Random Noise Increases Kolmogorov Complexity and Hausdorff Dimension

Gleb Posobin
Computer Science department, Columbia University, New York, USA
posobin@gmail.com

Alexander Shen
LIRMM CNRS & University of Montpellier, 161 rue Ada, 34095, Montpellier, France,
On leave from IITP RAS, Moscow
https://www.lirmm.fr/~ashen
alexander.shen@lirmm.fr

Abstract

Consider a bit string x of length n and Kolmogorov complexity αn (for some $\alpha < 1$). It is always possible to increase the complexity of x by changing a small fraction of bits in x [2]. What happens with the complexity of x when we *randomly* change each bit independently with some probability τ? We prove that a linear increase in complexity happens with high probability, but this increase is smaller than in the case of arbitrary change considered in [2]. The amount of the increase depends on x (strings of the same complexity could behave differently). We give exact lower and upper bounds for this increase (with $o(n)$ precision).

The same technique is used to prove the results about the (effective Hausdorff) dimension of infinite sequences. We show that random change increases the dimension with probability 1, and provide an optimal lower bound for the dimension of the changed sequence. We also improve a result from [5] and show that for every sequence ω of dimension α there exists a strongly α-random sequence ω' such that the Besicovitch distance between ω and ω' is 0.

The proofs use the combinatorial and probabilistic reformulations of complexity statements and the technique that goes back to Ahlswede, Gács and Körner [1].

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Kolmogorov complexity, effective Hausdorff dimension, random noise

Digital Object Identifier 10.4230/LIPIcs.STACS.2019.57

Related Version An extended version of the paper is available as https://arxiv.org/abs/1808.04626.

Funding Supported by RaCAF ANR-15-CE40-0016-01 grant.
Gleb Posobin: The work was done when G. Posobin was at the Laboratory of Theoretical Computer Science, National Research University Higher School of Economics, Moscow, Russia, and was supported by Russian Academic Excellence Project 5–100 and MK-5379.2018.1 grant. The preparation of the final version was supported by NSF CAREER Award CCF-1844887 and CCF-1563155 grants.
Alexander Shen: Supported in part by RFBR 19-01-00563A grant. Part of the work done while visiting Toyota Technological University, Chicago.

Acknowledgements Authors are grateful to the participants and organizers of the Heidelberg Kolmogorov complexity program where the question of the complexity increase was raised, and to all colleagues (from the ESCAPE team, LIRMM, Montpellier, Kolmogorov seminar and HSE Theoretical Computer Science Group, and other places) who participated in the discussions, in particular to B. Bauwens, N. Greenberg, K. Makarychev, Yu. Makarychev, J. Miller, A. Milovanov, F. Nazarov, I. Razenshteyn, A. Romashchenko, N. Vereshchagin, L. B. Westrick, and, last but not least, to Peter Gács who explained us how the tools from [1] can be used to provide the desired result about Kolmogorov complexity.

36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019).
Editors: Rolf Niedermeier and Christophe Paul; Article No. 57; pp. 57:1–57:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The Kolmogorov complexity $C(x)$ of a binary string x is defined as the minimal length of a program that generates x, assuming that we use an optimal programming language that makes the complexity function minimal up to an $O(1)$ additive term (see [8, 12] for details). There are several versions of Kolmogorov complexity; we consider the original version, called *plain* complexity. In fact, for our considerations the difference between different versions of Kolmogorov complexity does not matter, since they differ only by $O(\log n)$ additive term for n-bit strings, but we restrict ourselves to plain complexity for simplicity.

The complexity of n-bit strings is between 0 and n (we omit $O(1)$ additive terms). Consider a string x of length n that has some intermediate complexity, say $0.5n$. Imagine that we are allowed to change a small fraction of bits in x, say, 1% of all bits. Can we decrease the complexity of x? Can we increase the complexity of x? What happens if we change randomly chosen 1% of bits?

In other words, consider a Hamming ball with center x and radius $0.01n$, i.e., the set of strings that differ from x in at most $0.01n$ positions. What can be said about the minimal complexity of strings in this ball? the maximal complexity of strings in this ball? the typical complexity of strings in this ball?

The answer may depend on x: different strings of the same complexity may behave differently if we are interested in the complexities of neighbor strings. For example, if the first half of x is a random string, and the second half contains only zeros, the string x has complexity $0.5n$ and it is easy to decrease its complexity by shifting the boundary between the random part and zero part: to move the boundary to $0.48n$ from $0.5n$ we need to change about $0.01n$ bits, and the complexity becomes close to $0.48n$. On the other hand, if x is a random codeword of an error-correcting code with $2^{0.5n}$ codewords of length n that corrects up to $0.01n$ errors, then x also has complexity $0.5n$, but no change of $0.01n$ (or less) bits can decrease the complexity of x, since x can be reconstructed from the changed version.

The question about the complexity *decrease* is studied by algorithmic statistics (see [14] or the survey [13]), and the full answer is known. For each x one may consider the function

$$d \mapsto (\text{the minimal complexity of strings in the } d\text{-ball centered at } x).$$

It starts at $C(x)$ (when $d = 0$) and then decreases, reaching 0 at $d = n/2$ (since we can change all bits to zeros or to ones). The algorithmic statistic tells us which functions may appear in this way (see [13, section 6.2] or [12, theorem 257]).[1]

The question about the complexity *increase* is less studied. It is known that some complexity increase is always guaranteed, as shown in [2]. The amount of this increase may depend on x. If x is a random codeword of an error-correcting code, then the changed version of x contains all the information both about x itself and the places where it was changed. This leads to the maximal increase in complexity. The minimal increase, as shown in [2], happens for x that is a random element of the Hamming ball of some radius with center 0^n. However, the natural question: which functions may appear as $d \mapsto$ (the maximal complexity of strings in the d-ball centered at x), remains open.

In our paper we study the *typical* complexity of a string that can be obtained from x by changing a fraction of bits chosen randomly. Let us return to our example and consider again a string x of length n and complexity $0.5n$. Let us change about 1% of bits in x, changing

[1] Note that algorithmic statistics uses a different language. Instead of a string y in the d-ball centered at x, it speaks about a d-ball centered at y and containing x. This ball is considered as a statistical model for x.

each bit independently[2] with probability 0.01. Does this change increase the complexity of x? It depends on the changed bits, but it turns out that *random change increases the complexity of the string with high probability*: we get a string of complexity at least $0.501n$ with probability at least 99%, for all large enough n (the result is necessarily asymptotic, since the Kolmogorov complexity function is defined up to $O(1)$ terms).

Of course, the parameters above are chosen only as an example, and the following general statement is true. For some $\tau \in (0,1)$ consider the random noise N_τ that changes each position in a given n-bit string independently with probability τ.

▶ **Theorem 1.** *There exists a strictly positive function $\delta(\alpha,\tau)$ defined for $\alpha,\tau \in (0,1)$ with the following property: for all sufficiently large n, for every $\alpha \in (0,1)$, for every $\tau \in (0,1)$, for $\beta = \alpha + \delta(\alpha,\tau)$, and for every x such that $\mathrm{C}(x) \geqslant \alpha n$, the probability of the event "$\mathrm{C}(N_\tau(x)) > \beta n$" is at least $1 - 1/n$.*

▶ Remark 2. We use the inequality $\mathrm{C}(x) \geqslant \alpha n$ (and not an equality $\mathrm{C}(x) = \alpha n$) to avoid technical problems: the complexity $\mathrm{C}(x)$ is an integer, and αn may not be an integer.

▶ Remark 3. One may consider only $\tau \leqslant 1/2$ since reversing all bits does not change Kolmogorov complexity (so τ and $1-\tau$ give the same increase in complexity). For $\tau = 1/2$ the variable $N_\tau(x)$ is uniformly distributed in the Boolean cube $\mathbb{B}^n$, so its complexity is close to n, and the statement is easy (for arbitrary $\beta < 1$).

▶ Remark 4. We use α,τ as parameters while fixing the probability bound as $1 - 1/n$. As we will see, the choice of this bound is not important: we could use a stronger bound (e.g., $1 - 1/n^d$ for arbitrary d) as well.

Now a natural question arises: what is the optimal bound in Theorem 1, i.e., the maximal possible value of $\delta(\alpha,\tau)$? In other words, fix α and τ. Theorem 1 guarantees that there exists some $\beta > \alpha$ such that every string x of length n (sufficiently large) and complexity at least αn is guaranteed to have complexity at least βn after τ-noise N_τ (with high probability). *What is the maximal value of β for which such a statement is true* (for given α and τ)?

Before answering this question, we should note that the guaranteed complexity increase depends on x: for different strings of the same complexity the typical complexity of $N_\tau(x)$ could be different. Here are the two opposite examples (with minimal and maximal increase, as we will see).

▶ **Example 5.** Consider some $p \in (0,1)$ and the Bernoulli distribution B_p on the Boolean cube $\mathbb{B}^n$ (bits are independent; every bit equals 1 with probability p). With high probability the complexity of a B_p-random string is $o(n)$-close to $n\,\mathrm{H}(p)$ (see, e.g., [12, chapter 7]), where $\mathrm{H}(p)$ is the Shannon entropy function $\mathrm{H}(p) = -p\log p - (1-p)\log(1-p)$. After applying τ-noise the distribution B_p is transformed into $B_{N(\tau,p)}$, where $N(\tau,p) = p(1-\tau) + (1-p)\tau = p + \tau - 2p\tau$ is the probability to change the bit if we first change it with probability p and then (independently) with probability τ.[3] The complexity of $N_\tau(x)$ is close (with high probability) to $H(N(\tau,p))n$ since the B_p-random string x and the τ-noise are chosen independently. So in this case we have (with high probability) the complexity increase $\mathrm{H}(p)n \to \mathrm{H}(N(\tau,p))n$. Note that $N(\tau,p)$ is closer to $1/2$ than p, and H is strictly increasing on $[0,1/2]$, so indeed some increase happens.

[2] From the probabilistic viewpoint it is more naturally to change all the bits independently with the same probability 0.01. Then the number of changed bits is not exactly $0.01n$, but is close to $0.01n$ with high probability.

[3] We use the letter N (for "noise") both in $N_\tau(x)$ (random change with probability τ, one argument) and in $N(\tau,p)$ (the parameter of the Bernoulli distribution B_p after applying N_τ, no subscript, two arguments).

▶ **Example 6.** Now consider an error-correcting code that has $2^{\alpha n}$ codewords and corrects up to τn errors (this means that the Hamming distance between codewords is greater than $2\tau n$). Such a code may exist or not depending on the choice of α and τ. The basic result in coding theory, Gilbert's bound, guarantees that such a code exists if α and τ are not too large. Consider some pair of α and τ for which such a code exist; moreover, let us assume that it corrects up to $\tau' n$ errors for some $\tau' > \tau$. We assume also that the code itself (the list of codewords) has small complexity, say, $O(\log n)$. This can be achieved by choosing the first (in some ordering) code with required parameters.

Now take a random codeword of this code; most of the codewords have complexity close to αn. If we randomly change each bit with probability τ, then with high probability we get at most $\tau' n$ errors, therefore, decoding is possible and the pair (x, noise) can be reconstructed from $N_\tau(x)$, the noisy version of x. Then the complexity of $N_\tau(x)$ is close to the complexity of the pair (x, noise), which (due to independence) is close to $\alpha n + \mathrm{H}(\tau)n$ with high probability. So in this case we have the complexity increase $\alpha n \to (\alpha + \mathrm{H}(\tau))n$.

▶ Remark 7. Note that this increase is the maximal possible not only for random independent noise but for any change in x that changes a τ-fraction of bits. See below about the difference between random change and arbitrary change.

Now we formulate the result we promised. It says that the complexity increase observed in Example 5 is the minimal possible: such an increase is guaranteed for every string of given complexity.

▶ **Theorem 8.** *Let $\alpha = \mathrm{H}(p)$ for some $p \leqslant 1/2$. Let τ be an arbitrary number in $(0, 1)$. Let $\beta = \mathrm{H}(N(p, \tau))$. Then for sufficiently large n the following is true: for every string x of length n with $\mathrm{C}(x) \geqslant \alpha n$, we have $\Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n - o(n)] \geqslant 1 - \frac{1}{n}$.*

Here $o(n)$ denotes some function such that $o(n)/n \to 0$ as $n \to \infty$. This function does not depend on α, β, and τ. As the proof will show, we may take $o(n) = c\sqrt{n}\log^{3/2} n$ for some c.

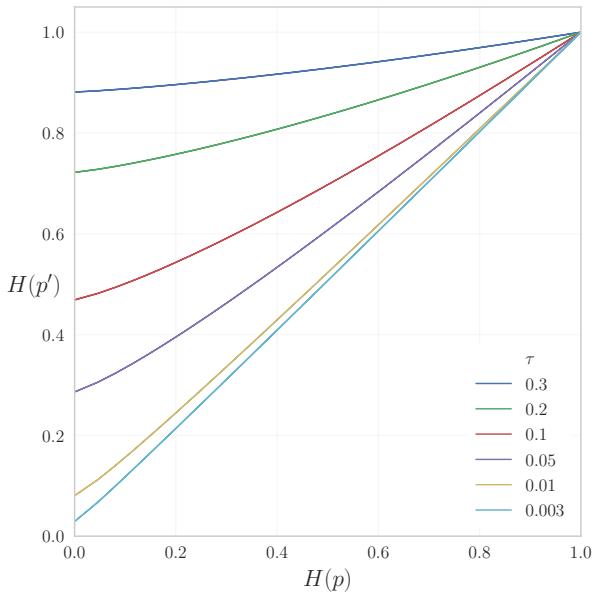

Figure 1 The curves $(\mathrm{H}(p), \mathrm{H}(p'))$ where $p' = N(p, \tau)$. Six different values of τ are shown.

Figure 1 shows the values of (α, β) where Theorem 8 can be applied, for six different values of τ. Example 5 shows that the value of β in this theorem is optimal.

In the next section we explain the scheme of the proof of Theorem 8. Then we explain the details of the proof. It starts with the Shannon information counterpart of our complexity statement that is proven in [15]. In Section 3 we derive two different combinatorial counterparts following [1]. Finally, in Section 4 we consider the details of the conversion of a combinatorial statement to a complexity one and finish the proof. In Section 5 we extend our techniques to infinite sequences and improve some results obtained in [5].

2 Proof sketch

2.1 Three ways to measure the amount of information

Kolmogorov's first paper on algorithmic information theory [7] was called "Three approaches to the Quantitative Definition of Information". These three approaches can be summarized as follows:

- (Combinatorial): an element of a set of cardinality N carries $\log N$ bits of information.
- (Algorithmic): a binary string x carries $\mathrm{C}(x)$ bits of information, where $\mathrm{C}(x)$ is the minimal bit length of a program that produces x.
- (Shannon information theory, or probabilistic approach): a random variable ξ that has k values with probabilities $p_1, \ldots, p_k$, carries $\mathrm{H}(\xi)$ bits of information, where $\mathrm{H}(\xi)$ is the Shannon entropy of ξ, defined as $\mathrm{H}(\xi) = p_1 \log \frac{1}{p_1} + \ldots + p_k \log \frac{1}{p_k}$

One cannot compare directly these three quantities since the measured objects are different (sets, strings, random variables). Still these quantities are closely related, and many statements that are true for one of these notions can be reformulated for other two. Several examples of this type are discussed in [12, chapters 7 and 10], and we use this technique in our proof.

2.2 Arbitrary change

We start by recalling an argument from [2] for the case when we are allowed to change arbitrary bits (only the number of changed bits is bounded) and want to increase complexity. (A similar reduction will be a part of our argument.)

Fix some parameters α (determining the complexity of the original string), τ (the maximal fraction of changed bits), and β (determining the complexity of the changed string). Let us repeat the complexity statement and give its combinatorial equivalent.

- (Complexity version) Every string x of length n and complexity at least αn can be changed in at most τn positions to obtain a string of complexity at least βn.
- (Combinatorial version) For every subset B of the Boolean cube $\mathbb{B}^n$ of cardinality at most $2^{\beta n}$, its τn-interior has cardinality at most $2^{\alpha n}$.

Here by d-interior of a set $X \subset \mathbb{B}^n$ we mean the set of strings $x \in \mathbb{B}^n$ such that the entire ball of radius d centered at x belongs to X. In other words, a string x does *not* belong to the d-interior of X if x can be changed in at most d positions to get a string outside X.

▶ Remark 9. The combinatorial statement can be reformulated in a dual way: for every set $A \subset \mathbb{B}^n$ of cardinality greater than $2^{\alpha n}$, its d-neighborhood has cardinality greater than $2^{\beta n}$.

These two statements (combinatorial and complexity versions) are almost equivalent: one of them implies the other if we allow a small change in α and β (in fact, $O(\log n)/n$ change is enough). Indeed, assume first that the combinatorial statement is true. Consider the set B of all n-bit strings that have complexity less than βn. Then $\#B < 2^{\beta n}$, so we can

apply the combinatorial statement.[4] It guarantees that the τn-interior of B (we denote it by A) has at most $2^{\alpha n}$ elements. The set A can be enumerated given n, βn and τn. Indeed, knowing n and βn, one can enumerate elements of B (by running in parallel all programs of length less than βn; note that there are less than $2^{\beta n}$ of them). Knowing also τn, we may enumerate A (if a ball is contained in B entirely, this will become known at some stage of the enumeration of B). Then every element of A can be encoded by its ordinal number in this enumeration. This guarantees that the complexity of all elements of A does not exceed $\alpha n + O(\log n)$ (the additional $O(\log n)$ bits are needed to encode n, βn, and τn). Therefore, if some x has complexity greater that $\alpha n + O(\log n)$, it is not in A, i.e., x can be changed in at most τn positions to get a string outside B. By the definition of B, this changed string has complexity at least βn, as required. The term $O(\log n)$ can be absorbed by a small change in α.

Let us explain also (though this direction is not needed for our purpose) why the complexity statement implies the combinatorial one. Assume that there are some sets B that violate the combinatorial statement, i.e., contain at most $2^{\beta n}$ strings but have τn-interior of size greater than $2^{\alpha n}$. Such a set can be found by exhaustive search, and the first set B that appears during the search has complexity $O(\log n)$. Its elements, therefore, have complexity $\beta n + O(\log n)$: to specify an element, we need to specify B and the ordinal number of the element in B. From this we conclude, using the complexity statement (and changing β slightly) that all elements of the τn-interior of B have complexity at most αn. Therefore, there are at most $O(2^{\alpha n})$ of them, and the size of the interior is bounded by $2^{\alpha n}$ (again up to a small change in α).

Now we return to the result from [2]. Let x be a string of length n and complexity at least $\alpha n + O(\log n)$, where $\alpha = \mathrm{H}(p)$ for some $p \leqslant 1/2$. Let τ be a real such that $p + \tau \leqslant 1/2$, and $\beta = \mathrm{H}(p+\tau)$. Then x can be changed in at most τn positions to get a string of complexity at least βn. As we have seen, this statement from [2] is a corollary of the following combinatorial result.

▶ **Proposition 10.** *Let $p \leqslant 1/2$ be some number and let $\alpha = \mathrm{H}(p)$. Let τ be some positive number so that $p + \tau \leqslant 1/2$, and let $\beta = \mathrm{H}(p + \tau)$. Let B be an arbitrary subset of $\mathbb{B}^n$ of size at most $2^{\beta n}$. Let A be a subset of $\mathbb{B}^n$, and for every $x \in A$ the Hamming ball of radius τn with center x is contained in B. Then the cardinality of A does not exceed* $\mathrm{poly}(n)2^{\alpha n}$.

This proposition is a direct consequence of Harper's theorem (see, e.g., [4]) that says that for a subset of $\mathbb{B}^n$ of a given size, its *d-interior* (for some fixed d) *is maximal when the subset is a Hamming ball* (formally speaking, is between two Hamming balls of sizes k and $k+1$ for some k). Or, in dual terms, Harper's theorem says that the *d-neighborhood of a set of a given size is minimal if this set is a Hamming ball.* The relation between $2^{\alpha n}$ and $2^{\beta n}$ in the proposition is just the relation between the sizes of balls of radii pn and $(p+\tau)n$ (if we ignore factors that are polynomial in n). Note that $p + \tau \leqslant 1/2$ is needed since otherwise the radius exceeds $n/2$ and then the log-size of the ball is close to n and not to $\mathrm{H}(p+\tau)n$. The $\mathrm{poly}(n)$ factor is needed due to the polynomial factor in the estimate for the ball size in terms of Shannon entropy (the ball of radius γn has size $\mathrm{poly}(n)2^{\mathrm{H}(\gamma)n}$).

We do not go into details here (and do not reproduce the proof of Harper's theorem) since we need this result only to motivate the corresponding relation between combinatorial and complexity statements for the case of a random noise we are interested in.

[4] For simplicity we assume that αn, βn, and τn are integers. This is not important, since we have $O(\log n)$ term anyway.

2.3 Random noise: four versions

For the random noise case we need a more complicated argument. First, we need to consider also the probabilistic version of the statement (in addition to the complexity and combinatorial versions). Second, we need *two* combinatorial versions (strong and weak). Fix some α, β and τ. Here are the four versions we are speaking about; all four statements are equivalent (are true for the same parameters α, τ, and β, up to $o(1)$-changes in the parameters):

- (Shannon information version, [15]) For every random variable P with values in $\mathbb{B}^n$ such that $\mathrm{H}(P) \geqslant \alpha n$, the variable $N_\tau(P)$ that is obtained from P by applying independent noise changing each bit with probability τ, has entropy $\mathrm{H}(N_\tau(P)) \geqslant \beta n$.
- (Complexity version) For every string x of length n and complexity $\mathrm{C}(x) \geqslant \alpha n$, the probability of the event "$\mathrm{C}(N_\tau(x)) \geqslant \beta n$" is at least $1 - 1/n$. (Again, N_τ is random noise that independently changes each bit with probability τ, but now it is applied to the string x and not to a random variable)
- (Strong combinatorial version) For every set $B \subset \mathbb{B}^n$ of size at most $2^{\beta n}$ the set A of all strings x such that $\Pr[N_\tau(x) \in B] \geqslant 1/n$ has size $\#A \leqslant 2^{\alpha n}$.
- (Weak combinatorial version) For every set $B \subset \mathbb{B}^n$ of size at most $2^{\beta n}$ the set A of all strings x such that $\Pr[N_\tau(x) \in B] \geqslant 1 - 1/n$ has size $\#A \leqslant 2^{\alpha n}$.

The difference between weak and strong combinatorial versions is due to the different thresholds for the probability. In the weak version the set A contains only strings that get into B after the noise *almost surely* (with probability at least $1 - 1/n$). In the strong version the set A is bigger and includes all strings that get into B *with non-negligible probability* (at least $1/n$), so the upper bound for $\#A$ becomes a stronger statement.

▶ Remark 11. In the case of arbitrary changes (the result from [2]) we consider the τn-interior of B, the set of points that remain in B after arbitrary change in (at most) τn positions. If a point is *not* in the interior, it can be moved outside B by changing at most τn bits. Now we consider (in the strong version) the set of points that get into B with probability at least $1/n$. If a point is *not* in this set, the random τ-noise will move it outside B almost surely (with probability at least $1 - 1/n$). Again the complexity and (strong) combinatorial versions are equivalent up to $o(1)$ changes in parameters, for the same reasons.

This explains why we are interested in the strong combinatorial statement. The weak one is used as an intermediate step in the chain of arguments. This chain goes as follows:

- First the Shannon entropy statement is proven using tools from information theory (one-letter characterization and inequalities for Shannon entropy); this was done in [15].
- Then we derive the weak combinatorial statement from the entropy statement using a simple coding argument from [1].
- Then we show that weak combinatorial statement implies the strong one, using a tool that is called the "blowing-up lemma" in [1] (now it is more popular under the name of "concentration inequalities").
- Finally, we note that the strong combinatorial statement implies the complexity statement (using the argument sketched above).

2.4 Tools used in the proof

Let us give a brief description of the tools used in these arguments.

To prove the Shannon entropy statement, following [15], fix some τ. Consider the set S_n of all pairs $(\mathrm{H}(P), \mathrm{H}(N_\tau(P)))$ for all random variables with values in $\mathbb{B}^n$. For each n we get a subset of the square $[0, n] \times [0, n]$. For $n = 1$ it is a curve made of all points

$(\mathrm{H}(p), \mathrm{H}(N(p,\tau)))$ (shown in Figure 1 for six different values of τ). We start by showing that this curve is convex (performing some computation with power series). Then we show, using the convexity of the curve and some inequalities for entropies, that for every n the set S_n is above the same curve (scaled by factor n), and this is the entropy statement we need. (See the arxiv version of this paper for details.)

To derive the weak combinatorial statement from the entropy statement, we use a coding argument. Assume that two sets A and B are given, and for every point $x \in A$ the random point $N_\tau(x)$ belongs to B with probability at least $1 - 1/n$. Consider a random variable U_A that is uniformly distributed in A. Then $\mathrm{H}(U_A) = \log \#A$, and if $\#A \geqslant 2^{\alpha n}$, then $\mathrm{H}(U_A) \geqslant \alpha n$ and $\mathrm{H}(N_\tau(U_A)) \geqslant \beta n$ (assuming the entropy statement is true for given α, β, and τ). On the other hand, the variable $N_\tau(U_A)$ can be encoded as follows:

- one bit (flag) says whether $N_\tau(U_A)$ is in B or not;
- if yes, then $\log \#B$ bits are used to encode an element of B;
- otherwise n bits are used to encode the value of $N_\tau(U_A)$ (trivial encoding).

The average length of this code for $N_\tau(U_A)$) does not exceed

$$1 + \left(1 - \frac{1}{n}\right) \log \#B + \frac{1}{n} \cdot n \leqslant \log \#B + O(1).$$

(Note that if the second case has probability less than $1/n$, the average length is even smaller.) The entropy of a random variable $N_\tau(U_A)$ does not exceed the average length of the code. So we get $\beta n \leqslant \mathrm{H}(N_\tau(U_A)) \leqslant \log \#B + O(1)$ and $\log \#B \geqslant \beta n - O(1)$, assuming that $\log \#A \geqslant \alpha n$.

The next step is to derive the strong combinatorial version from the weak one. Assume that two sets $A, B \subset \mathbb{B}^n$ are given, and for each $x \in A$ the probability of the event $N_\tau(x) \in B$ is at least $1/n$. For some d consider the set B_d, the d-neighborhood of B. We will prove (using the concentration inequalities) that for some $d = o(n)$ the probability of the event $N_\tau(x) \in B_d$ is at least $1 - 1/n$ (for each $x \in A$). So one can apply the weak combinatorial statement to B_d and get a lower bound for $\#B_d$. On the other hand, there is a simple upper bound: $\#B_d \leqslant \#B \times$ (the size of d-ball); combining them, we get the required bound for $\#B$. See Section 3 for details.

▶ Remark 12. One may also note (though it is not needed for our purposes) that the entropy statement is an easy corollary of the complexity statement, and therefore all four are equivalent up to small changes in parameters. This can be proven in a standard way. Consider N independent copies of random variable P and independently apply noise to all of them. Then we write the inequality for the typical values of the complexities; in most cases they are close to the corresponding entropies (up to $o(N)$ error). Therefore, we get the inequality for entropies with $o(N)$ precision (for N copies) and with $o(1)$ precision for one copy (the entropies are divided by N). As $N \to \infty$, the additional term $o(1)$ disappears and we get an exact inequality for entropies.

3 Combinatorial version

Recall the entropy bound from [15] discussed above (we reproduce its proof in the arxiv version for reader's convenience):

▶ **Proposition 13.** *Let P be an arbitrary random variable with values in $\mathbb{B}^n$, and let $P' = N_\tau(P)$ be its noisy version obtained by applying N_τ independently to each bit in P. Choose $p \leqslant 1/2$ in such a way that $\mathrm{H}(P) = n\,\mathrm{H}(p)$. Then consider $q = N(p,\tau)$, the probability to get 1 if we apply N_τ to a variable that equals 1 with probability p. Then $\mathrm{H}(P') \geqslant n\,\mathrm{H}(q)$.*

In this section we use this entropy bound to prove the combinatorial bounds. We start with the weak one and then amplify it to get the strong one, as discussed in Section 2. First, let us formulate explicitly the weak bound that is derived from Proposition 13 using the argument of Section 2.

▶ **Proposition 14.** *Let* $\alpha = \mathrm{H}(p)$ *and* $\beta = \mathrm{H}(N(p,\tau))$. *Let* $A, B \subset \mathbb{B}^n$ *and for every* $x \in A$ *the probability of the event "*$N_\tau(x) \in B$*" is at least* $1 - 1/n$. *If* $\log \#A \geqslant \alpha n$, *then* $\log \#B \geqslant \beta n - O(1)$.

In fact, this "$O(1)$" is just 2, but we do not want to be too specific here.

Now we need to extend the bound to the case when the probability of the event $N_\tau(x) \in B$ is at least $1/n$. We already discussed how this is done. Consider for some d (depending on n) the Hamming d-neighborhood B_d of B. We need to show that

$$\Pr[N_\tau(x) \in B] \geqslant \frac{1}{n} \Rightarrow \Pr[N_\tau(x) \in B_d] \geqslant 1 - \frac{1}{n}.$$

for every $x \in \mathbb{B}^n$ (for a suitable d). In fact, x does not matter here: we may assume that $x = 0\ldots0$ (flipping bits in x and B simultaneously). In other terms, we use the following property of the Bernoulli distribution with parameter τ: *if some set* B *has probability not too small according to this distribution, then its neighborhood* B_d *has probability close to* 1. We need this property for $d = o(n)$, see below about the exact value of d.

Such a statement is called a *blowing-up lemma* in [1]. There are several (and quite different) ways to prove statements of this type. The original proof in [1] used a result of Margulis from [9] that says that the (Bernoulli) measure of a boundary of an arbitrary set $U \subset \mathbb{B}^n$ is not too small compared to the measure of a boundary of a ball of the same size. Iterating this statement (a neighborhood is obtained by adding boundary layer several times), we get the lower bound for the measure of the neighborhood. Another proof was suggested by Marton [10]; it is based on the information-theoretical considerations that involve transportation cost inequalities for bounding measure concentration. In this paper we provide a simple proof that uses the McDiarmid inequality [11], a simple consequence of the Azuma–Hoeffding inequality [6]. This proof works for $d = O(\sqrt{n \log n})$.

Let us state the blowing-up lemma in a slightly more general version than we need. Let $X_1, \ldots, X_n$ be (finite) probability spaces. Consider the space $X = X_1 \times \ldots \times X_n$ with the product measure μ (so the coordinates are independent) and Hamming distance d (the number of coordinates that differ). In our case $X = \mathbb{B}^n$ and μ is the Bernoulli measure with parameter τ. The blowing-up lemma says, informally speaking, that if a set is not too small, then its neighborhood has small complement (the size is measured by μ). It can be reformulated in a more symmetric way: *if two sets are not too small, then the distance between them is rather small.* (Then this symmetric statement is applied to the original set and the complement of its neighborhood.) Here is the symmetric statement:

▶ **Proposition 15** (Blowing-up lemma, symmetric version)**.** *Let* B, B' *be two subsets of* $X = X_1 \times \ldots \times X_n$ *with the product measure* μ. *Then*

$$d(B, B') \leqslant \sqrt{(n/2)\ln(1/\mu(B))} + \sqrt{(n/2)\ln(1/\mu(B'))}.$$

To prove the blowing-up lemma, we use the McDiarmid concentration inequality:

▶ **Proposition 16** (McDiarmid's inequality, [11])**.** *Consider a function* $f\colon X_1 \times \ldots \times X_n \to \mathbb{R}$. *Assume that changing the* i*-th coordinate changes the value of* f *at most by some* c_i*:* $|f(x) - f(x')| \leqslant c_i$, *if* x *and* x' *coincide everywhere except for the* i*th coordinate. Then*

$$\Pr[f - \mathbb{E}\, f \geqslant z] \leqslant \exp\left(-\frac{2z^2}{\sum_{i=1}^n c_i^2}\right)$$

for arbitrary $z \geqslant 0$.

Here the probability and expectation are considered with respect to the product distribution μ (the same as in the blowing-up lemma, see above). This inequality shows that f cannot be much larger than its average on a big set. Applying this inequality to $-f$, we get the same bound for the points where the function is less than its average by z or more. (For the reader's convenience we reproduce the proof of the McDiarmid inequality in the arxiv version of this paper.)

Now let us show why it implies the blowing-up lemma (in the symmetric version).

Proof of the blowing-up lemma. Let $f(x) = d(x, B)$ be the distance between x and B, i.e., the minimal number of coordinates that one has to change in x to get into B. This function satisfies the bounded differences property with $c_i = 1$, so we can apply the McDiarmid inequality to it. Let m be the expectation of f. The function f equals zero for arguments in B and therefore is below its expectation at least by m (everywhere in B), so

$$\mu(B) \leqslant \exp\left(-\frac{2m^2}{n}\right), \quad \text{or} \quad m \leqslant \sqrt{(n/2)\ln(1/\mu(B))}$$

On the other hand, the function f is at least $d(B, B')$ for arguments in B', so it exceeds its expectation at least by $d(B, B') - m$ (everywhere in B'), therefore the McDiarmid inequality gives

$$d(B, B') - m \leqslant \sqrt{(n/2)\ln(1/\mu(B'))},$$

and it remains to combine the last two inequalities. ◀

Here is the special case of the blowing-up lemma we need:

▶ **Corollary 17.** *If μ is a distribution on $\mathbb{B}^n$ with independent coordinates, and $B \subset \mathbb{B}^n$ has measure $\mu(B) \geqslant 1/n$, then for $d = O(\sqrt{n \log n})$ we have $\mu(B_d) \geqslant 1 - 1/n$.*

Indeed, we may apply the blowing-up lemma to B and B', where B' is a complement of B_d. If both B and B' have measures at least $1/n$, we get a contradiction for $d \geqslant 2\sqrt{(n/2)\ln n}$ (the distance between B and the complement of its neighborhood B_d exceeds d).

▶ Remark 18. In the same way we get a similar result for probabilities $1/n^c$ and $1 - 1/n^c$ for arbitrary constant c (only the constant factor in $O(\sqrt{n \log n})$ will be different).

Now we are ready to prove the strong combinatorial version:

▶ **Proposition 19.** *Let $\alpha = \mathrm{H}(p)$ and $\beta = \mathrm{H}(N(p, \tau))$. Let $A, B \subset \mathbb{B}^n$ and for every $x \in A$ the probability of the event "$N_\tau(x) \in B$" is at least $1/n$. If $\log \#A \geqslant \alpha n$, then $\log \#B \geqslant \beta n - O(\sqrt{n} \log^{3/2} n)$.*

Proof. As we have seen, the weak combinatorial version (Proposition 14) can be applied to the neighborhood B_d for $d = O(\sqrt{n \log n})$. The size of B_d can be bounded by the size of B multiplied by the size of a Hamming ball of radius d. The latter is $\mathrm{poly}(n) 2^{n\,\mathrm{H}(d/n)}$. Combining the inequalities, we get

$$\log \#B \geqslant \log \#B_d - n\,\mathrm{H}(d/n) - O(\log n) \geqslant \beta n - n\,\mathrm{H}(d/n) - O(\log n).$$

For small p we have $\mathrm{H}(p) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p} = p \log \frac{1}{p} + p + o(p) = O\left(p \log \frac{1}{p}\right)$. We have $p = d/n = O(\sqrt{\log n / n})$, so $n\,\mathrm{H}(d/n) = nO(\sqrt{\log n / n} \log n) = O(\sqrt{n} \log^{3/2} n)$, as promised. ◀

4 Complexity statement

Now we combine all pieces and prove Theorem 8. It states:

> Let $\alpha = \mathrm{H}(p)$ for some $p \leqslant 1/2$. Let τ be an arbitrary number in $(0,1)$. Let $\beta = \mathrm{H}(N(p,\tau))$. Then for sufficiently large n the following is true: for every string x of length n with $\mathrm{C}(x) \geqslant \alpha n$, we have $\Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n - o(n)] \geqslant 1 - \frac{1}{n}$.

Here $o(n)$ is actually $O(\sqrt{n}\log^{3/2} n)$.

We already have all the necessary tools for the proof, but some adjustments are needed. We already know how to convert a combinatorial statement into a complexity one. For that we consider the set B of all strings in $\mathbb{B}^n$ that have complexity less than $\beta n - c\sqrt{n}\log^{3/2} n$ for some c (to be chosen later). Then we consider the set A of all x such that $\Pr[N_\tau(x) \in B] \geqslant 1/n$. The combinatorial statement (strong version, Proposition 19) guarantees that $\#A \leqslant 2^{\alpha n}$. We would like to conclude that all elements of A have complexity only slightly exceeding αn. (Then we have to deal with this excess, see below.) For that we need an algorithm that enumerates A. First, we need to enumerate B, and for that it is enough to know n and the complexity bound for elements of B. But now (unlike the case of arbitrary change where we need to know only the maximal number of allowed changes) we need to compute the probability $\Pr[N_\tau(x) \in B]$, and the value of τ may not be computable, and an infinite amount of information is needed to specify τ. How can we overcome this difficulty?

Note that it is enough to enumerate some set that contains A but has only slightly larger size. Consider some rational τ' that is close to τ and the set $A' = \{x\colon \Pr[N_{\tau'}(x) \in B] > 1/2n\}$ The combinatorial statement remains true (as we noted in Remark 18, even $1/n^c$ would be OK, not only $1/2n$), so we may still assume that $\#A' \leqslant 2^{\alpha n}$. We want $A' \supset A$. This will be guaranteed if the difference between $\Pr[N_\tau(x) \in B]$ and $\Pr[N_{\tau'}(x) \in B]$ is less than $1/2n$. To use the coupling argument, let us assume that $N_\tau(x)$ and $N_{\tau'}(x)$ are defined on the same space: to decide whether the noise changes ith bit, we generate a fresh uniformly random real in $[0,1]$ and compare it with thresholds τ and τ'. This comparison gives different results if this random real falls into the gap between τ and τ'. Using the union bound for all bits, we conclude that $\Pr[N_\tau(x) \neq N_{\tau'}(x)]$ in this setting is bounded by $n|\tau' - \tau|$. Therefore, if the approximation error $|\tau' - \tau|$ is less than $1/2n^2$, we get the desired result, and to specify τ' that approximates τ with this precision we need only $O(\log n)$ bits. This gives us the following statement:

> for every string x of length n with $\mathrm{C}(x) \geqslant \alpha n + O(\log n)$, we have $\Pr[\mathrm{C}(N_\tau(x)) \geqslant \beta n - o(n)] \geqslant 1 - \frac{1}{n}$.

The only difference with the statement of Theorem 8 is that we have a stronger requirement $\mathrm{C}(x) \geqslant \alpha n + O(\log n)$ instead of $\mathrm{C}(x) \geqslant \alpha n$. To compensate for this, we need to decrease α a bit and apply the statement we have proven to $\alpha' = \alpha - O(\log n/n)$. Then the corresponding value of β also should be changed, to get a point (α', β') on the curve (Figure 1) on the left of the original point (α, β). Note that the slope of the curve is bounded by 1 (it is the case at the right end where the curve reaches $(1,1)$, since the curve is above the diagonal $\alpha = \beta$, and the slope increases with α due to convexity). Therefore, the difference between β and β' is also $O(\log n/n)$ and is absorbed by the bigger term $O(\sqrt{n}\log^{3/2} n)$.

Theorem 8 is proven.

In the next section we apply our technique to get some related results about infinite bit sequences and their effective Hausdorff dimension. We finish the part about finite strings with the following natural question.

▶ **Question 1.** *Fix some x and apply random noise N_τ. The complexity of $N_\tau(x)$ becomes a random variable. What is the distribution of this variable? The blowing-up lemma implies that it is concentrated around some value. Indeed, if we look at strings below 1%-quantile and above 99%-quantile, the blowing-up lemma guarantees that the Hamming distance between these two sets is at most $O(\sqrt{n})$, and therefore the thresholds for Kolmogorov complexity differ at most by $O(\sqrt{n}\log n)$ (recall that for two strings of length n that differ in i positions, their complexities differ at most by $O(i \log n)$, since it is enough to add information about i positions and each position can be encoded by $\log n$ bits).*

So with high probability the complexity of $N_\tau(x)$ is concentrated around some value (defined up to $O(\sqrt{n}\log n)$ precision). For each τ we get some number (expected complexity, with guaranteed concentration) that depends not only on n and $\mathrm{C}(x)$, but on some more specific properties of x. What are these properties? Among the properties of this type there exists a Vitanyi–Vereshchagin profile curve for balls, the minimal *complexity in the neighborhood as function of the radius (see [12, section 14.4]); is it somehow related?*

As we have mentioned, this question is open also for maximal *complexity in d-balls around x, not only for* typical *complexity after τ-noise.*

5 Infinite sequences and Hausdorff dimension

Let $X = x_1x_2x_3\ldots$ be an infinite bit sequence. The effective Hausdorff dimension of X is defined as $\liminf_{n\to\infty}(\mathrm{C}(x_1\ldots x_n)/n)$. A natural question arises: *what happens with the Hausdorff dimension of a sequence when each its bit is independently changed with some probability τ?* The following result states that the dimension increases with probability 1 (assuming the dimension was less than 1, of course), and the guaranteed increase follows the same curve as for finite sequences.

▶ **Theorem 20.** *Let $p, \tau \in (0, 1/2)$ be some reals, $\alpha = \mathrm{H}(p)$ and $\beta = \mathrm{H}(N(p, \tau))$. Let X be an infinite sequence that has effective Hausdorff dimension at least α. Then the effective Hausdorff dimension of the sequence $N_\tau(X)$ that is obtained from X by applying random τ-noise independently to each position, is at least β with probability* 1.

Proof. It is enough to show, for every $\beta' < \beta$, that the dimension of $N_\tau(X)$ is at least β' with probability 1. Consider $\alpha' < \alpha$ so that the pair (α', β') lies on the boundary curve. By definition of the effective Hausdorff dimension, we know that $\mathrm{C}(x_1\ldots x_n) > \alpha' n$ for all sufficiently large n. Then Theorem 8 can be applied to α' and β'. It guarantees that with probability at least $1 - 1/n$ the changed string has complexity at least $\beta' n - o(n)$. Moreover, as we have said, the same is true with probability at least $1 - 1/n^2$. This improvement is important for us: the series $\sum 1/n^2$ converges, so the Borel–Cantelli lemma says that with probability 1 only finitely many prefixes have complexity less than $\beta' n - o(n)$, therefore the dimension of $N_\tau(X)$ is at least β' with probability 1. ◀

In the next result we randomly change bits with probabilities depending on the bit position. The probability of change in the nth position converges to 0 as $n \to \infty$. This guarantees that with probability 1 we get a sequence that is Besicovitch-close to a given one. Recall that the Besicovitch distance between two bit sequences $X = x_1x_2\ldots$ and $Y = y_1y_2\ldots$ is defined as $\limsup_{n\to\infty}(d(x_1\ldots x_n, y_1\ldots y_n)/n)$, where d stands for the Hamming distance. So $d(X, Y) = 0$ means that the fraction of different bits in the n-bit prefixes of two sequences converges to 0 as $n \to \infty$. The strong law of large numbers implies that if we start with some sequence X and change ith bit independently with probability τ_i with $\lim_n \tau_n = 0$, we get (with probability 1) the sequence X' such that the Besicovitch distance between X and X' is 0. This allows us to prove the following result using a probabilistic argument.

▶ **Theorem 21.** *Let $X = x_1x_2\ldots$ be a bit sequence whose effective Hausdorff dimension is at least γ for some $\gamma < 1$. Let δ_n be a sequence of positive reals such that $\lim_n \delta_n = 0$. Then there exists a sequence $X' = x'_1x'_2\ldots$ such that:*

- *the Besicovitch distance between X and X' is 0;*
- *$\mathrm{C}(x'_1\ldots x'_n)$ is at least $n(\gamma+\delta_n)$ for all sufficiently large n.*

Proof. For this result we use some decreasing sequence $\tau_i \to 0$ and change ith bit with probability τ_i. Since $\tau_i \to 0$, with probability 1 the changed sequence is Besicovitch-equivalent (distance 0) to the original one. It remains to prove that the probability of the last claim (the lower bound for complexities) is also 1 for the changed sequence, if we choose $\tau_i \to 0$ in a suitable way.

To use different τ_i for different i, we have to look again at our arguments. We start with Proposition 13: the proof remains valid if each bit is changed independently with probability $\tau_i \geqslant \tau$ depending on the bit's position (see the arxiv version of the paper for details). Indeed, for every $\tau' \geqslant \tau$ the corresponding τ'-curve is above the τ-curve, so the pairs of entropies (original bit, bit with noise) are above the τ-curve and we may apply the same convexity argument.

The derivation of the combinatorial statement (first the weak one, then the strong one) also remains unchanged. The proof of the weak version does not mention the exact nature of the noise at all; in the strong version we use only that different bits are independent (to apply the McDiarmid inequality and the blowing-up lemma). The only problem arises when we derive the complexity version from the combinatorial one. In our argument we need to know τ (or some approximation for τ) to enumerate A. If for each bit we have its own value of τ, even one bit to specify this value is too much for us.

To overcome this difficulty, let us agree that we start with $\tau_i = 1/2$, then change them to $1/4$ at some point, then to $1/8$ etc. If for nth bit we use $\tau_n = 2^{-m}$, then to specify all the τ_i for $i \leqslant n$ we need to specify $O(m\log n)$ bits (each moment of change requires $O(\log n)$ bits). For $\tau = 2^{-m}$ we choose a pair (α,β) on the τ-curve such that $\alpha < \gamma < \beta$. To decide when we can start using this value of τ, we wait until $\mathrm{C}(x_1\ldots x_n) > \alpha n + O(m\log n)$ becomes true and stays true forever, and also $\gamma + \delta_n < \beta - O(\sqrt{n}\log^{3/2} n)$ becomes and stays true. Note that m is fixed when we decide when to start using $\tau = 2^{-m}$, so such an n can be found. In this way we guarantee that the probability that $x'_1\ldots x'_n$ will have complexity more than $(\gamma+\delta_n)$ is at least $1 - 1/n^2$ (we need a converging series, so we use the bound with n^2), and it remains to apply the Borel–Cantelli lemma. ◀

Theorem 21 implies that for every X that has effective Hausdorff dimension α there exist a Besicovitch equivalent X' that is α-random (due to the complexity criterion for α-randomness, see [5]), and we get the result of [5, Theorem 2.5] as a corollary. Moreover, we can get this result in a stronger version than in [5], since for slowly converging sequence δ_n, for example, $\delta_n = 1/\log n$, we get *strong* α-randomness instead of *weak* α-randomness used in [5]. (For the definition of weak and strong α-randomness and for the complexity criteria for them see [3, Section 13.5].)

Final remarks

In fact, if we are interested only in *some* increase of entropy when applying noise, and do not insist on the optimal lower bound, some simpler arguments (that do not involve entropy arguments and just prove the combinatorial statement with a weaker bound) are enough. One of them uses Fourier transform and was suggested by Fedor Nazarov; one can also use the hypercontractivity argument to improve this bound, but the resulting bound is not optimal either. Both arguments are explained in the arxiv version of the paper.

The arxiv version also contains the proof of the result from [1] (about the increase in entropy caused by random noise) for reader's convenience, and also because we need a slightly more general version for non-constant noise probability. Short (and quite standard) proofs of the McDiarmid inequality as a corollary of the Azuma–Hoeffding inequality, and of the Azuma–Hoeffding inequality itself are also given there to make the paper self-contained.

References

1 Rudolf Ahlswede, Peter Gács, and János Körner. Bounds on conditional probabilities with applications in multi-user communication. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 34(2):157–177, 1976.

2 Harry Buhrman, Lance Fortnow, Ilan Newman, and Nikolai K. Vereshchagin. Increasing Kolmogorov Complexity. In Volker Diekert and Bruno Durand, editors, *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, volume 3404 of *Lecture Notes in Computer Science*, pages 412–421. Springer, 2005. doi:10.1007/978-3-540-31856-9_34.

3 Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Theory and Applications of Computability. Springer, 2010. doi:10.1007/978-0-387-68441-3.

4 Peter Frankl and Zoltán Füredi. A short proof for a theorem of Harper about Hamming-spheres. *Discrete Mathematics*, 34(3):311–313, 1981. doi:10.1016/0012-365X(81)90009-1.

5 Noam Greenberg, Joseph S. Miller, Alexander Shen, and Linda Brown Westrick. Dimension 1 sequences are close to randoms. *Theoretical Computer Science*, 705:99–112, 2018. doi:10.1016/j.tcs.2017.09.031.

6 Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

7 Andrei N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):3–11, 1965.

8 Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, Third Edition*. Texts in Computer Science. Springer, 2008. doi:10.1007/978-0-387-49820-1.

9 Grigory A. Margulis. Probabilistic properties of highly connected graphs. *Problems of Information Transmission*, 10(2):174–179, 1974.

10 Katalin Marton. A simple proof of the blowing-up lemma. *IEEE Transactions on Information Theory*, 32(3):445–446, 1986. doi:10.1109/TIT.1986.1057176.

11 Colin McDiarmid. *On the method of bounded differences*, pages 148–188. London Mathematical Society Lecture Note Series. Cambridge University Press, 1989. doi:10.1017/CBO9781107359949.008.

12 Alexander Shen, Vladimir A. Uspensky, and Nikolay Vereshchagin. *Kolmogorov complexity and algorithmic randomness*, volume 220. American Mathematical Society, 2017.

13 Nikolai K. Vereshchagin and Alexander Shen. Algorithmic statistics: forty years later. In *Computability and Complexity. Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, v. 10010*, pages 669–737. Springer, July 2017. arXiv:1607.08077.

14 Nikolai K. Vereshchagin and Paul M. B. Vitányi. Rate Distortion and Denoising of Individual Data Using Kolmogorov Complexity. *IEEE Transactions on Information Theory*, 56(7):3438–3454, 2010.

15 Aaron D. Wyner and Jacob Ziv. A Theorem on the Entropy of Certain Binary Sequences and Applications: Part I. *IEEE Transactions on Information Theory*, 19(6):769–772, 1973. doi:10.1109/TIT.1973.1055107.