ELSEVIER

Discrete Mathematics 223 (2000) 367–372

DISCRETE MATHEMATICS

Note

# A proof of the Beyer–Stein–Ulam relation between complexity and entropy

Stefano Galatolo

*Dipartimento di Matematica, Università di Pisa, Via Buonarroti 2, 56127 Pisa, Italy*

Received 20 May 1999; revised 13 December 1999; accepted 31 January 2000

**Abstract**

We present a proof of a conjecture stated by Beyer, Stein and Ulam in 1971. The authors conjectured a relation between Kolmogorov complexity and information entropy.© 2000 Elsevier Science B.V. All rights reserved.

*MSC:* 68Q30; 94A17

*Keywords:* Kolmogorov; Complexity; Entropy

## 1. Introduction

Beyer et al. [1] defined and discussed several notions regarding complexity of integers and strings. They also made computer experiments to study the behavior of the complexities and to support a conjecture about a relation between Kolmogorov complexity and Shannon entropy. Lynch [8] proves a slightly weaker form of this conjecture adding some technical hypotheses (for example that the involved probabilities are computable numbers). In this paper we prove the conjecture. The proof is also elementary and does not require almost any notion of probability theory. Only the weak form of the law of large numbers is needed (see e.g. [3] and [4]).

We summarize the concepts leading to the definition of Kolmogorov complexity and to the statement of the conjecture (see [2] or [7] for an introduction to Algorithmic Information Theory). Let $A$ be an algorithm (Partial Recursive Function) that transforms finite binary strings to finite binary strings. The Kolmogorov complexity $K_A(x)$ of a string $x$ relative to $A$ is the length of the shortest string $p$ such that $A(p)=x$, if there are no strings $p$ such that $A(p)=x$ then $K_A(x)=\infty$. The string $p$ can be imagined

*E-mail address:* galatolo@dm.unipi.it (S. Galatolo).

PII: S0012-365X(00)00100-X

as a program given to a Turing machine representing $A$, and the value $A(p)$ can be imagined as the output of the computation. If $A$ is an algorithm transforming pairs of binary strings to binary strings, the conditional complexity $K_A(x|y)$ of $x$ given $y$ is the length of the shortest string $p$ such that $A(p,y)=x$, or if no such string exists, it is $\infty$. We will consider $K_A(x|n)$, where $n$ is a string representing the binary expansion of the length of the string $x$. In other words, $K_A(x|n)$ is the complexity of $x$ given its length.

Now, let us consider the set $\{0,1\}^n$ of binary strings of length $n$, and assign a probability to each digit 0, 1: let $\mathrm{pr}(0)=P_0$ and $\mathrm{pr}(1)=P_1=1-P_0$ $(0\leqslant P_0\leqslant 1)$. The probability of a finite string $a_1\ldots a_n$ is $\prod_{i=1}^{n}\mathrm{pr}(a_i)$. The Shannon entropy of the probability scheme

$$\begin{pmatrix} 0 & 1 \\ P_0 & P_1 \end{pmatrix}$$

(see [6] or [9]) is the real number

$$H=-P_0\log P_0-P_1\log P_1$$

(all the logarithms are in base 2). We will not discuss here the important features of the number $H$ or the theory of information sources; an introduction can be found in [6,9]. We remark that in this paper we only consider 0-memory information sources, as it was in [1]. Generalizations of the statements are possible if we consider sources with memory (see [8]).

## 2. The conjecture

Now, we are ready to state the conjecture as it was stated by the authors in [1]:

**Conjecture 1.** For every natural number $n$, for every real $r\in(1/2,1)$: let $x_1,\ldots,x_{2^n}$ be the sequence of all binary strings of length $n$ arranged in order of decreasing probability. Let $k(n)$ be the least integer such that $\sum_{i=1}^{k(n)}\mathrm{pr}(x_i)>r$. If $K_A$ is normalized so that

$$\frac{1}{k(n)}\sum_{i=1}^{k(n)}K_A(x_i|n)=1$$

when $P_0=P_1=1/2$ then

$$H\simeq\frac{1}{k(n)}\sum_{i=1}^{k(n)}K_A(x_i|n)$$

for arbitrary $P_0\in(0,1)$.

As already observed by Lynch [8], the normalization factor is $1/n$ and it is natural to require the algorithm $A$ to be universal.[1] The reasons are that in the definition of Kolmogorov complexity, it is standard to require that the algorithm be universal.[2] Moreover, the conjecture is not true for all algorithms. It is easy to construct counter examples using nonuniversal algorithms. For example, if $A$ is the identity $A(x)=x$ we have a counterexample to the conjecture as it was stated.

We prove that:

**Theorem 2.** *Let $A$ be universal, for each $P_0, P_1 \in (0,1)$ s.t. $P_0+P_1=1$. If the strings are ordered and $k(n)$ is defined as in conjecture* 1:

$$H = \lim_{n\to\infty} \frac{1}{nk(n)} \sum_{i=1}^{k(n)} K_A(x_i|n) \tag{1}$$

**Proof** (*Upper bound*). Let us define $Q_n = \{x_1,\dots,x_{k(n)}\} \subset \{0,1\}^n$, the set over which the summation is performed. Let $\varepsilon_n$ be a sequence converging to zero such that the set

$$S_n = \{x \in \{0,1\}^n \mid \mathrm{pr}(x) > 2^{-n(H+\varepsilon_n)}\}$$

is such that $\mathrm{pr}(S_n) \to 1$. The existence of $\varepsilon_n$ follows from the weak law of large numbers.[3] We also remark that we can require $\varepsilon_n$ to be a rational number of the form $\varepsilon_n = 2^{-f_n}$ with $f_n \in N$.

Let us define $S'_n = S_n \cap Q_n$; since $\mathrm{pr}(S_n) \to 1$ then $S'_n = Q_n$ eventually.

For each $x \in S'_n$ we will find a program $p(\varepsilon_n, n, m_x, H'_n, L^0_n, L^1_n)$ such that $x = A(p(\varepsilon_n, n, m_x, H'_n, L^0_n, L^1_n))$ and the length of $p(\varepsilon_n, n, m_x, H'_n, L^0_n, L^1_n) = nH_n + o(n)$. This will prove that

$$\limsup_{n\to\infty} \frac{1}{nk(n)} \sum_{i=1}^{k(n)} K_A(x_i|n) \leqslant H.$$

The parameters $\varepsilon_n, n, m_x, H'_n, L^0_n, L^1_n$ are rational numbers codified by binary strings. Their meaning will be specified in the following lines.

First, let us estimate the number of strings in $S'_n$: since the probability of each string is greater than $2^{-Hn-\varepsilon_n n}$ and the total sum of probabilities is not greater than 1, if $N_n$ is the number of strings in $S'_n$ then $N_n 2^{-Hn-\varepsilon_n n} \leqslant 1$ and $N_n \leqslant 2^{Hn+\varepsilon_n n}$.

If $a$ is a real number, we define $\mathrm{trunc}_n(a) = \mathrm{int}(2^n a)/2^n$ (where int() is the integer part). The binary expansion of $\mathrm{trunc}_n(a)$ will stop at the $n$th digit. Now, let us define

[1] Roughly speaking, a universal algorithm is an algorithm that can simulate any other algorithm if an appropriate input is given. For a precise definition see any book of recursion.

[2] Because if $U$ and $U'$ are universal then $K_U(s) \leqslant K_{U'}(s) + c$ where $c$ is a constant depending only on $U$ and $U'$, this tells that when using the universal algorithm $U$ the complexity of $s$ with respect to $U$ depends only on $s$ up to a fixed constant.

[3] Let $v^i(x)$ be the number of times that the digit $i \in \{0,1\}$ occurs in the string $x \in \{0,1\}^n$. By an immediate corollary of the weak law of large numbers there is a sequence $\delta_n$ converging to 0, such that $\lim_{n\to\infty} \mathrm{pr}\{|v^i(x)/n - P_i| < \delta_n\} = 1$. The existence of $\varepsilon_n$ follows from the following identity $\mathrm{pr}(x) = e^{n([v^0(x)/n]\log(P_0)+[v^1(x)/n]\log(P_1))}$.

$l = \mathrm{int}(\log(n^2) + 1)$ and $H_n' = \mathrm{trunc}_l(H) + 1/2^l$; this rational number can be codified by a string of length not greater than constant $+\log(n^2)$. We also have $H_n' > H$ and $|H_n' - H| < 1/2^{l-1}$. Consider $L_n^0 = \mathrm{trunc}_l(\log(P_0)) - 1/2^l$ and $L_n^1 = \mathrm{trunc}_l(\log(P_1)) - 1/2^l$. $L_n^0$ and $L_n^1$ can be codified by strings of length not greater than constant2 $+\log(n^2)$ and $L_n^1 < \log P_1$, $L_n^0 < \log P_0$. Moreover, if $v^i(x)$, $i \in \{0,1\}$ is the number of times that the digit $i$ occurs in the string $x$, then

$$\left| \log \mathrm{pr}(x) - \sum_{i \in \{0,1\}} v^i(x) L_n^i \right| < \frac{n}{2^{l-1}}$$

and then, if $x \in S_n'$

$$-\sum_{i \in \{0,1\}} v^i(x) L_n^i < -\log(\mathrm{pr}(x)) + \frac{n}{2^{l-1}} < n(H + \varepsilon_n) + \frac{n}{2^{l-1}}$$

$$< n(H_n' + \varepsilon_n) + \frac{n}{2^{l-1}}.$$

It is possible to estimate the number $m$ of strings satisfying the condition

$$-\sum_{i \in \{0,1\}} v^i(x) L_n^i < n(H_n' + \varepsilon_n) + \frac{n}{2^{l-1}}. \tag{2}$$

Since $L_n^i < \log P^i$ then $-\sum_{i \in \{0,1\}} v^i(x) L_n^i > -\log(\mathrm{pr}(x))$ this implies that if $x$ satisfies condition (2) then it satisfies $\mathrm{pr}(x) > 2^{-n(H_n'+\varepsilon_n)-n/2^{l-1}}$ and then the number $m$ is not greater than the number of $x$ satisfying $\mathrm{pr}(x) > 2^{-n(H_n'+\varepsilon_n)-n/2^{l-1}}$. This implies that

$$m \leqslant 2^{n(H_n'+\varepsilon_n)+n/(2^{l-1})} < 2^{n(H+1/(2^{l-1})+\varepsilon_n)+n/(2^{l-1})}. \tag{3}$$

From this last consideration, it follows that there is an integer $m_x \leqslant m$ such that the string $x$ is the output of the program $p(\varepsilon_n, n, m_x, H_n', L_n^0, L_n^1)$ which does the following things:

```
1 Order (in lexicographic order) all the strings of length $n$ such
that $-\sum v^i(x)L_n^i < n(H_n' + \varepsilon_n) + n/2^{l-1}$ until the $m_x$-nt
2 the output is this string
```

The length of this program is given by a constant (the program $p$), a representation of $\varepsilon_n$ that can be done in $O(\log n)$ digits ($\varepsilon_n$ can be chosen to be of the form $2^{-f_n}$), a string representing $n$ that can be given in $O(\log n)$ digits, a representation of $H_n'$ which can be given by a string of length constant$'+O(\log n^2)$, two strings representing $L_n^0$, $L_n^1$ with length 2 (constant$''+O(\log n^2)$) and a representation of $m_x$ (the main part). Eq. (3) shows that $m_x$ can be represented by a string of length $nH + o(n)$. The total length of the program is $nH + o(n) + O(\log n) + O(\log n^2) +$ constants, which proves the upper bound.

*Lower bound*: Let $\varepsilon_n$ be a sequence converging to 0 such that the set

$$M_n = \{x \in \{0,1\}^n \text{ s.t. } 2^{-Hn-\varepsilon_n n} < \mathrm{pr}(x) < 2^{-Hn+\varepsilon_n n}\}$$

is such that $\mathrm{pr}(M_n) \to 1$. As before the existence of $\varepsilon_n$ follows from the law of large numbers.

Let us define $M_n' = M_n \cap Q_n$.

Since $r < \mathrm{pr}(Q_n) < r + \max(P^0, P^1)^n$, we have, from the law of large numbers that

$$\mathrm{pr}(M_n') \to r.$$

Now, we estimate the cardinality of $M_n'$. For all $x$ in $M_n' \mathrm{pr}(x) < 2^{-Hn+\varepsilon_n n}$, if $N = \#(M_n')$ is the cardinality of $M_n'$, since the probability of $M_n'$ tends to $r$ then $N2^{-Hn+\varepsilon_n n} > r - \gamma_n$, where $\gamma_n$ is a real sequence converging to 0, then $N > (r - \gamma_n)2^{H - \varepsilon_n n}$.

Since $M_n' \subset Q_n$ this implies that $\#Q_n > (r - \gamma_n)2^{Hn - \varepsilon_n n}$.

To estimate (1) consider the set

$$Q_n^c = \{x \in Q_n \text{ s.t. } K_A(x|n) < Hn - \varepsilon_n n - c\}$$

since the number of programs of length $\leqslant k$ is less than $2^{k+1}$ we have

$$\#Q_n^c < 2^{Hn - \varepsilon_n - c + 1},$$

$$\frac{\#Q_n^c}{\#Q_n} < \frac{2^{Hn - \varepsilon_n n - c + 1}}{(r - \gamma_n)2^{Hn - \varepsilon_n n}} < 2^{-c+1-\log(r-\gamma_n)}$$

this implies that $\forall c$

$$\frac{1}{k(n)n} \sum_{x \in Q_n} K(x|n) > \frac{1}{n}(1 - 2^{-c+1-\log(r-\gamma_n)})(Hn - \varepsilon_n n - c)$$

if $n \to \infty$ the right side tends to $(1 - 2^{-c+1-\log(r)})H$, if $c$ is large enough this is as near as we want to $H$, and this proves the assertion. □

We remark that in the lower bound proof we did not make use of any property of the algorithm $A$. This implies that the lower bound is true even if we use non universal algorithms to define Kolmogorov complexity.

We also remark that the proof of Theorem 1 can be generalized to the case where $n$-ary strings are considered instead of binary ones [5].

According to [1,8] the philosophical meaning of Conjecture 1 is that 'The most likely sequences from $A$ have complexity approximately equal to the entropy'. Since Eq. (1) estimates the mean complexity of the strings in $Q_n$, to be more precise we could replace the word 'approximately' with 'on average'. We remark that the proof of Theorem 2 can be slightly modified to prove a stronger result.

**Proposition 3.** *Let $A$ be universal and let us define the set*

$$K_n^\varepsilon = \left\{x \in \{0,1\}^n \text{ s.t. } H - \varepsilon < \frac{K_A(x)}{n} < H + \varepsilon\right\}$$

*then* $\forall\varepsilon \lim_{n\to\infty} \mathrm{pr}(K_n^\varepsilon) = 1$.

That is *almost all sequences* (*for the probability measure*) *have a complexity asymptotically equal to the entropy*. Similar and more general results can be found in

[5], where algorithmic complexity is used to define a notion of entropy for points in metric spaces (If we consider information sources as a metric space, [5] we obtain a result similar to Proposition 3).

## References

[1] W.A. Beyer, L. Stein, S.M. Ulam, The notion of complexity, Los Alamos Report LA-4822, 1971.

[2] G.J. Chaitin, Information, randomness and incompleteness. Papers on Algorithmic Information Theory, World Scientific, Singapore, 1987.

[3] W. Feller, An introduction to Probability Theory and its Applications. Vol. II, 2nd Edition, Wiley, New York, 1971.

[4] B.V. Gnedenko, The Theory of Probability, Mir, Moscow, 1982.

[5] S. Galatolo, Pointwise information entropy for metric spaces, Nonlinearity 12 (5) (1999) 1289–1298.

[6] A.I. Khinchin, Mathematical Foundations of Information Theory, Dover Publications, New York, 1957.

[7] M. Li, P. Vitanyi, An introduction to Kolmogorov complexity and its applications, 2nd Edition, Graduate Texts in Computer Science, Springer, New York, 1997.

[8] J.F. Lynch, A relation between complexity and entropy, Ulam Quarterly 3 (1995) 7–14. http://www.ulam.usm.edu/.

[9] S. Roman, Introduction to coding and information theory, Undergraduate Texts in Mathematics, Springer, New York, 1997.