INSTITUTE OF PHYSICS PUBLISHING EUROPEAN JOURNAL OF PHYSICS

Eur. J. Phys. **26** (2005) S69–S77 doi:10.1088/0143-0807/26/5/S08

Measuring complexity with zippers

Andrea Baronchelli[1], Emanuele Caglioti[2] and Vittorio Loreto[1]

[1] Physics Department and INFM-SMC, La Sapienza University, Ple A Moro 2, 00185 Rome, Italy
[2] Mathematics Department, La Sapienza University, Ple A Moro 2, 00185 Rome, Italy

E-mail: andrea.baronchelli@roma1.infn.it

Received 23 May 2005, in final form 14 June 2005
Published 8 July 2005
Online at stacks.iop.org/EJP/26/S69

Abstract

Physics concepts have often been borrowed and independently developed by other fields of science. In this perspective, a significant example is that of the entropy in information theory. The aim of this paper is to provide a short and pedagogical introduction to the use of data compression techniques for the estimate of the entropy and other relevant quantities in information theory and algorithmic information theory. We consider in particular the LZ77 algorithm as a case study and discuss how a zipper can be used for information extraction.

1. Introduction

Strings of symbols are nowadays widespread in all fields of science. On the one hand, many systems are intrinsically described by sequences of characters: DNA, written texts, bits in the transmission of digital data, magnetic domains in storage data devices, etc. On the other hand, a string of characters is often the only possible description of a natural phenomenon. In many experiments, for example, one is interested in recording the variation in time of a given physical observable (for instance, the temperature of a system), thus obtaining a time series, which, when suitably codified, results in a sequence of symbols.

Given a string of symbols, the main problem consists in quantifying and then extracting the information it contains. This acquires different meanings in different contexts. For a DNA string, for instance, one could be interested in separating the portions of coding for proteins from the non-coding parts. In contrast, in a written text important information is the language in which it is written, its author, the subject treated etc.

Information theory (IT) is the branch of science which deals, among other things, with the problems we have mentioned. In a seminal paper dated 1948, Claude Shannon pointed out the possibility of quantifying the information contained in a (infinite) string of characters [1]. Adopting a probabilistic approach, i.e., focusing the attention on the source generating a string, the famous Shannon–McMillan theorem shows that there is a limit to the possibility of

compressing a string without losing the information it brings. This limit is proportional to the *entropy* (or informatic content) of that string [1, 2].

A remark is interesting now. The name entropy is not accidental, and information theory represents one of the best examples of a concept developed in physics whose role became of primary importance also in other fields. Historically, the concept of entropy was initially introduced in thermodynamics in a phenomenological context. Later, mainly by the contribution of Boltzmann, a probabilistic interpretation of the entropy was developed in order to clarify its deep relation with the microscopic structure underlying the macroscopic bodies [3]. More recently, Shannon, generalizing the concept of entropy in the apparently unrelated field of communication systems, was able to establish a self-consistent information theory. For a recent excursus about the notion of entropy see [4]. We shall describe more precisely Shannon's approach in the following section, but we refer the interested reader to [5] for a discussion of the connections between the Shannon and microscopic entropies.

A radically different approach to the information problem, namely the algorithmic information theory (AIT) [6–9], was developed in the 1960s. It showed again, from a different point of view, that a good way of quantifying the information embedded in a string is that of trying to describe it in the shortest possible way.

In this framework, it seems natural to look at those algorithms as expressly conceived to compress a file (i.e., a string of bytes), known as *zippers*. A zipper takes a file and tries to minimize its length. However, as we have mentioned, there is a theoretical limit, represented by the entropy of the considered sequence, to the performance of a zipper. A compression algorithm able to reach this theoretical limit is said to be 'optimal'. Thus an optimal zipper can be seen as an ideal tool to estimate the informatic content of a string, i.e. to quantify the information it brings. In this paper, we shall discuss this possible application of data compression algorithms together with its shortcomings.

Finally, besides the important scientific problem of measuring how much information is contained in a string, one could ask if it is possible to extract that information. With a slight abuse of the word, we can address the level of the kind of information contained in a sequence as the semantic level. We are then interested in asking whether it is possible to access the semantic level from a information theoretical, 'syntactic', analysis of a string. We shall show that, under certain assumptions, this is indeed the case in many different circumstances.

The outline of this paper is as follows. In section 2 we make a short introduction to some information theory concepts; in section 3 we describe the optimal compression algorithm LZ77; in section 4, finally, we illustrate with some examples the possible applications of information extraction techniques.

2. Entropy and complexity

In Shannon's probabilistic approach to information, developed in an engineering context, the communication scheme is fundamental. A message is first produced by a source of information, then is codified in a way proper for the transmission in a channel and finally, before arriving at the receiver, it must be brought back to the original form.

All these steps are of great theoretical interest, but for our purposes we will concentrate on the source uniquely. This is a device able to form a message adding one symbol per unit time, chosen in agreement with some probabilistic rules, to the previously emitted ones. Here we consider only the cases in which the possible characters are finite in number, i.e., the alphabet $\mathcal{X}$ is finite. The source can then be identified with the stochastic process it obeys. Shannon's IT always deals with ergodic sources. A rigorous definition of ergodic processes is out of the scope of this paper. We shall limit ourselves to an intuitive definition. A source is ergodic

if it is stationary (the probability rules of the source do not vary in time) and the following property holds. If N_l is the number of occurrences of a generic sequence $Y = y_1, \ldots, y_s$ in a string X of length $l > s$, then

$$\lim_{l\to\infty} P\left\{\left|\frac{N_l}{l} - P\left(x_{i_1}, \ldots, x_{i_s} = y_1, \ldots, y_s\right)\right| < \epsilon\right\} = 1 \qquad \forall \epsilon, y_s, \tag{1}$$

i.e., the averages made over an emitted string, $\frac{N_l}{l}$, coincide with those made over time $P\left(x_{i_1}, \ldots, x_{x_s} = y_1, \ldots, y_s\right)$, in the limit of infinite string length.

Now, if $\mathbf{x}$ is an n-symbols sequence chosen from the $\mathcal{X}^n$ possible sequences of that length, we introduce the N-block entropy as

$$H_n = H(X_1, X_2, \ldots, X_n) = -\sum_{\mathbf{x}\in\mathcal{X}^n} p(\mathbf{x}) \log p(\mathbf{x}) \tag{2}$$

where $p(\mathbf{x})$ is the probability of the string $\mathbf{x}$ to be emitted. The differential entropy $h_n = H_{n+1} - H_n$ represents the average information carried by the $n + 1$ symbol when the n previously emitted characters are known. Noting that the knowledge of a longer past history cannot increase the uncertainty on the next symbol, we have that h_n cannot increase with n, i.e., $h_{n+1} \leqslant h_n$ and for an ergodic source we define the Shannon entropy h as

$$h = \lim_{n\to\infty} h_n = \lim_{n\to\infty} \frac{H_n}{n}. \tag{3}$$

The entropy of a source is a measure of the information it produces. In other words h can be viewed as a measure of the surprise we have while analysing a string generated by a stochastic process. Consider, for example, the case of a source emitting a unique symbol A with probability 1. For that source $h = 0$, and in fact we would have no surprise observing a new A. On the other hand, if the probability of occurrence of the symbol A is quite small our surprise will be proportionally large. In particular, it turns out to be proportional to the absolute value of the logarithm of its probability. Then h is precisely the average surprise obtained by the stochastic process. Remarkably, it can be shown that h, apart from multiplicative coefficients, is the only quantity that measures the surprise generated by a stochastic process [2].

More precisely, the role of h as an information measure can be fully recognized in the Shannon–McMillan theorem [1, 2]. Given an N characters-long message emitted by an ergodic source, it states that

(i) there exists a coding for which the probability of the message to require more than $Nh_2 = (Nh/\log 2)$ bits tends to zero when N tends to infinity;
(ii) there does not exist a coding for which the probability of the message to require less than Nh_2 bits tends to one when N tends to infinity.

A completely different approach to information-related problems is that of algorithmic information theory [6–9]. In this context, the focus is on the single sequence, rather than on its source, and the basic concept is the algorithmic complexity: the entropy of a string of characters is the length (in bits) of the smallest program which produces as output the string and stops afterwards. This definition is abstract. In particular it is impossible, even in principle, to find such a program and as a consequence the algorithmic complexity is a non-computable quantity. This impossibility is related to the halting problem and to Godel's theorem [10]. Nevertheless, this second approach also indicates that searching for the most concise description of a sequence is the way to estimate the amount of information it contains. As one could expect, in fact, there is a connection between the algorithmic complexity of a string and the entropy of its source, but we refer the interested reader to [10] for a detailed discussion.

Up to this point our attention has been devoted to the characterization of a single string. Both IT and AIT, however, provide several measures of relations of remoteness, or proximity, between different sequences. Among these, it is interesting to recall the notion of relative entropy (or Kullback–Leibler divergence) [17, 18, 11] which is a measure of the statistical remoteness between two distributions. Its essence can be easily grasped with the following example.

Let us consider two stationary memoryless sources $\mathcal{A}$ and $\mathcal{B}$ emitting sequences of 0 and 1: $\mathcal{A}$ emits 0 with probability p and 1 with probability $1-p$ while $\mathcal{B}$ emits 0 with probability q and 1 with probability $1-q$. The optimal coding for a sequence emitted by $\mathcal{A}$ codifies on average every character with $h(\mathcal{A}) = -p\log_2 p - (1-p)\log_2(1-p)$ bits (the Shannon entropy of the source). This optimal coding will not be the optimal one for the sequence emitted by $\mathcal{B}$. In particular, the entropy per character of the sequence emitted by $\mathcal{B}$ in the coding optimal for $\mathcal{A}$ will be

$$C(\mathcal{A}|\mathcal{B}) = -q\log_2 p - (1-q)\log_2(1-p) \tag{4}$$

while the entropy per character of the sequence emitted by $\mathcal{B}$ in its optimal coding is $-q\log_2 q - (1-q)\log_2(1-q)$. Equation (4) defines the so-called cross entropy per character of $\mathcal{A}$ and $\mathcal{B}$. The number of bits per character wasted to encode the sequence emitted by $\mathcal{B}$ with the coding optimal for $\mathcal{A}$ is the relative entropy of $\mathcal{A}$ and $\mathcal{B}$:

$$d(\mathcal{A}||\mathcal{B}) = C(\mathcal{A}|\mathcal{B}) - h(\mathcal{A}) = -q\log_2\frac{p}{q} - (1-q)\log_2\frac{1-p}{1-q}. \tag{5}$$

A linguistic example will help to clarify the situation: transmitting an Italian text with a Morse code optimized for English will result in the need of transmitting an extra number of bits with respect to another coding optimized for Italian; the difference is a measure of the relative entropy between, in this case, Italian and English (supposing the two texts are archetypal representations of their respective languages, which is not the case).

It is important to remark that the relative and cross entropies are not distances (metric) in the mathematical sense, since they are not symmetric and do not satisfy in general the triangular inequality. Defining a true distance between strings is an important issue both for theoretical and practical reasons (see for some recent approaches [12–14] and for a short review [21]).

3. Zippers

In the previous section, we have seen two different approaches to the characterization of the information, the classical and the algorithmic ITs. We have also seen that, despite their profound differences, both of them indicate that the way to quantify the information of a string is to find its shortest description, i.e., to compress it. Driven by this fact, in this paragraph we shall illustrate the LZ77 compression algorithm, that, asymptotically, is able to get to the Shannon limit.

The Lempel and Ziv algorithm LZ77 [15] (see figure 1) (used, for instance, by *gzip* and *zip* commercial zippers) achieves compression exploiting the presence of duplicated strings in the input data. The second occurrence of a string is replaced by a pointer to the previous string given by two numbers: a distance, representing how far back into the window the sequence starts, and a length, representing the number of characters for which the sequence is identical. More specifically, the zipper reads sequentially the input N-symbols sequence, $x = x_1, \ldots, x_N$. When n symbols have already been analysed, LZ77 finds the longest string starting at symbol $n+1$ which has already been encountered in the previous n characters.

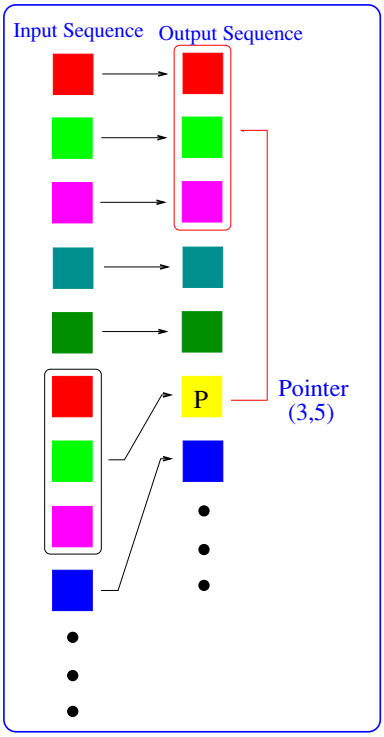

Figure 1. The scheme of the LZ77 algorithm: the LZ77 algorithm searches in the look-ahead buffer for the longest substring (in this case a substring of colours) already occurring and replaces it with a pointer represented by two numbers—the length of the matching and its distance.

(This figure is in colour only in the electronic version)

In other words LZ77 looks for the largest integer m such that the string $x_{n+1}, \ldots, x_{n+m}$ already appearing in $x_1, \ldots, x_n$. The string found is then codified with two numbers: its length m and the distance from its previous occurrence. If no already encountered string starts at position n the zipper simply writes the symbol appearing in that position in the compressed sequence and starts a new search from position $n + 1$.

From the above description it is intuitive that LZ77 performs better and better as the number of processed symbols grows. In particular, for infinitely long strings (emitted by ergodic sources), its performance is 'optimal', i.e., the length of the zipped file divided by the length of the original file tends to $h/\ln 2$ [16]. The convergence to this limit, however, is extremely slow. By defining as the code rate the average bits per symbol needed to encode the sequence, we have

$$\text{code rate} \simeq h_2 + \mathcal{O}\left(\frac{\ln \ln N}{\ln N}\right). \tag{6}$$

Notwithstanding its limitations, LZ77 can then be seen as a tool for estimating the entropy of a sequence. However, the knowledge of h_2, though interesting from a theoretical point of view is often scarcely useful in applications. For practical purposes, on the other hand, methods able to make *comparisons* between strings are often required. A very common case, for instance, is that in which one has to classify an unknown sequence with respect to a dataset of known strings; i.e., one has to decide which known string is closer (in some sense) to the unknown string.

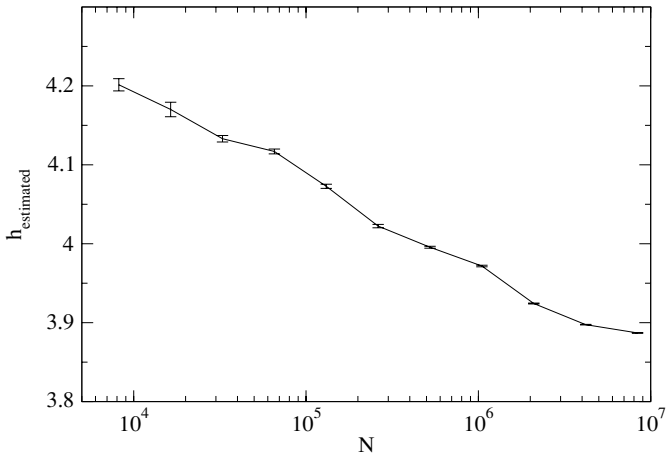

Figure 2. Entropy estimation: the number of bits per character of the zipped sequence $h_{\text{estimated}}$ is plotted versus the length N of the original one. Bernoulli sequences with $K = 10$ symbols are analysed. The zipper performs better with longer strings, but the convergence towards the optimal compression, though theoretically proved, is extremely slow. The Shannon entropy of the considered sequences is $h_2 \simeq 3.32$ and, for strings of approximately 8×10^6 characters, $h_{\text{estimated}}$ is 18% larger than this value.

In section 2 we have introduced the relative entropy and the cross entropy between two sources. Recently, a method has been proposed for the estimate of the cross entropy between two strings based on LZ77 [12]. Recalling that the cross entropy $C(A|B)$ between two strings A and B, is given by the entropy per character of B in the optimal coding for A, the idea is that of appending the two sequences and zipping the resulting file $A + B$. In this way the zipper 'learns' the A file and, on encountering the B subsequence, tries to compress it with a coding optimized for A. If B is not too long [20, 21], thus preventing LZ77 from learning it as well, the cross entropy per character can be estimated as

$$C(\mathcal{A}|\mathcal{B}) \simeq \frac{L_{A+B} - L_A}{L_B} \tag{7}$$

where L_X is the length of the compressed X sequence. This method is strictly related to the Ziv–Merhav algorithm [19] to estimate the relative entropy between two individual sequences.

4. Examples and numerical results

In this section, we illustrate the behaviour of LZ77 in experiments of entropy estimation and of recognition with two examples. Figure 2 reports the LZ77 code rates when zipping Bernoulli sequences of various lengths. A Bernoulli string is generated by extracting randomly one of K allowed symbols with probability $1/K$ ($K = 10$ in our case). The entropy of such strings is simply $\log K$. From the figure it is evident that the zipper performs better and better with longer strings, though, as seen in (6), the convergence is extremely slow. It is important to remark on how there exist more efficient algorithms to estimate the entropy of a string. We refer the interested reader to [22] for a recent review. It is nevertheless useful to quote the so-called Shannon game to estimate the entropy of English (see [23] for an applet) where the identification of the entropy with a measure of surprise is particularly clear.

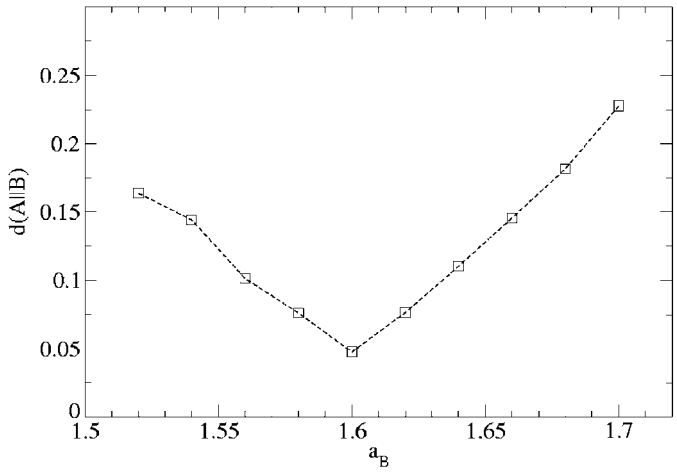

Figure 3. A recognition experiment: the relative entropy, estimated by means of LZ77 as discussed in the text, between an unknown sequence A and a set of known strings B allows us to identify the source of A. All the sequences are generated by a Lozi map, and the problem is to identify the parameter $a_A = 1.6$ of the source of A. The minimum of the relative entropy clearly allows one to identify this parameter indicating that A is closer to the string B generated with $a_B = a_A$ than to any strings generated with different values of a.

In figure 3 an experiment of recognition is reported [20]. Here an unknown sequence is compared, in the sense discussed in the previous section, with a number of known strings. The idea is to test that the unknown sequence was emitted by the same source of the closer known one. The source here is a Lozi map, i.e., a dynamical system of the form

$$\begin{cases} x_{n+1} = 1 - a|x_n| + y_n \\ y_{n+1} = bx_n \end{cases}$$

where a and b are parameters. The sequence of symbols used in the following test is obtained taking 0 when $x \leqslant 0$ and 1 when $x > 0$. For $b = 0.5$, numerical studies show that the Lozi map is chaotic for a in the interval $(1.51, 1.7)$. For a discussion of the Lozi map, computation of Lyapunov exponents and representation of its symbolic dynamics in terms of Markov chains, see [24].

Figure 3 reports the result of this test. A Lozi map with $a = 1.6, b = 0.5$ and the initial condition $x = 0.1, y = 0.1$ has been used to generate the sequence A, of length 10 000, that will be used as an unknown sequence. As probing sequences, we have generated a set of sequences, B of length 1000, obtained with Lozi maps with the parameters $b = 0.5$ and a_B varying between 1.52 and 1.7. The quantity computed and reported in the graph is an estimate of the Kullback–Leibler entropy $\mathrm{d}(B||A) = C(A|B) - C(B^*|B)$, where $C(B^*|B)$ is the estimate, in the framework of our scheme, of the entropy rate of B and B^* is another set of sequences of length 10 000. As is evident in our experiment, the closer sequence to the unknown one is the one with $a = 1.6$ and this means that the recognition experiment was successful.

5. Conclusions

The possibility of quantifying information contained in a string of symbols has been one of the great advancements in science of the last 60 years. Both Shannon's and the algorithmic

approaches indicate that finding a synthetic description of a string is a way to determine how much information it stores. It is then natural to focus on those algorithms conceived expressly to compress a string, also known as zippers. In this paper, we have introduced some fundamental concepts of information theory and we have described the LZ77 compressor. This zipper has the property of being asymptotically optimal, thus being also a potential tool for estimating the entropy of a string. More interestingly, we have discussed the possibility of using LZ77 for the estimation of quantities such as the cross or relative entropy which measure the remoteness between different strings. Finally, we have shown a simple example of entropy estimation for a Bernoulli sequence and a successful experiment of recognition between strings emitted by a Lozi map with different parameters.

Acknowledgments

We thank Valentina Alfi, Dario Benedetto, Andrea Puglisi and Angelo Vulpiani for many interesting discussions and contributions to this work. VL acknowledges the partial support of the ECAgents project funded by the Future and Emerging Technologies programme (IST-FET) of the European Commission under the EU RD contract IST-1940. EC acknowledges the partial support of the European Commission through its 6th Framework Programme 'Structuring the European Research Area' and the contract no RITA-CT-2004-505493 for the provision of Transnational Access implemented as Specific Support Action.

Appendix

We report here an example of implementation of LZ77. It must be intended as a didactic illustration, since actual implementations of the algorithm contain several optimizations.

- Build a vector V whose j^{th} component $V[j]$ is the j^{th} symbol of string S that must be compressed;
- Build a vector I whose j^{th} component, $I[j]$, is the position of the closest previous occurrence of symbol v appearing in $V[j]$, or 0 if symbol v has never appeared before;
- Build an empty vector C which will contain the processed V (i.e. the processed S);
- define $i = 0$;
  while $(i < |V|)$ do:

  define $p = I[i]$, $lmax = 1$, $pmax = 0$;
  while $(p \neq 0)$ do:
  define $l = 1$;
  while $(V[i + l] = V[p + l]$ and $(p + l) < i)$ do:
  $l = l + 1$;
  if $l > lmax$ do $lmax = l$, $pmax = p$;
  $p = I[p]$;
  if $(l > 1)$ append to vector C the token $(lmax, i - pmax)$;
  else, if $(l = 1)$, append to vector C the token $(0, 0, V[i])$;
  $i = i + l$;

Before concluding we mention two of the most common adjoining features of the LZ77 algorithm. The first aims to codify better the length–distance token. This is often achieved by zipping further the compressed string exploiting its statistical properties with the Huffman

algorithm [23]. The second feature is due to the necessity of speeding up the zipping process in commercial zippers. It consists in preventing LZ77 from looking back for more than a certain number w of symbols. Such a modified zipper is said to have a 'w-long sliding window'.

References

[1] Shannon C E 1948 *Bell Syst. Tech. J.* **27** 623
[2] Khinchin A I 1957 *Mathematical Foundations of Information Theory* (New York: Dover)
[3] Callen H B 1985 *Thermodynamics and an Introduction to Thermostatistics* 2nd edn (New York: Wiley)
[4] Falcioni M, Loreto V and Vulpiani A 2003 *The Kolmogorov Legacy in Physics (Lecture Note in Physics* vol 636*)* ed R Livi and A Vulpiani (Berlin: Springer) see also G Parisi's contribution in the same volume
[5] Parisi G 1988 *Statistical Field Theory* (New York: Addison-Wesley)
[6] Chaitin G J 1966 *J. Assoc. Comput. Mach.* **13** 547
[7] Chaitin G J 2002 *Information, Randomness and Incompleteness* 2nd edn (Singapore: World Scientific)
[8] Kolmogorov A N 1965 *Probl. Inf. Transm.* **1** 1
[9] Solomonov R J 1964 *Inf. Control* **7** 1
Solomonov R J 1964 *Inf. Control* **7** 224
[10] Li M and Vitányi P M B 1997 *An Introduction to Kolmogorov Complexity and its Applications* 2nd edn (Berlin: Springer)
[11] Cover T and Thomas J 1991 *Elements of Information Theory* (New York: Wiley)
[12] Benedetto D, Caglioti E and Loreto V 2002 *Phys. Rev. Lett.* **88** 048702
[13] Otu H H and Sayood K 2003 *Bioinformatics* **19** 2122
[14] Li M, Chen X, Li X, Ma B and Vitanyi P M B 2004 *IEEE Trans. Inf. Theory* **50** 3250
[15] Lempel A and Ziv J 1977 *IEEE Trans. Inf. Theory* **23** 337
[16] Wyner A D and Ziv J 1994 *Proc. IEEE* **82** 872
[17] Kullback S and Leibler R A 1951 *Ann. Math. Stat.* **22** 79
[18] Kullback S 1959 *Information Theory and Statistics* (New York: Wiley)
[19] Ziv J and Merhav N 1993 *IEEE Trans. Inf. Theory* **39** 1270
[20] Puglisi A, Benedetto D, Caglioti E, Loreto V and Vulpiani A 2003 *Physica* D **180** 92
[21] Baronchelli A, Caglioti E and Loreto V 2005 *J. Stat. Mech.* P04002
[22] Schuermann T and Grassberger P 1996 *Chaos* **6** 414
[23] http://www.math.ucsd.edu/~crypto/java/ENTROPY/
[24] Crisanti A, Paladin G and Vulpiani A 1993 *Products of Random Matrices in Statistical Physics* (Berlin: Springer)
[25] Huffman D A 1952 *Proc. Inst. Radio Eng.* **40** 1098