Random walk of mixed coin tosses and RANDU.

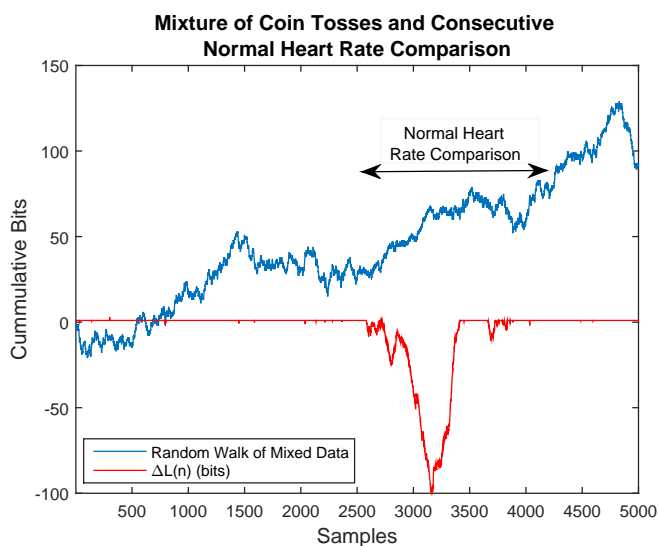


Fig. 8. Random walk of mixed coin tosses and HRV.

## B. DNA

In this collection of experiments, we detect viral and bacterial insertion into human genomic DNA. DNA from foreign species can be inserted into the human genome either through natural processes [61], typically through viral infections, bacterial infections, or through genetic engineering [62]. The inverse also occurs in genetic engineering experiments during the creation of “transgenic” organisms, with the insertion of human DNA into bacteria, yeast, worms, or mice. In the experiments we show here, we have focused on the former case. We train the CTW algorithm

on pure human genomic DNA.

The data that we have used was comprised of ~20 kilobases of human genomic DNA (each sequence from a different chromosome) with either bacterial or viral random DNA sequences (~2 kilobases per insertion) inserted. Since our software is too slow to find atypical sequences of length more than a few hundreds, we removed the middle of the insertions. Notice that this actually makes detection harder. We used some of the human DNA for training, but not the same as the test sequences.

In the first experiment we tried to detect short sequences from *Streptococcus Pneumoniae* (a bacterial infection with a high fatality rate, and a frequent cause of death in the elderly) randomly inserted into larger segments of human genomic DNA. Fig. 9 illustrates the result of the experiment. Based on the figure, the inserted *Streptococcus Pneumoniae* DNA fragment was detected by our algorithms.

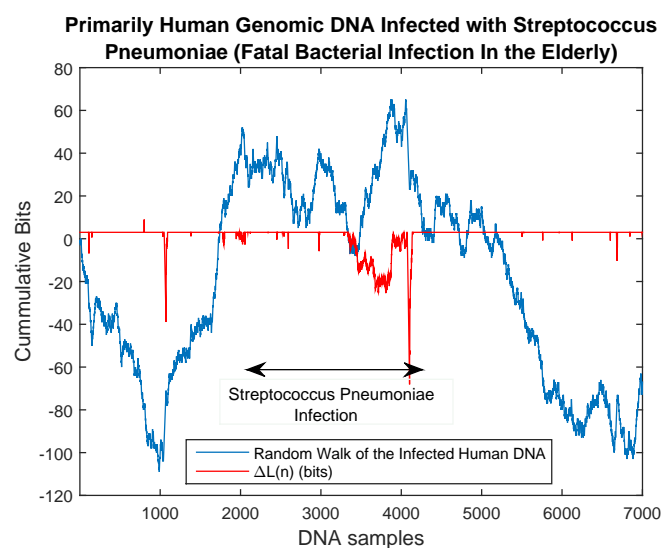


Fig. 9. Random walk of human DNA with bacterial infection.

In the second experiment we tried to detect HIV inserted into human genomic DNA to mimic viral infection, which is a more realistic experiment since viruses typically insert their DNA into the host genome every time a human obtains a viral infection. Fig. 10 illustrates the result of the experiment. As can be seen, the infected viral fragment was detected by our algorithms.

### C. HRV

While HRV (Heart Rate Variability) can be a powerful indicator for arrhythmia [8], the common issue is that it is not known exactly what to look for in the data. Our aim for this application is to use atypicality to localize signs of subtle and complex arrhythmia. In [63] based on our modest goal of localizing a simple type of known arrhythmia, we managed to find premature beats using HRV signal, but here we attempted to detect more subtle arrhythmia. The HRV signals that we used were downloaded from MIT- BIH database [60]. We used “MIT-BIH normal sinus

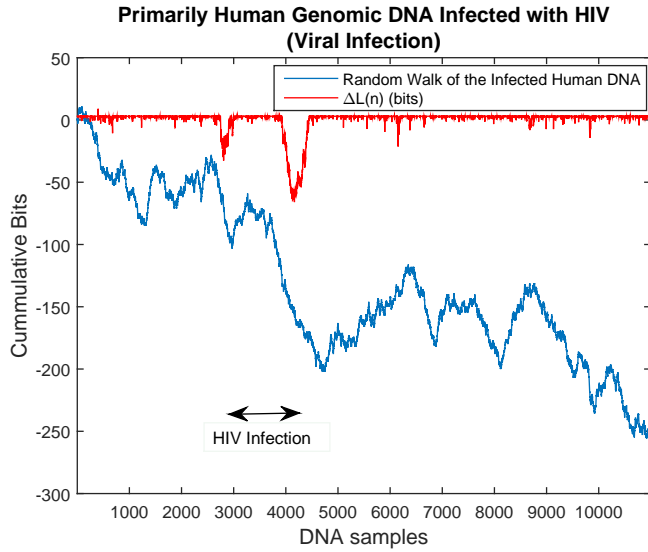


Fig. 10. Random walk of human DNA with viral infection.

rhythm database (nsrdb)” and “MIT-BIH supraventricular arrhythmia database (svdb)”. Encoding of HRV signals were done by same manner as the text word length comparison of subsection V-A. In this experiment, CTW was trained with HRV of normal sinus rhythm, then applied to a HRV signal that has supraventricular rhythms. Fig. 11 shows the result of the simulation. The algorithm was able to localize the segment that suffers from abnormal rhythms.

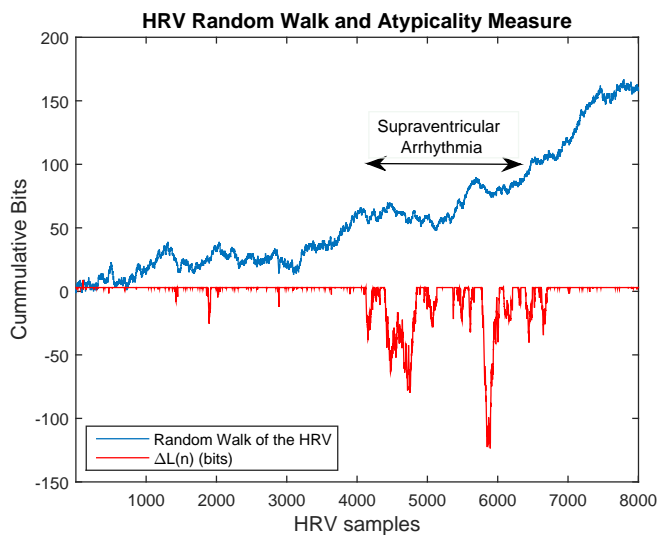


Fig. 11. Random walk of HRV.

## VI. CONCLUSION

In this paper we have developed a criterion for finding atypical (sub)sequences in large datasets. The criterion is based on a solid theoretical foundation, and we have shown that the criterion is amenable to theoretical analysis. In particular, we have shown that the probability that a sample is intrinsically atypical is less than 1, an important theoretical requirement that is not trivially satisfied.

We have also shows in a few examples with real-world data that the method is able to find known atypical subsequences. Our aim set out in the introduction was more ambitiously to find 'interesting' data in large datasets. In that context, the purpose of the current paper is only to introduce the methodology, and show some theoretical properties. In order to analyze really big datasets, we need much faster (probably approximate) algorithms, much more efficient software (say in Python or C), and faster computers. We are working on that as future work.

## REFERENCES

- [1] L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos, "Fast and reliable anomaly detection in categorical data," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 415–424.
- [2] K. Smets and J. Vreeken, *The Odd One Out: Identifying and Characterising Anomalies*, 2011, ch. 69.
- [3] W. Liu, I. Park, and J. Principe, "An information theoretic approach of designing sparse kernel adaptive filters," *Neural Networks, IEEE Transactions on*, vol. 20, no. 12, pp. 1950–1961, Dec 2009.
- [4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 5, pp. 823–839, may 2012.
- [5] D. Rumsfeld, *Known and Unknown: A Memoir*. Penguin, 2011.
- [6] K. Hamano and H. Yamamoto, "A randomness test based on t-codes," in *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, dec. 2008, pp. 1–6.
- [7] A. Nies, *Computability and Randomness*. Oxford University Press, 2009.
- [8] M. Malik, "Heart rate variability," *Annals of Noninvasive Electrocardiology*, vol. 1, no. 2, pp. 151–181, April 1996.
- [9] M. F. Hilton, R. A. Bates, K. R. Godfrey, and et al., "Evaluation of frequency and time-frequency spectral analysis of heart rate variability as a diagnostic marker of the sleep apnea syndrome," *Med. Biol. Eng. Comput.*, vol. 37, no. 6, pp. 760–769, November 1999.
- [10] N. V. Thakor and Y. S. Zhu, "Application of adaptive filtering to ECG analysis: Noise cancellation and arrhythmia detection," *IEEE Trans. on Biomed. Eng.*, vol. 38, pp. 785–794, August 1991.
- [11] J. F. Thayer, S. S. Yamamoto, and J. F. Brosschot, "The relationship of autonomic imbalance, heart rate variability and cardiovascular disease risk factors," *International Journal of Cardiology*, vol. 141, no. 2, pp. 122–131, May 2009.
- [12] J. Fondon and H. Garner, "Probing human cardiovascular congenital disease using transgenic mouse models," *Proc Natl Acad Sci U S A*, vol. 101, no. 52, pp. 18058–63, 2004.
- [13] D. K. Mellinger, K. M. Stafford, S. E. Moore, R. P. Dziak, and H. Matsumoto, "An overview of fixed passive acoustic observation methods for cetaceans," *Oceanography*, vol. 20, no. 4, pp. 36–45, 2007.
- [14] M. Thottan and C. Ji, "Anomaly detection in ip networks," *Signal Processing, IEEE Transactions on*, vol. 51, no. 8, pp. 2191–2204, aug. 2003.
- [15] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed. Springer, 2008.
- [16] T. Cover and J. Thomas, *Information Theory, 2nd Edition*. John Wiley, 2006.
- [17] J. Rissanen, "Complexity of strings in the class of markov sources," *Information Theory, IEEE Transactions on*, vol. 32, no. 4, pp. 526–532, jul 1986.
- [18] —, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, no. 2, pp. 416–431, 1983.
- [19] —, "Universal coding, information, prediction, and estimation," *Information Theory, IEEE Transactions on*, vol. 30, no. 4, pp. 629–636, jul 1984.

- [20] —, “Stochastic complexity and modeling,” *The Annals of Statistics*, no. 3, pp. 1080–1100, Sep. 1986.
- [21] S. Evans, B. Barnett, S. Bush, and G. Saulnier, “Minimum description length principles for detection and classification of ftp exploits,” in *Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE*, vol. 1, oct.-3 nov. 2004, pp. 473 – 479 Vol. 1.
- [22] N. Wang, J. Han, and J. Fang, “An anomaly detection algorithm based on lossless compression,” in *Networking, Architecture and Storage (NAS), 2012 IEEE 7th International Conference on*, 2012, pp. 31–38.
- [23] W. Lee and D. Xiang, “Information-theoretic measures for anomaly detection,” in *Security and Privacy, 2001. S P 2001. Proceedings. 2001 IEEE Symposium on*, 2001, pp. 130–143.
- [24] I. Paschalidis and G. Smaragdakis, “Spatio-temporal network anomaly detection by assessing deviations of empirical measures,” *Networking, IEEE/ACM Transactions on*, vol. 17, no. 3, pp. 685–697, 2009.
- [25] C.-K. Han and H.-K. Choi, “Effective discovery of attacks using entropy of packet dynamics,” *Network, IEEE*, vol. 23, no. 5, pp. 4–12, 2009.
- [26] P. Baliga and T. Lin, “Kolmogorov complexity based automata modeling for intrusion detection,” in *Granular Computing, 2005 IEEE International Conference on*, vol. 2, 2005, pp. 387–392 Vol. 2.
- [27] H. Shahriar and M. Zulkernine, “Information-theoretic detection of sql injection attacks,” in *High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on*, 2012, pp. 40–47.
- [28] Y. Xiang, K. Li, and W. Zhou, “Low-rate ddos attacks detection and traceback by using new information metrics,” *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 2, pp. 426–437, 2011.
- [29] F. Pan and W. Wang, “Anomaly detection based-on the regularity of normal behaviors,” in *Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006. 1st International Symposium on*, 2006, pp. 6 pp.–1046.
- [30] E. Eiland and L. Liebrock, “An application of information theory to intrusion detection,” in *Information Assurance, 2006. IWIA 2006. Fourth IEEE International Workshop on*, 2006, pp. 16 pp.–134.
- [31] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, “The similarity metric,” *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 3250 – 3264, dec. 2004.
- [32] E. Keogh, S. Lonardi, and C. A. Ratanamahatana, “Towards parameter-free data mining,” in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 206–215.
- [33] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice-Hall, 1993.
- [34] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley, 1990.
- [35] G. Shamir, “On the mdl principle for i.i.d. sources with large alphabets,” *Information Theory, IEEE Transactions on*, vol. 52, no. 5, pp. 1939–1955, May 2006.
- [36] P. Elias, “Universal codeword sets and representations of the integers,” *Information Theory, IEEE Transactions on*, vol. 21, no. 2, pp. 194 – 203, mar 1975.
- [37] H. V. Poor, *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1994.
- [38] W. Hoeffding, “Probability inequalities for sums of bounded random variables,” *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.
- [39] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. Springer, 1998.
- [40] E. L. Lehmann, *Testing Statistical Hypotheses*. Springer, 2005.
- [41] G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes, Third Edition*. Oxford University Press, 2001.
- [42] J. Rissanen, “Modeling by shortest data description,” *Automatica*, pp. 465–471, 1978.
- [43] F. M. J. Willems, Y. Shtarkov, and T. Tjalkens, “The context-tree weighting method: basic properties,” *Information Theory, IEEE Transactions on*, vol. 41, no. 3, pp. 653–664, 1995.
- [44] J. Ziv and A. Lempel, “A universal algorithm for sequential data compression,” *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 337 – 343, may 1977.
- [45] —, “Compression of individual sequences via variable-rate coding,” *Information Theory, IEEE Transactions on*, vol. 24, no. 5, pp. 530 – 536, sep 1978.
- [46] M. Effros, K. Visweswariah, S. Kulkarni, and S. Verdu, “Universal lossless source coding with the burrows wheeler transform,” *Information Theory, IEEE Transactions on*, vol. 48, no. 5, pp. 1061 –1081, may 2002.



- [47] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *Communications, IEEE Transactions on*, vol. 32, no. 4, pp. 396 – 402, apr 1984.
- [48] A. Moffat, "Implementing the ppm data compression scheme," *Communications, IEEE Transactions on*, vol. 38, no. 11, pp. 1917 –1921, nov 1990.
- [49] M. Titchener, "Deterministic computation of complexity, information and entropy," in *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, aug 1998, p. 326.
- [50] K. Kawaharada, K. Ohzeki, and U. Speidel, "Information and entropy measurements on video sequences," in *Information, Communications and Signal Processing, 2005 Fifth International Conference on*, 0-0 2005, pp. 1150 –1154.
- [51] U. Speidel, "A note on the estimation of string complexity for short strings," in *Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on*, dec. 2009, pp. 1 –5.
- [52] U. Speidel, R. Eimann, and N. Brownlee, "Detecting network events via t-entropy," in *Information, Communications Signal Processing, 2007 6th International Conference on*, dec. 2007, pp. 1 –5.
- [53] U. Speidel and T. Gulliver, "An analytic upper bound on t-complexity," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, july 2012, pp. 2706 –2710.
- [54] J. Yang and U. Speidel, "String parsing-based similarity detection," in *Information Theory Workshop, 2005 IEEE*, aug.-1 sept. 2005, p. 5 pp.
- [55] P. A. J. Volf and F. M. J. Willems, "Switching between two universal source coding algorithms," in *Data Compression Conference, 1998. DCC '98. Proceedings*, 1998, pp. 491–500.
- [56] P. Jacquet and W. Szpankowski, "Limiting distribution of lempel ziv'78 redundancy," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, 31 2011-aug. 5 2011, pp. 1509 –1513.
- [57] F. Willems, Y. Shtarkov, and T. Tjalkens, "Reflections on "the context tree weighting method: Basic properties"," *Newsletter of the IEEE Information Theory Society*, vol. 47, no. 1, 1997.
- [58] F. Willems, "The context-tree weighting method: extensions," *Information Theory, IEEE Transactions on*, vol. 44, no. 2, pp. 792–798, Mar 1998.
- [59] [Online]. Available: <http://www.stat.berkeley.edu>
- [60] [Online]. Available: <http://physionet.org/cgi-bin/atm/ATM>
- [61] R. J. Britten, "DNA sequence insertion and evolutionary variation in gene regulation," *Proceedings of the National Academy of Sciences*, vol. 93, no. 18, pp. 9374–9377, 1996. [Online]. Available: <http://www.pnas.org/content/93/18/9374.abstract>
- [62] J. Z. K. Khattak, S. Rauf, Z. Anwar, H. M. Wahedi, and T. Jamil, "Recent advances in genetic engineering - a review," *Current Research Journal of Biological Sciences*, vol. 4, no. 1, 2012.
- [63] A. Høst-Madsen, E. Sabeti, and C. Walton, "Information theory for atypical sequences," in *IEEE Information Theory Workshop (ITW'13), Seville, Spain*, 2013.