Time-universal data compression and prediction

Boris Ryabko
Institute of Computational Technologies of SB RAS
Novosibirsk state university
Novosibirsk, Russian Federation
Email: boris@ryabko.net

***Abstract*—Suppose there is a large file which should be transmitted (or stored) and there are several (say, m) admissible data-compressors. It seems natural to try all the compressors and then choose the best, i.e. the one that gives the shortest compressed file. Then transfer (or store) the index number of the best compressor (it requires $\lceil \log m \rceil$ bits) and the compressed file. The only problem is the time, which essentially increases due to the need to compress the file m times (in order to find the best compressor). We propose a method that encodes the file with the optimal compressor, but uses a relatively small additional time: the ratio of this extra time and the total time of calculation can be limited by an arbitrary positive constant. A similar situation occurs when forecasting time series.**

Generally speaking, in many situations it may be necessary find the best data compressor (or predictor) out of a given set, which is often done by comparing them empirically. One of the goals of this work is to turn such a selection process into a part of the data compression method, automating and optimizing it.

I. Introduction and preliminaries

A. General description of the problems and results

Nowadays there are many efficient lossless data-compressors (or archivers) which are widely used in information technologies. These compressors are based on different ideas and approaches, among which we note the PPM universal code [1] (which is used along with the arithmetic code [2]), the Lempel-Ziv (LZ) compression methods [3], the Burrows-Wheeler transform [4] (which is used along with the book-stack (or MTF) code [5]–[7]), the class of grammar-based codes [8], [9] and some others [10]–[12]. All these codes are universal. This means that, asymptotically, the length of the compressed file goes to the smallest possible value (i.e. the Shannon entropy per letter), if the compressed sequence is generated by a stationary source.

Currently, several dozens of archivers are known, each of which has certain merits and it is impossible to single out one of the best or even remove the worst ones. The main part of them are universal codes (as far as a computer program can meet asymptotic properties). Thus, the one faces the problem of choosing the best method to compress a given file. A similar situation occurs when forecasting time series. Often there are many reasonable forecasting methods and the task of choosing the most precision for the forecast arises. We propose a method that allows us to use asymptotically the best forecast expending a short extra-time.

Suppose someone wants to compress a certain file in order to store it (or transfer it). It seems natural to use for compression the best compressor: the one which gives the shortest compressed file. In such a case one can try to compress the file in turn by all the compressors and then store the name of the best compressor (as a prefix) and the file, compressed by the best method. An obvious drawback of this approach is the need to spend a lot of time in order to first compress the file by all the compressors.

In this paper we suggest time-universal methods for data compression and forecasting.

In this paper we show that there exists a method that encodes the file with the optimal compressor, but uses a relatively small additional time. Very briefly, the main idea of the suggested approach is as follows: in order to find the best, try all the archivers, but, when doing it, use for compression only a small part of the file. Then apply the best archiver for the compression of the whole file. It turns out, that under certain conditions on the source of the files, the total time can be made as close to the minimal as required. Thus, we call such methods "time-universal".

In this paper we suggest time-universal methods for data compression and forecasting. To the best of our knowledge, the suggested approach to compression is new, but close ideas have been considered in algorithmic information theory and artificial intelligence, where they were developed for solving other problems [13], [14].

B. The over-fitting problem

If someone wants to find the best method of prediction or data compression, she/he should take into account the so-called over-fitting problem. The over-fitting problem is the phenomenon in which the accuracy of the model on unseen data is poor whereas the training accuracy is nearly perfect. In our case, there is a set of either data compressors $F = \{\varphi_1, \varphi_2, ...\}$ or predictors $\Pi = \{\pi_1, \pi_2, ...\}$. Besides, there is a sequence $x_1 x_2 ... x_n, n > 1$, and one should choose a good method from the set of predictors (or data compressors) based on investigating of a short initial part $x_1 x_2 ... x_l$, $l < n$. In a case of data compression, it is natural to choose such a method $\hat{\varphi} \in F$, for which $|\hat{\varphi}(x_1 x_2 ... x_l)|$ is minimal. In a case of forecasting it is natural to choose such a predictor $\hat{\pi} \in \Pi$ for which $\hat{\pi}(x_1 x_2 ... x_l)$ maximal (it is well-known maximum likelihood principle.)

In this situation the problem of over-fitting is as follows: if $x_1 ... x_l$ is relatively short sequence, and the set of methods F is large or even infinite, it is possible that a performance of the chosen method φ on $x_1 ... x_l$ is good, but on the whole sequence

$x_1 ... x_n$ is bad. The over-fitting problem for prediction is similar: the error of the chosen predictor on unseen data is large whereas the training error is nearly zero.

We consider a solution of this problem based on approach developed in the theory of universal coding [11], [15], but note that the similar solution can be obtained in a framework of MDL (minimal description length) method suggested by J. Rissanen [16], [17] and widely developed in numerous papers [18]–[20]. For this we need such a probability distribution ω on the set $1, 2, 3, ...$ for which all $\omega_i > 0$. For example, such a distribution can be as follows:

$$\omega_k = \frac{1}{k(k+1)}\,,\ k = 1, 2, 3, ... \,. \qquad (1)$$

(Clear, it is the probability distribution, because $\omega_k = 1/k - 1/(k+1)$). The described approach to the over-fitting problem suggests to find a data-compressor φ_s for which $-\log \omega_s + |\varphi_s(x_1 x_2 ... x_l)|$ is minimal: $-\log \omega_s + |\varphi_s(x_1 x_2 ... x_l)| = \min_{i=1,2,...} (-\log \omega_i + |\varphi_i(x_1 x_2 ... x_l)|)$. Note, that if the set of data-compressors is finite, it is possible to use an uniform distribution $\omega_i = 1/|F|$, $i = 1, ..., |F|$. It is worth noting that there is a natural interpretation of the considered solution. The value $\lceil -\log \omega_i \rceil + |\varphi_i(x_1 x_2 ... x_n)|$ can be considered as a codeword length, where the first part $\lceil -\log \omega_i \rceil$ codes the number i, whereas the second part $\varphi_i(x_1 x_2 ... x_n)$ encodes $x_1 x_2 ... x_n$ by the data compressor φ_i.

In the case of prediction the solution of the over-fitting problem is similar: find a predictor π_s for which $\omega_s \pi_s((x_1 x_2 ... x_l)$ is maximal.

II. Description of problems and the main notations

In this section we first consider the following problem: There is a set of data compressors $F = \{\varphi_1, \varphi_2, ...\}$ and let $x_1 x_2 ...$ be a sequence of letters from a finite alphabet A whose initial part $x_1 ... x_n$ should be compressed by some $\varphi \in F$. Let, as before, v_i be the time spent on encoding one letter by the data compressor φ_i and suppose that all v_i are upper-bounded by a certain constant v, i.e. $\sup_{i=1,2,...,} v_i \leq v$. (Note, that v_i can be unknown beforehand.)

It is important to note that in practice we do not need to know the speed of the data compressors, but we can estimate $delta$. For example, suppose someone has 20 data compressors and a file. Suppose he wants to find a good data compressor for this file. For this, firstly, he takes the initial 1 % segment of the file, and compresses it with each data compressor. Then he chooses the best one and compresses the file with this compressor. Thus, the share of total extra time ($delta$) is bounded above by $20 \times 0.01 = 0.2$. Therefore, he may not know the speed of all of the data compressors (which, of course, may depend on the computer). In the examples below, we use this approach.

So, the goal is to find a good data compressor from F in order to compress $x_1 ... x_n$ in such a way that the total time spent for all calculations and compressions does not exceed $T(1+\delta)$, $\delta > 0$, where $T = v\, n$ is the minimum time that must be reserved for compression and δT is the additional time that can be used to find the good compressor (among $\varphi_1, \varphi_2, ...$). But, as mentioned above, we can estimate δ without knowing the speeds.

In order to accurately describe the problem, we suppose also that there is a probability distribution $\omega = \omega_1, \omega_2, ...$ such that all $\omega_i > 0$. The goal is to fined such φ_i that the value

$$\lceil -\log \omega_i \rceil + |\varphi_i(x_1 x_2 ... x_n)|$$

is close to minimal. (Here the first part $\lceil -\log \omega_i \rceil$ is used for encoding number i.) The decoder first finds i and then $x_1 x_2 ... x_n$ using the decoder corresponding φ_i.

Definition 1: We call any method that encodes a sequence $x_1 x_2 ... x_n$, $n \geq 1$, $x_i \in A$, by the binary word of the length $\lceil -\log \omega_j \rceil + |\varphi_j(x_1 x_2 ... x_n)|$ for some $\varphi_j \in F$, a time-adaptive code and denote it by $\hat{\Phi}^{\delta}_{compr}$. The output of $\hat{\Phi}^{\delta}_{compr}$ is the following word:

$$\hat{\Phi}^{\delta}_{compr}(x_1 x_2 ... x_n) = <\omega_i> \ \varphi_i(x_1 x_2 ... x_n)\,, \qquad (2)$$

where $<\omega_i>$ is $\lceil -\log \omega_i \rceil$-bit word that encodes i, whereas the time of encoding is not grater than $T(1+\delta)$.

If for a time-adaptive code $\hat{\Phi}^{\delta}_{compr}$ the following equation is valid

$$\lim_{t\to\infty} \hat{\Phi}^{\delta}_{compr}(x_1 ... x_t)/t = \inf_{1=1,2,...} \lim_{t\to\infty} \varphi_i(x_1 ... x_t)/t\,,$$

this code is called time-universal.

The definition for the forecast is as follows: Let there be a set of predictors $\Pi = \{\pi_1, \pi_2, ...\}$. The goal of the time-adaptive predictor $\hat{\Phi}^{\delta}_{pred}$ is to spend the extra time δT in order to find such π_i that the value

$$\omega_i \, \pi_i(x_1 x_2 ... x_n)$$

is close to maximal. By definition, the output of the time-adaptive predictor $\hat{\Phi}^{\delta}_{pred}$ is the following set of forecasts (conditional probabilities):

$$\{\pi_j(a|x_1 ... x_n),\ a \in A\},$$

for a certain $\pi_j \in \Pi$. It will be convenient to define

$$\hat{\Phi}^{\delta}_{pred}(x_1 x_2 ... x_n) = \omega_i \, \pi_j(x_1 ... x_n)\ . \qquad (3)$$

If for a predictor $\hat{\Phi}^{\delta}_{pred}$ the following equation is valid

$$\lim_{t\to\infty} (-\log \hat{\Phi}^{\delta}_{pred}(x_1 ... x_t))/t =$$

$$\inf_{1=1,2,...} \lim_{t\to\infty} (-\log \pi_i(x_1 ... x_t))/t\,,$$

and, for any t, time of calculation is not grater than $T(1+\delta)$ this predictor is called time-universal.

In what follows, we will consider the case of data compression, taking into account the fact that all methods and results are easily transferred to the case of forecasting.

Comment 1. In the case of data compression it will be convenient to reckon that the whole sequence is compressed not by per one letter, but by sub-words each of which, say, a few kilobytes in length. More formally, let, as before, there be a sequence $x_1x_2...$ where x_i , $i = 1, 2, ...$ are sub-words whose length (say, L) can be a few kilobytes. In this case $x_i \in \{0, 1\}^{8L}$.

Comment 2. Here and below we did not take into account the time required for the calculation of $\log \omega_i$ and some other auxiliary calculations. If in a certain situation this time is not negligible, it is possible to reduce $\hat{T}$ in advance by the required value.

III. The direct method

Suppose that there is a file $x_1x_2...x_n$ and data compressors $\varphi_1, ..., \varphi_m$, $n \geq 1, m \geq 1$. Let, as before, v_i be the time spent on encoding one letter by the data compressor φ_i,

$$v = \max_{i=1,...,n} v_i, \;\; T = n\, v\,, \tag{4}$$

and let

$$\hat{T} = T(1 + \delta)\,, \;\; \delta > 0. \tag{5}$$

The goal is to find the data compressor φ_j, $j = 1, ..., m$, that compresses the file $x_1x_2...x_n$ in the best way in time $\hat{T}$. Seemingly, the simplest method is as follows:

Step 1 Calculate $r = \lfloor \delta T/v \rfloor$.

Step 2 Compress the file $x_1x_2...x_r$ by φ_1 and find the length of compressed file $|\varphi_1(x_1...x_r)|$, then , likewise, find $|\varphi_2(x_1...x_r)|$, $|\varphi_3(x_1...x_r)|$, etc.

Step 3 Calculate $s = \arg\min_{i=1,...,m} |\varphi_i(x_1...x_r)|$

Step 4 Compress the whole file $x_1x_2...x_n$ by φ_s and compose the codeword $\langle s \rangle\ \varphi_s(x_1...x_n)$, where $\langle s \rangle$ is $\lceil \log m \rceil$-bit word with the presentation of s.

The decoding is obvious. Denote this method by Φ_1^δ.

The asymptotic properties of the method Φ_1^δ are as follows:

Claim 1. Let there be an infinite sequence $x_1, x_2, ...$ and data compressors $\varphi_1, ..., \varphi_m$ such that there exist the following limits

$$\lim_{n \to \infty} |\varphi_i(x_1x_2...x_n)|/n\ , \tag{6}$$

for all $i = 1, ..., m$. Then, for any $\delta > 0$

$$\lim_{n \to \infty} |\Phi_1^\delta(x_1x_2...x_n)|/n \; = \min_{1,...,m} \lim_{n \to \infty} |\varphi_i(x_1x_2...x_n)|/n\ ,$$

i.e. Φ_1^δ is time-universal. Later we describe a more general method, for which Claim 1 is a special case.

IV. General consideration

We saw in the previous paragraphs that there are many reasonable methods and each of them has a lot of parameters. That is why, it could be useful to use multidimensional optimization approaches, such as machine learning, so-called deep learning, etc. Since this is the first paper devoted to time-adaptive and time-universal data compression, we consider only some general conditions needed for time-universality.

For a time-adaptive data-compressor $\hat{\Phi}$ and $x_1...x_t$ we define for any φ_i

$$\tau_{\varphi_i}(t) = \max\{r : \varphi_i(x_1...x_r)\ is\ caculated,$$

$$when\ \hat{\Phi}(x_1...x_n)\ is\ applied.$$

Theorem 1: Let there be a time-adaptive method $\hat{\Phi}$ whose additional time of calculation is not grater than δT. If the following properties are valid:

i) for all $i = 1, 2, ...$

$$\lim_{t \to \infty} \tau_i(t) = \infty, \tag{7}$$

ii) for any t the method $\hat{\Phi}$ uses such a compressor $\varphi_{s(t)}$ for which, for any i and $r = \min\{\tau_i, \tau_{s(t)}\}$

$$-\log \omega_{s(t)} + |\varphi_{s(t)}(x_1...x_r)| \leq -\log \omega_i + |\varphi_i(x_1...x_r)|\,, \tag{8}$$

iii) the limits $\lim_{t\to\infty} \varphi_i(x_1...x_t)/t$ exist for all φ_i.

Then $\hat{\Phi}(x_1...x_n)$ is time universal, i.e., in a case of data compression,

$$\lim_{t \to \infty} \hat{\Phi}(x_1...x_t)/t \; = \inf_{i=1,2,...} \lim_{t \to \infty} |\varphi_i(x_1...x_t)|/t \tag{9}$$

A proof is given in Appendix, but here we note that this theorem is valid for the methods described earlier.

Comment 5. If the sequence $x_1x_2...$ is generated by a stationary source and all φ_i are universal codes, the property iii) is valid with probability 1 (See, for example, [21]). Hence, this theorem (and the claims) are valid for this case.

In general, the property iii) shows that the sequence under consideration has some stability. In turn, it gives a possibility to estimate characteristics of the whole sequence $x_1x_2...$ based on its initial part.

V. Examples

The experiments described below should give an idea of the potential of the proposed approach. To do this, we took 22 data compressors from [22] and 14 files of different lengths. For each file we applied the following two-step scheme: first we took 1% of the file and sequentially compressed it with all the data compressors. Then we selected the three best compressors, took 5% of the file, and sequentially compressed it with the three compressors selected. Finally, we selected the best of these compressors and compressed the file with this compressor. Thus, the total additional time is limited to 22 $\times$ 0.01 + 3 $\times$ 0.05 = 0.37, i.e. $\delta \leq 0.37$. Table 1 contains the obtained data. This table shows that the larger the file,

TABLE I: Two-step compression. Extra-time δ = 0.37.

file	legth	best compressor	chosen compressor	chosen/best (ratio of length)
BIB	111261	nanozip	lpaq8	1.06
BOOK1	768771	nanozip	nanozip	1
BOOK 2	610856	nanozip	nanozip	1
GEO	102400	nanozip	ccm	1.07
NEWS	377109	nanozip	nanozip	1
OBJ1	21504	nanozip	tornado	1.23
OBJ2	246814	nanozip	lpaq8	1.08
PAPER1	53161	nanozip	tornado	1.52
PAPER2	82199	nanozip	tornado	1.54
PIC	513216	zpaq	bbb	1.25
PROGC	39611	nanozip	tornado	1.42
PROGL	71646	nanozip	tornado	1.44
PROGP	49379	lpaq8	tornado	1.4
TRANS	93695	lpaq8	lpaq8	1

the better the compression. The following table gives some insight into the effect of the extra time. Here we used the same two-stage scheme, but the size of the parts was 2% and 10% for the first step and the second, respectively, while the extra time was 0.74. From the table it can be seen that the performance of the considered scheme increases significantly when the additional time increases.

TABLE II: Two-step compression. Extra-time δ = 0.74.

file	legth	best compressor	chosen compressor	chosen/best (ratio of length)
BIB	111261	nanozip	nanozip	1
BOOK1	768771	nanozip	nanozip	1
BOOK 2	610856	nanozip	nanozip	1
GEO	102400	nanozip	nanozip	1
NEWS	377109	nanozip	lpq1v2	1.14
OBJ1	21504	nanozip	ccm	1.17
OBJ2	246814	nanozip	nanozip	1
PAPER1	53161	nanozip	lpaq8	1.19
PAPER2	82199	nanozip	nanozip	1
PIC	513216	zpaq	bbb	1.25
PROGC	39611	nanozip	lpaq8	1.04
PROGL	71646	nanozip	lpaq8	1.03
PROGP	49379	lpaq8	lpaq8	1
TRANS	93695	lpaq8	lpaq8	1

VI. The time-universal code for stationary ergodic sources

In this section we describe a time-universal code for stationary sources. It is based on optimal universal codes for Markov chains, developed by Krichevsky [23], [24] and the twice-universal code [11]. Denote by M_i, $i = 1, 2, ...$ the set of Markov chains with memory (connectivity) i, and let M_0 be the set of Bernoulli sources. For stationary ergodic μ and an integer r we denote by $h_r(\mu)$ the r-order entropy (per letter) and let $h_\infty(\mu)$ be the limit entropy; see for definitions [21].

Krichevsky [23], [24] described the codes $\psi_0, \psi_1, ...$ which are asymptotically optimal for $M_0, M_1, ...$, correspondingly. If the sequence $x_1x_2...x_n$, $x_i \in A$, is generated by a source $\mu \in M_i$, the following inequalities are valid almost surely (a.s.):

$$h_i(\mu) \le |\psi_i(x_1...x_t)|/t \ \le h_i(\mu) + ((|A|-1)|A|^i + C)/t, \tag{10}$$

where t grows. (Here C is a constant.) The length of a codeword of the twice-universal code ρ is defined as the following "mixture":

$$|\rho(x_1...x_t)| = -\log \sum_{i=0}^{\infty} \omega_{i+1}\, 2^{-|\psi_i(x_1...x_t)|} \tag{11}$$

(It is well-known in Information Theory [21] that there exists a code with such codeword lengths, because $\sum_{x_1...x_t \in A^t} 2^{-|\rho(x_1...x_t)|} = 1$.) This code is called twice-universal because for any M_i, $i = 0, 1, ...$, and $\mu \in M_i$ the equality (10) is valid (with different C). Besides, for any stationary ergodic source μ a.s.

$$\lim_{t\to\infty} |\rho_i(x_1...x_t)|/t \ = h_\infty(\mu). \tag{12}$$

Let us estimate the time of calculations necessary when using ρ. First, note that it suffices to sum a finite number of terms in (11), because all the terms $2^{-|\psi_i(x_1...x_t)|}$ are equal for $i \ge t$. On the other hand, the number of different terms grows, where $t \to \infty$ and, hence, the encoder should calculate $2^{-|\psi_i(x_1...x_t)|}$ for growing number i's. It is known [11] that the time spent for encoding one letter is close for different codes ψ_i. Hence, the time spent for encoding one letter by the code ρ grows to infinity, when t grows. The described below time-universal code Ψ^δ has the same asymptotic performance, but the time spent for encoding one letter is a constant.

In order to describe the time-universal code Ψ^δ we give some definitions. Let, as before, v be an upper-bound of the time spent for encoding one letter by any ψ_i, $x_1...x_t$ be the generated word,

$$T = t\, v,\ N(t) = \delta T/v = \delta\, t,$$
$$m(t) = \lfloor \log\log N(t) \rfloor,\ s(t) = \lfloor N(t)/m(t) \rfloor . \tag{13}$$

Denote by Ψ^δ the following method:

Step 1 Calculate $m(t), s(t)$ and

$$|\psi_0(x_1...x_{s(t)})|, |\psi_1(x_1...x_{s(t)})|, ..., |\psi_{m(t)}(x_1...x_{s(t)})| .$$

Step 2 Find such a j that

$$|\psi_j(x_1...x_{s(t)})| = \min_{i=0,...,m(t)} |\psi_i(x_1...x_{s(t)})|.$$

Step 3 Calculate the codeword $\psi_j(x_1...x_t)$ and output

$$\Psi^\delta(x_1...x_t) = <j> \psi_j(x_1...x_t)\,,$$

where $<j>$ is the $\lceil -\log \omega_{j+1} \rceil$-bit codeword of j.

The decoding is obvious.

Theorem 2: Let $x_1x_2...$ be a sequence generated by a stationary source and the code Ψ^δ be applied. Then this code is time-universal, i.e. a.s.

$$\lim_{t\to\infty} |\Psi^\delta(x_1...x_t)|/t = \inf_{i=0,1,...} \lim_{t\to\infty} |\psi_i(x_1...x_t)|/t\,. \tag{14}$$

VII. Conclusion

Here we note some possible generalisations. Firstly, the suggested approach can be extended to the problem of time-series forecasting, because data compression and time-series forecasting are very similar mathematically, see [15], [25]. Besides, the suggested approach can be applied to lossy compression and source coding with a fidelity criterion.

VIII. Appendix

Proof of Theorem 1. Define $\lambda_i = \lim_{t\to\infty} |\varphi_i(x_1...x_t)|/t$, and let $\lambda_o = \min_i \lambda_i$ (these limits exist, see iii) , and

$$\lim_{t\to\infty} |\varphi_o(x_1...x_t)|/t \ = \ \lambda_o\,. \tag{15}$$

Let ϵ be any positive number. Having taken into account that the set F is finite, from these definitions we can see that there exists such t_1 that

$$|\ |\varphi_i(x_1...x_t)|/t - \lambda_i| < \epsilon \ for\, \varphi_i \in F,\, t > t_1\,. \tag{16}$$

Taking into account the property i), we can see that there exists such a number t_2 for which $\tau_i(t)$ is defined for all φ_i and $t > t_2$, and denote $t_3 = \max\{t_2, t_3\}$. Take any $n > t_3$

and suppose that a data-compressor φ_s was chosen, when $\hat{\Phi}$ was applied to $x_1x_2...x_n$. Hence, from the property ii) we can see that there exists t_4, such that

$$(-\log\omega_s+|\varphi_s(x_1...x_{t_4})|)/t_4 \le (-\log\omega_o+|\varphi_o(x_1...x_{t_4})|)/t_4\,. \quad (17)$$

From (15) we obtain the following two inequalities

$$(-\log\omega_s+|\varphi_s(x_1...x_{t_4})|)/t_4 \ge \lambda_s-\epsilon\,,$$

$$(-\log\omega_o+|\varphi_o(x_1...x_{t_4})|)/t_4 \le \lambda_0+\epsilon\,.$$

Having taken into account (17) we can see from the two latest inequalities that $\lambda_s-\epsilon<\lambda_o+\epsilon$ and, hence, $\lambda_s<\lambda_o+2\epsilon$. Taking into account, that, by definition, $\lambda_o<\lambda_s$, we obtain

$$\lambda_o<\lambda_s<\lambda_o+2\epsilon\,. \quad (18)$$

Since $n>t_1$, we can see from (16) that

$$\lambda_s-\epsilon<(-\log\omega_s+|\varphi_s(x_1...x_n)|)/n<\lambda_s+\epsilon\,.$$

Taking into account that $(-\log\omega_s+|\varphi_s(x_1...x_n)|)/n = \hat{\Phi}(x_1...x_n)/n$ we obtain from (18) that

$$\lambda_o-\epsilon<\hat{\Phi}(x_1...x_n)/n<\lambda_o+3\epsilon\,,$$

and, hence,

$$\lambda_o-\epsilon<\lim_{n\to\infty}\hat{\Phi}(x_1...x_n)/n<\lambda_o+3\epsilon\,.$$

It is true for any $\epsilon>0$, hence, $\lim_{n\to\infty}\hat{\Phi}(x_1...x_n)/n=\lambda_o$. The theorem is proven.

Proof of Theorem 2. It is known in Information Theory [21] that $h_r(\mu)\ge h_{r+1}(\mu)\ge h_\infty(\mu)$ for any r and (by definition) $\lim_{r\to\infty}h_r(\mu)=h_\infty(\mu)$. Let $\epsilon>0$ and r be such an integer that $h_r-h_\infty<\epsilon$. From (13) we can see that there exists such t_1 that $m(t)\ge r$ if $t\ge t_1$. Taking into account (10) and (13), we can see that there exists t_2 for which a.s. $||\psi_r(x_1...x_t)|/t-h_r(\mu)|<\epsilon$ if $t>t_2$. From the description of Ψ^δ (the step 3) we can see that there exists such $t_3>\max\{t_1,t_2\}$ for which a.s.

$$||\psi_r(x_1...x_t)|/t-h_\infty(\mu)|\le||\psi_r(x_1...x_t)|/t-h_r(\mu)|$$

$$+(h_r(\mu)-h_\infty(\mu))<2\epsilon\,,$$

if $t>t_3$. By definition,

$$|\Psi^\delta(x_1...x_t)|/t\le(|\psi_r(x_1...x_t)|-\log\omega_{r+1})/t.$$

Having taken into account that ϵ is an arbitrary number and two latest inequalities as well as the fact that a.s. $\inf_{i=0,1,...}\lim_{t\to\infty}|\psi_r(x_1...x_t)|/t=h_\infty(\mu)$, we obtain (14). The theorem is proven.

ACKNOWLEDGMENT

This work was supported by Russian Foundation for Basic Research (grant 18-29-03005).

REFERENCES

[1] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE transactions on Communications*, vol. 32, no. 4, pp. 396–402, 1984.

[2] J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM Journal of research and development*, vol. 23, no. 2, pp. 149–162, 1979.

[3] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on information theory*, vol. 23, no. 3, pp. 337–343, 1977.

[4] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," 1994.

[5] B. Y. Ryabko, "Data compression by means of a book stack," *Problems of Information Transmission*, vol. 16, no. 4, pp. 265–269, 1980.

[6] J. Bentley, D. Sleator, R. Tarjan, and V. Wei, " A locally adaptive data compression scheme," *Communications of the ACM*, vol. 29, no. 4, pp. 320–330, 1986.

[7] B. Ryabko, N. R. Horspool, G. V. Cormack, S. Sekar, and S. B. Ahuja, "Technical correspondence," *Communications of the ACM*, vol. 30, no. 9, pp. 792–797, 1987.

[8] J. C. Kieffer and E.-H. Yang, "Grammar-based codes: a new class of universal lossless source codes," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 737–754, 2000.

[9] E.-H. Yang and J. C. Kieffer, "Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform. i. without context models," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 755–777, 2000.

[10] M. Drmota, Yu. Reznik, and W. Szpankowski, " Tunstall code, Khodak variations, and random walks," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2928–2937, 2010.

[11] B. Ryabko, "Twice-universal coding," *Problems of Information Transmission*, vol. 3, pp. 173–177, 1984.

[12] F.M.J. Willems, Yu. Shtarkov, and T.J. Tjalkens, " The context-tree weighting method: basic properties," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[13] M. Li and P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Applications, 3rd edition*. New York: Springer, 2008.

[14] M. Hutter, *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Berlin: Springer, 2005.

[15] B. Ryabko, "Prediction of random sequences and universal coding," *Problems of Information Transmission*, vol. 24, pp. 87–96, 1988.

[16] J. J. Rissanen, "Modeling by shortest data description." *Automatica*, vol. 14, pp. 465–471, 1978.

[17] ——, *Stochastic Complexity in Statistical Inquiry*. World Scientific Publ. Co., 1989.

[18] A. Barron, J. Rissanen, and B. Yu, "The mdl principle in modeling and coding', special issue of ieee trans," *Information Theory to commemorate*, vol. 50, pp. 2743–2760, 1998.

[19] P. Grünwald, *The Minimum Description Length Principle*. The MIT Press, Cambridge, 2007.

[20] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald, "On predictive distributions and bayesian networks," *Statistics and Computing*, vol. 10, no. 1, pp. 39–54, 2000.

[21] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY, USA: Wiley-Interscience, 2006.

[22] M. Mahoney. "Data Compression Programs." http://mattmahoney.net/dc/

[23] R. Krichevsky, "A relation between the plausibility of information about a source and encoding redundancy," *Problems Inform. Transmission*, vol. 4, no. 3, pp. 48–57, 1968.

[24] ——, *Universal Compression and Retrival*. Kluwer Academic Publishers, 1993.

[25] B. Ryabko, "Compression-based methods for nonparametric prediction and estimation of some characteristics of time series," *IEEE Transactions on Information Theory*, vol. 55, pp. 4309–4315, 2009.