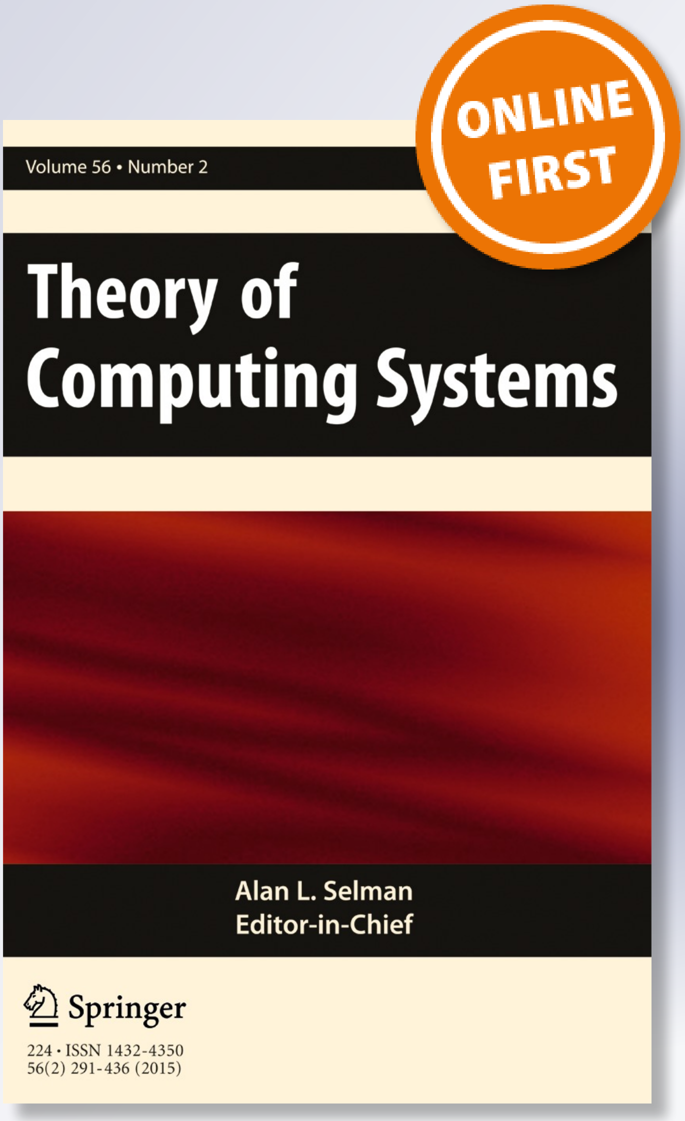
Algorithmic Minimal Sufficient Statistics: a New Approach

Nikolay Vereshchagin

Theory of Computing Systems

ISSN 1432-4350

Theory Comput Syst
DOI 10.1007/s00224-014-9605-1



Theory Comput Syst
DOI 10.1007/s00224-014-9605-1

Algorithmic Minimal Sufficient Statistics: a New Approach

Nikolay Vereshchagin

Abstract We introduce the notion of a strong sufficient statistic for a given data string. We show that strong sufficient statistics have better properties than just sufficient statistics. We prove that there are "strange" data strings, whose minimal strong sufficient statistic have much larger complexity than the minimal sufficient statistic.

Keywords Algorithmic statistics · Minimum description length · Randomness deficiency · Sufficient statistic · Denoising

1 Introduction

The goal of statistics is to provide adequate statistical hypotheses (models) for observed data. But what is an "adequate" model? To answer this question, one can use the notions of algorithmic information theory.

Assume that the given data x is a binary string. As models, consider probability distributions P on finite sets of strings with rational values. The quality of such a model, as an explanation for x, is measured by three parameters:

- Kolmogorov complexity $C(P)$ of P,[1]
- the minus log-likelihood $-\log_2 P(x)$ of P, and

[1] Kolmogorov complexity of P is defined as follows. We fix any computable bijection $P \mapsto [P]$ from the family of probability distributions to the set of binary strings, called *encoding*. Then we define $C(P)$ as the complexity $C([P])$ of the code $[P]$ of P.

Some results from this paper appeared in a preliminary form in Proceedings of CiE [8] and [9].

N. Vereshchagin (✉)
Moscow State University, National Research University Higher School of Economics (HSE),
Moscow, Russian Federation
e-mail: ver@mccme.ru

- the randomness deficiency of x with respect to P, defined as $-\log_2 P(x) - C(x|P)$.[2]

The less the parameters are the better the model is. Without loss of generality (see, e.g. [7, Appendix II, pp. 3282]) we can restrict the class of models for x to uniform probability distributions over finite sets $A \ni x$,[3] in which case the parameters become

- Kolmogorov complexity $C(A)$ of A,[4]
- $\log |A|$, the log-cardinality of A, and
- $\log |A| - C(x|A)$, the randomness deficiency of x in A.

In this paper we focus on the first two parameters. For a given string x an i, j*-description* for x is any set $A \ni x$ of complexity at most i and log-cardinality at most j.

Sufficient Statistics A finite set $A \subset \{0, 1\}^*$ is called a *sufficient statistic for* x if $x \in A$ and the sum of the Kolmogorov complexity of A and log-cardinality of A is close to the Kolmogorov complexity of x:

$$C(A) + \log |A| \approx C(x).$$

More specifically, we call A an ε*-sufficient* statistic for x if the left hand side exceeds the right hand side by at most ε. We do not require the inverse inequality, as it holds with precision $O(\log C(x))$ anyway.

For every x the singleton $\{x\}$ is an $O(1)$-sufficient statistic for x. The complexity of this statistic is about $C(x)$. If x is a random string of length n (that is, $C(x) \approx n$) then there is a $O(\log n)$-sufficient statistic for x of much lower complexity: the set of all strings of length n, whose complexity is about $\log n$, is a $O(\log n)$-sufficient statistic for x. We will think further of ε as having the order $O(\log n)$ and call such values *negligible*.

Sufficient Statistics and Useful Information Sufficient statistics for x are thought as models squeezing a noise from x. The explanation is the following. Let A be a sufficient statistic for x. One can show that in this case both the randomness deficiency of x in A and $C(A|x)$ are negligible. Let z be the binary notation of the ordinal number of x in A (with respect to the lexicographical order on A). As $C(A|x)$ is negligible, both conditional complexities $C(x|A, z)$ and $C(A, z|x)$ are also negligible.[5] Speaking informally, the two part code (A, z) of x has the same information as x itself, and its second part z is a string of length $\log |A|$ that is random conditional to its first part

[2] $C(x|P)$ and $C(P|x)$ are defined as $C(x|[P])$ and $C([P]|x)$, respectively, where $P \mapsto [P]$ is a fixed computable encoding of distributions by strings (see the previous footnote).

[3] Indeed, let P be a probability distribution on a finite set of strings with rational values. Let 2^{-m} stand for $P(x)$ rounded down to an integer power of 2. If $m > n = |x|$, then let U be the uniform distribution on all strings of length n. Otherwise let U be the uniform distribution on the set $A = \{x \mid P(x) \geqslant 2^{-m}\}$. It is easy to verify that all the three parameters of U are at most $O(\log n)$ larger than those of P.

[4] Kolmogorov complexity of finite subsets of $\{0, 1\}^*$ is defined via an encoding, similarly to complexity of distributions.

[5] $C(x|A, z)$ is defined as $C(x|[[A], z])$, where $(x, y) \mapsto [x, y]$ is a computable bijection between pairs of strings and strings; the notation $C(A, z|x)$ is understood in a similar way.

A.[6] This encourages us to qualify z as an accidental information (noise) in the pair (A, z), and hence in x. In other words, all useful information from x is captured by the set A.

Minimal Sufficient Statistics The most valuable sufficient statistic is the one that squeezes as much noise as possible from the data, that is, that has smallest complexity and largest cardinality. Such statistics are informally called *minimal sufficient statistics, MSS,* for x.

Thus any MSS for x extracts all useful information from x. What is that information? The answer is a bit confusing. Let Ω_k denote the number of strings of complexity at most k.[7] Let A be an MSS for a string x and let $m = C(A)$. As shown in [7, Theorem VIII.4(iii), p. 3281], both conditional complexities $C(A|\Omega_m)$ and $C(\Omega_m|A)$ are negligible. That is, all MSS of the same complexity (even if they are MSS for different strings) are equivalent. Thus any MSS for x does not provide any specific information about x: it provides the information about the number of strings of bounded complexity.

Total Conditional Complexity A possible way to resolve this paradox is the following. It seems that our definition of "having the same information" is too broad, we implicitly assumed that u and v have the same information, if both $C(u|v)$ and $C(v|u)$ are negligible. Under this assumption every string x has the same information as its shortest program x^*. In the context of separating the information into a useful one and an accidental one, such an assumption is certainly misleading. Indeed, the entire information in x^* (which is a random string) is noise, while x may have useful information. In algorithmic statistics, it is more helpful to think that u and v have the same information only if *total* conditional complexities $CT(u|v)$ and $CT(v|u)$ are negligible. The total conditional complexity $CT(u|v)$ is defined as the minimal length of a total program p for u conditional to v: $CT(u|v) = \min\{|p| : p(v) = u \text{ and } p(v') \text{ halts for all } v'\}$. The total conditional complexity can be much greater than the plain one. This fact was first observed in [6]. In this paper we show that there is a string x such that $CT(x|x^*)$ is large for all short programs x^* for x; on the other hand, for all strings x there is a short program x^* for x such that $CT(x^*|x)$ is negligible (Theorem 17 in the Appendix).

If both $CT(u|v)$ and $CT(v|u)$ are negligible, then u, v have similar statistical properties. We will call such strings *equivalent* in the sequel. In contrast to this, if both plain conditional complexities $C(u|v)$ are $C(v|u)$ are negligible then we will call u, v *weakly equivalent*

Strong Sufficient Statistics: the Definition Adopting this viewpoint we cannot consider any two MSS of the same complexity as equivalent objects any more. It might happen that a certain string has several non-equivalent MSS (in this paper we provide

[6] Indeed, $C(z|A)$ is up to an additive constant equal to $C(x|A)$, which is close to $\log |A|$.

[7] The information in Ω_k is a part of the information in Ω_l for $l \geqslant k$ (i.e., $C(\Omega_k|\Omega_l) = O(\log l)$). In fact, Ω_k contains the same information (up to $O(\log k)$ conditional complexity in both directions) as first k bits of Chaitin's Ω-number (a lower semicomputable random real), so we use the same letter Ω to denote it.

such an example). For such strings we would like to have a tool to distinguish between "true" and "false" models.

In this paper we propose the following way to do that. We call a set $A \ni x$ a *strong* statistic (or model) for x if $CT(A|x)$ is negligible. We call a set $A \ni x$ a *good* model (or statistic) for x if it is both strong and sufficient. As we mentioned, the sufficiency requirement implies only that plain (not total) conditional complexity $C(A|x)$ is negligible, thus a sufficient statistic might be not strong (we provide such an example later). Specifically, we call A an *ε-strong* model for x if $CT(A|x) \leqslant \varepsilon$ and we call A an *ε-good* model for x if A is both ε-strong and ε-sufficient for x. We call (quite informally) a set A a strong MSS for x if A is an MSS for x and A is a strong model for x. It might happen that no MSS for x is strong, in which case x has no strong MSS and all good model for x have larger complexity than that of MSS.

A set A is a strong model for x iff both total complexities $CT(x|A, z)$, $CT(A, z|x)$ are negligible, where z is the ordinal number of x in A. Indeed, given the pair (A, z) we can find x by means of a short total program (even if A is not strong). Conversely, if A is a strong statistic for x, then from x we can compute A by means of a short total program and then compute the ordinal number of x in A.

Example 1 Here is an example of a good model. Let y be any string and let $x = yz$, where z is a string of length m that is random conditional to y (that is, $C(z|y) \approx m$). Intuitively, x is obtained from y by adding m bits of noise and y captures all useful information from x. Consider the set $A = \{yz' : |z'| = m\}$ as a model for x. This model is good and is equivalent to y, which supports the viewpoint that good models are indeed "true" models of the data.

Again the most valuable strong sufficient statistics are those having minimal complexity, *minimal good statistics, MGS*. We will see that the complexity of MGS may be much larger than that of MSS.

Remark 1 Without loss of generality we can require in the definition of ε-strong model for x that there is a total program of length at most ε that maps *every* $x' \in A$ to $[A]$. Indeed, assume that A is an ε-strong model for x and p a short total program with $p(x) = [A]$. Then the parameters of the model $A' = \{x' \in A \mid p(x') = [A]\}$ are not much worse than those of A: its complexity is only about $|p|$ bits more and its log-cardinality is the same or less. And the following total program of length about $|p|$ transforms any $x' \in A'$ to $[A']$: given x' apply p to x' to find A and return (the code of) the set consisting of all $x'' \in A$ with $p(x'') = [A]$.

Strong Statistics: Existence Our first main result, Theorem 1, provides an example of a string x when the set of all models and strong models differ in a maximal possible way. To state it let *the optimality deficiency of a model A for x* be defined as $C(A) + \log |A| - C(x)$. Consider the upward closed set P_x, called the *profile of x*, consisting of all pairs $(i, j) \in \mathbb{N}^2$ such that x has a model of complexity at most i and optimality deficiency at most j:

$$P_x = \{(i, j) \mid (\exists A \ni x)\ C(A) \leqslant i,\ C(A) + \log |A| \leqslant C(x) + j\}.$$

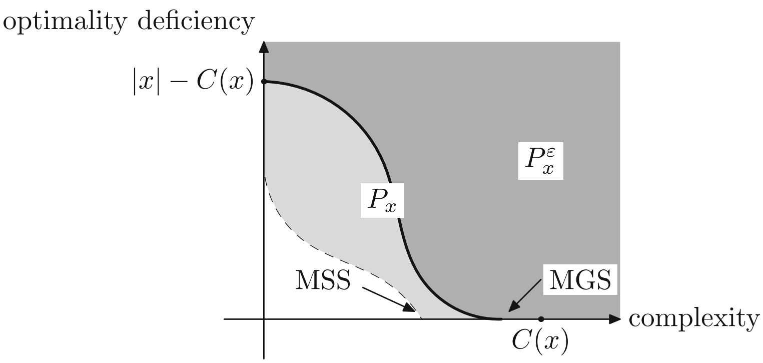


Fig. 1 The set P_x is to the right of the *dashed line*. The set P_x^ε is to the right of the *bold line*

Similarly, let P_x^ε denote the set of all pairs $(i, j) \in \mathbb{N}^2$ such that x has an ε-strong model of complexity at most i and optimality deficiency at most j. Obviously the set P_x includes the set P_x^ε (see Fig. 1). A set A is an ε-sufficient statistic for x iff the optimality deficiency of A is at most ε.

The larger the profile of x is the more stochastic the string x is. Kolmogorov called *stochastic* those strings whose profile is close to the set of all pairs of naturals.[8]

We provide an example of a string x such that P_x^ε is much smaller than P_x for a large enough ε (see Fig. 2).

Theorem 1 (on existence of strange strings, informal) *If $\varepsilon \leqslant k \leqslant n$ then there is a string x of length n and complexity k whose sets P_x and P_x^ε are close to those shown on* Fig. 2.

For instance, let $\varepsilon = n/3$, $k = 2n/3$ and x a string existing by Theorem 1. Basically, the only good model for x is the singleton set $\{x\}$. The complexity of any MSS for x is $n/3$, which is much smaller than the complexity of any MGS ($2n/3$). Hence x has no strong MSS. Informally, we will call such strings *strange*.

On the other extreme there are strings x for which the sets P_x and P_x^ε are close to each other (say, each of them is in $O(\log C(x))$-neighborhood of the other) for some negligible ε. In contrast to strange strings, we will call such strings x *normal*. Every normal string has a strong MSS (the complexities of MGS and MSS are close). One can wonder if normal strings can have bad MSS at all. Our second result is an example of such a string.

Theorem 2 (on existence of a normal string having a bad MSS, informal) *For every k there is a string x of length $3k$ and complexity $2k$ such that both sets P_x and P_x^ε are close to the set shown on* Fig. 3 *(for $\varepsilon = O(\log k)$). Moreover, the string x has a minimal sufficient statistic B such that $CT(B|x)$ is close to k.*

[8] More specifically, he called a string x α, β-stochastic if there is a model for x of complexity at most α and log-cardinality at most $C(x) + \beta$. With logarithmic accuracy this means that the pair $(\alpha, \alpha + \beta)$ is in P_x.

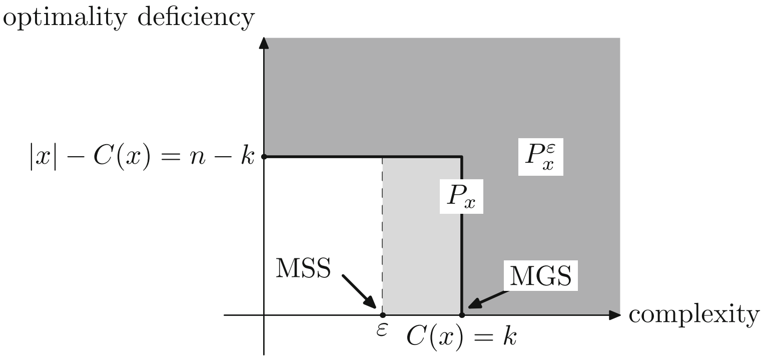


Fig. 2 The sets P_x and P_x^ε for the strange string from Theorem 1. The set P_x is to the right of the *dashed line*. The set P_x^ε is to the right of the *bold line*

Strong Sufficient Statistics: Nice Properties We are able to show that good strong sufficient statistics have two nice properties. The first property is the following: any strong MSS for x can be obtained by a short total program from any other sufficient statistic for x.

Theorem 3 (on total complexity of any strong MSS given any sufficient statistic, informal) *For any strong MSS A for x and for any sufficient statistic B for x the total complexity $CT(A|B)$ is negligible.*

For the class of all models (and not only strong ones) this property holds for plain conditional complexity C in place of total complexity CT, see Corollary 12 below.

Theorem 3 implies that for any normal string x there is a unique MGS for x. Indeed, if A, B are minimal good statistics for a normal string x then A, B are also minimal sufficient statistics for x. Being both strong, A, B satisfy the assumptions of Theorem 3 and thus $CT(A|B)$ is negligible. The same applies for $CT(B|A)$. It remains an open question whether MGS is unique for strange strings.

The second property of good models is the following: for any good model A for x the set $P_{[A]}$ is close to the set P_x (we say that two sets of pairs are close, if each of them is included in a small neighborhood of the other).

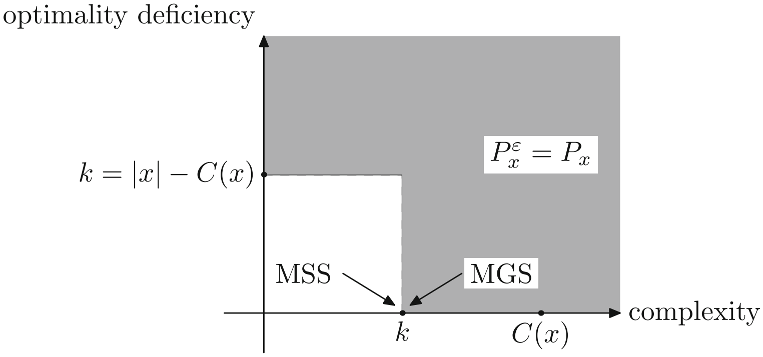


Fig. 3 The sets $P_x = P_x^\varepsilon$ for the normal string from Theorem 2

Theorem 4 (on the profile of good models, informal) *If A is a good model for x then the sets P_x and $P_{[A]}$ are close.*

The profile of a string expresses how stochastic the string is. Adding or removing noise should not affect stochasticity in any way and thus it should not change the profile. This theorem says that noise identification via strong sufficient statistics is consistent with this intuition. Notice that this theorem does not hold for all sufficient statistics [8, Theorem 3]. That is why we think that strong sufficient statistics capture better the intuition of noise removal than just sufficient statistics.

Step-wise Denoising Assume that we have removed some noise from a normal data string x by finding a good model A for x. Theorem 3 states that to remove the remaining noise from x we do not need x anymore: any strong MSS B for x has negligible total complexity conditional to A. Moreover, if A is itself normal then a strong MSS B for x can be obtained by denoising A, in a full accordance with our intuition of step-wise noise removal. Indeed, let $\mathcal{A}$ stand for a strong MSS for A. Consider then the set

$$B = \cup_{A' \in \mathcal{A} : \lceil \log |A'| \rceil = \lceil \log |A| \rceil} A'.$$

This set is a $C(\mathcal{A})$, $\log |\mathcal{A}| + \log |A|$-description for x, which is an MSS for x by Theorem 4. Finally, $CT(B|x) \leqslant CT(B|\mathcal{A}) + CT(\mathcal{A}|A) + CT(A|x) \approx 0 + 0 + 0 = 0$, thus B is a strong model for x.

It remains an open question whether any strong MSS for a normal string is normal.

2 Preliminaries

We consider strings over the binary alphabet $\{0, 1\}$, and by $|x|$ we denote the length of a string x. We will use the notation $\log i$ for $\lceil \log_2 i \rceil$. We say that two subsets P, Q of $\mathbb{N}^2$ are ε-close if each of them is in ε-neighborhood of the other (with respect to L_2-norm).

2.1 Kolmogorov Complexity

The Kolmogorov complexity $C(x)$ of a binary string x and conditional Kolmogorov complexity $C(x|y)$ of a binary string x given another string y are defined as follows. We call a Turing machine U with input alphabet 0,1 *universal* if for every other Turing machine V with the same input alphabet there is a binary string c_V such that $U(c_V p, y) = V(p, y)$ for all binary strings p, y. The notation $V(p, y)$ refers to the output of machine V when run on input binary strings p, y separated by the blank symbol. Fix a universal machine U and denote $U(p, y)$ by $p(y)$. The conditional Kolmogorov complexity is defined as

$$C(x|y) = \min\{|p| : p(y) = x\}.$$

If $p(y) = x$, we say that p is a program for x conditional to (or given) y. Unconditional Kolmogorov complexity $C(x)$ is defined as $C(x|\text{empty string})$.

The *total* Kolmogorov complexity is defined as

$$CT(x|y) = \min\{|p| : p(y) = x, \text{ and } p(y') \text{ is defined for all } y'\}.$$

Kolmogorov complexity of a finite set of strings is defined as follows. We fix any computable bijection $A \mapsto [A]$ between finite sets of binary strings and binary strings and let $C(A) = C([A])$. The expressions $C(x|A)$, $C(A|x)$ are understood as $C(x|[A])$, $C([A]|x)$, respectively. In a similar way we define Kolmogorov complexity of pairs of strings: by definition $C(x, y) = C([x, y])$, where $(x, y) \mapsto [x, y]$ is a computable bijection between pairs of strings and strings.

We will use in the sequel without reference the following properties of Kolmogorov complexity:

- The number of strings of Kolmogorov complexity less than k is less than 2^k.
- $C(x) \leqslant |x| + c$, $C(x|y) \leqslant C(x) + c$, for some c and all x, y;
- For every computable function f mapping strings to strings there exists c such that $C(f(x)|x) \leqslant c$ and $C(f(x)) \leqslant C(x) + c$ for all x;
- (Conditional version of the previous inequality.) For every computable function f mapping pairs of strings to strings there exists c such that $C(f(x, y)|y) \leqslant C(x|y) + c$ for all x, y;
- Symmetry of information: $C(x, y) \approx C(x) + C(y|x)$. This equality holds with "logarithmic precision". Specifically, we have

  $$C(x, y) \leqslant C(x) + C(y|x) + 2\log\min\{C(x), C(y|x)\} + c$$

  for some c and all x, y, and

  $$C(x) + C(y|x) \leqslant C(x, y) + 3\log C(x, y) + c.$$

- (Conditional version of the symmetry of information). For all x, y, z,

  $$C(x, y|z) \approx C(x|z) + C(y|x, z).$$

  Here one inequality is true up to a $2\log\min\{C(x|z), C(y|x, z)\} + c$ error term and the other one up to a $3\log C(x, y|z) + c$ error term.

2.2 Optimality Deficiency and Randomness Deficiency

The difference $C(A) + \log|A| - C(x)$ is called *the optimality deficiency of A as a model for x*. The randomness deficiency is always less than the optimality deficiency and the difference between them equals $C(A|x)$ (with additive error $O(\log C(x) + \log C(A))$). This follows from the symmetry of information:

$$C(x) + C(A|x) = C(x, A) = C(A) + C(x|A).$$

Adding $\log|A| - C(x) - C(x|A)$ to both sides we obtain the equality

$$C(A|x) + (\log|A| - C(x|A)) = C(A) + \log|A| - C(x).$$

In particular if A is ε-sufficient statistic for x then both $C(A|x)$ and randomness deficiency of x in A do not exceed ε (with additive error $O(\log C(x) + \log C(A))$).

3 Strange Strings

Our first main result establishes the existence of "strange" strings x.

Theorem 5 (on existence of strange strings, formal) *Assume that natural numbers k, n, ε satisfy the inequalities*

$$O(1) \leqslant \varepsilon \leqslant k \leqslant n.$$

Then there is a string x of length n and complexity $k + O(\log n)$ such that the sets P_x and P_x^ε are $O(\log n)$-close to the sets shown on Fig. 2.

Theorem 5 does not say anything about how rare are strange strings. Such strings are rare, as for majority of strings x of length n the set $\{0, 1\}^n$ is a strong MSS for x. A more meaningful question is whether such strings might appear with high probability in a statistical experiment. More specifically, assume that we sample a string x in a given set $A \subset \{0, 1\}^n$, where all elements are equiprobable. Might it happen that with high probability (say with probability 99 %) x is strange? An affirmative answer to this question is given in the following

Theorem 6 *Assume that natural $k, n, \varepsilon, \delta$ satisfy the inequalities*

$$O(1) \leqslant \varepsilon \leqslant k \leqslant n, \quad \delta \leqslant k - \varepsilon.$$

Then there is set $A \subset \{0, 1\}^n$ of cardinality $2^{k-\varepsilon}$ and complexity at most $\varepsilon + O(\delta + \log n)$ such that all but $2^{k-\varepsilon-\delta}$ its elements x have complexity $k + O(\delta + \log n)$ and the sets P_x and P_x^ε are $O(\delta + \log n)$-close to the sets shown on Fig. 2.

Proofs of Theorems 5 and 6. We start with the following observation from [2, 3, 5].

Lemma 7 *If a string x has a model A of complexity at most i with $C(A) + \log |A| \leqslant m$ then x has a $(i + O(\log n)), (m - i)$-description B. The same applies to strong models: if A is ε-strong then B is $\varepsilon + O(\log n)$-strong.*

Proof If $\log |A| \leqslant m - i$ there is nothing to prove. Otherwise, let $l = \log |A| - (m - i)$ and chop A into 2^l parts of size 2^{m-i}. The part B containing x has complexity at most $C(A) + l + O(\log l)$ and $C(A) + l = C(A) + \log |A| - (m - i) \leqslant i$.

To prove the second statement add to the given short total program to transform x to A the information l in $\log l \leqslant \log m \leqslant \log n$ bits. The last inequality holds, as without loss of generality we may assume that $m \leqslant n$. □

We will use also another simple observation:

Lemma 8 *Assume that A is an ε-strong statistic for a string x of length n. Let $y = [A]$ be the code of A. Then y has an $(\varepsilon + O(\log n)), n$-description.*

Proof Let p be a string of length at most ε such that $p(x) = y$ and $p(x')$ is defined for all strings x'. Consider the set $\{p(x') \mid x' \in \{0,1\}^n\}$. Its cardinality is at most 2^n and complexity at most $\varepsilon + O(\log n)$. □

To prove Theorem 5 it suffices to find a set $A \subset \{0,1\}^n$ with

(a) $C(A) \leqslant \varepsilon + O(\log n)$, $\log |A| \leqslant k - \varepsilon$
which is not covered by sets from the following three families:
(b) the family $\mathcal{B}$ consisting of all sets $B \subset \{0,1\}^*$ with with $C(B) \leqslant \varepsilon$, $\log |B| \leqslant n - \varepsilon - 4$,
(c) the family $\mathcal{C}$ consisting of all sets M with $C(M) \leqslant k$, $\log |M| \leqslant n - k - 4$ whose code $[M]$ has a $(\varepsilon + O(\log n))$, n-description, and
(d) the family $\mathcal{D}$ consisting of all singletons sets $\{x\}$ where $C(x) < k$.

Indeed, assume that we have such set A. As x we can take any non-covered string in A. Notice that item (a) implies that the complexity of x is at most $k + O(\log n)$, and item (d) implies that it is at least k. Thus the membership of the pair $(k + O(\log n), O(\log n))$ in P_x^ε is witnessed by the singleton $\{x\}$, provided ε is greater than the constant from the inequality $CT(\{x\}|x) = O(1)$. The membership of the pair $(O(\log n), n - k + O(\log n))$ in P_x^ε and P_x will be witnessed by the set of all strings of length n, provided ε is greater than then constant from the inequality $CT(\{0,1\}^n|x) = O(1)$. The membership of the pair $(\varepsilon + O(\log n), O(\log n))$ in P_x will be witnessed by the set A.

The upper bounds for P_x^ε and P_x follow from (b), (c) and Lemmas 7 and 8. Indeed, item (b) implies that the pair $(\varepsilon - O(\log n), n - k - O(\log n))$ is not in P_x. Item (c) implies that the pair $(k - O(\log n), n - k - O(\log n))$ is not in P_x^ε.

Let us show that there is a set A of n-bit strings satisfying (a), (b), (c) and (d). A direct counting reveals that the family $\mathcal{B} \cup \mathcal{C} \cup \mathcal{D}$ covers at most

$$2^{\varepsilon+1}2^{n-\varepsilon-4} + 2^{k+1}2^{n-k-4} + 2^k \leqslant 2^{n-3} + 2^{n-3} + 2^{n-4} < 2^{n-1}$$

strings and hence at least half of all n-bit strings are non-covered. However we cannot let A be any $2^{k-\varepsilon}$-element non-covered set of n-bit strings, as in that case $C(A)$ could be large.

We first show how to find A, as in (a), that is not covered by $\mathcal{B} \cup \mathcal{D}$ (but may be covered by $\mathcal{C}$). This is done using the method of [7]. To construct A notice that both the families $\mathcal{B}$ and $\mathcal{D}$ can be enumerated given k, ε, n by running the universal machine U in parallel on all inputs. We start such an enumeration and construct A "in several attempts". During the construction we maintain the list of all strings covered by sets from $\mathcal{B} \cup \mathcal{D}$ enumerated so far. Such strings are called *marked*. Initially, no strings are marked and A contains the lexicographic first $2^{k-\varepsilon}$ strings of length n. Each time a new set $B \in \mathcal{B}$ appears, all its elements receive a b-mark and we replace A by any set consisting of $2^{k-\varepsilon}$ yet non-marked n-bit strings. Each time a new set $\{x\}$ in D appears, the string x receives a d-mark, but we do not immediately replace A. However we do that when all strings in A receive a d-mark, replacing it by any set consisting of $2^{k-\varepsilon}$ yet non-marked n-bit strings. The above counting shows that such replacements are always possible.

The last version of A (i.e. the version obtained after the last set in $\mathcal{B} \cup \mathcal{D}$ have appeared) is the sought set. Indeed, by construction $|A| = 2^{k-\varepsilon}$ and A is not covered by sets in $\mathcal{B} \cup \mathcal{D}$. It remains to verify that $C(A) \leqslant \varepsilon + O(\log n)$. This follows from the fact that A is replaced at most $O(2^{\varepsilon})$ times, and hence can be identified by the number of its replacements and ε, k, n (we run the above construction of A and wait until the given number of replacements are made).

Why is A replaced at most $O(2^{\varepsilon})$ times? The number of replacements caused by appearance of a new set $B \in \mathcal{B}$ is at most $2^{\varepsilon+1}$. The number of strings with a d-mark is at most 2^k and hence A can be replaced at most $2^k/2^{k-\varepsilon} = 2^{\varepsilon}$ times due to receiving d-marks.

Now we have to take into account strings covered by sets from the family $\mathcal{C}$. To this end modify the construction as follows: put a c-mark on all strings from each set C enumerated into $\mathcal{C}$ and replace A each time when all its elements have received c or d marks (or when a new set is enumerated into $\mathcal{B}$).

However this modification alone is not enough. Indeed, up to $\Omega(2^n)$ strings may receive a c-mark, and hence A might be replaced up to $\Omega(2^{n-(k-\varepsilon)})$ times due to c-marks. The crucial modification is the following: each time A is replaced, its new version is not just any set of $2^{k-\varepsilon}$ non-marked n-bit strings but a carefully chosen such set.

To explain how to choose A we first represent $\mathcal{C}$ as an intersection of two families, $\mathcal{C}'$ and $\mathcal{C}''$. The first family $\mathcal{C}'$ consists of all sets M with $C(M) \leqslant k$ and the second family $\mathcal{C}''$ of all sets C with $\log |C| \leqslant n - k - 4$ whose code $[C]$ has a $(\varepsilon + O(\log n), n)$-description. The first family is small (less than 2^{k+1} sets) and the second family has only small sets (at most 2^{n-k-4}-element sets) and is not very large ($|\mathcal{C}''| = 2^{O(n)}$). Both families can be enumerated given ε, k, n and, moreover, the sets from $\mathcal{C}''$ appear in the enumeration in at most $2^{\varepsilon+O(\log n)}$ portions. Due to this property of $\mathcal{C}''$ we can update A each time a new portion of sets in $\mathcal{C}''$ appears—this will increase the number of replacements of A by $2^{\varepsilon+O(\log n)}$, which is OK.

The crucial change in construction is the following: each time A is replaced, its new version is *a set of* $2^{k-\varepsilon}$ *non-marked n-bit strings that has at most $O(n)$ common strings with every set from the part of $\mathcal{C}''$ enumerated so far*. (We will show later that such a set always exists).

Why does this solve the problem? There are two types of replacements of A: those caused by enumerating a new set in $\mathcal{B}$ or a new bunch of sets in $\mathcal{C}''$ and those caused by that all elements in A have received c- or d-marks. The number of replacement of the first type is at most $2^{\varepsilon+O(\log n)}$. Replacements of the second type are caused by enumerating new singleton sets in $\mathcal{D}$ and by enumerating new sets C in $\mathcal{C}'$ which were enumerated into $\mathcal{C}''$ on earlier steps. Due to the careful choice of A, when each such set C appears in the enumeration of $\mathcal{C}'$ it can mark only $O(n)$ strings in the current version of A. The total number of sets in $\mathcal{C}'$ is at most 2^k. Therefore the total number of events "a string in the current version of A receives a c-mark" is at most $O(n2^k)$. The total number of d-marks is at most 2^k. Hence the number of replacements of the second type is at most

$$(O(n2^k) + 2^k)/2^{k-\varepsilon} = O(n2^{\varepsilon}).$$

Thus it remains to show that we indeed can always choose A, as described above. This will follow from a lemma that says that in a large universe one can always choose a large set that has a small intersection with every set from a given small family of small sets.

Lemma 9 *Assume that a finite family $\mathcal{C}$ of subsets of a finite universe U is given and each set in $\mathcal{C}$ has at most s elements. If*

$$|\mathcal{C}|\binom{N}{t+1}\left(\frac{s}{|U|-t}\right)^{t+1}<1$$

then there is an N-element set $A\subset U$ that has at most t common elements with each set in $\mathcal{C}$.

Proof To prove the lemma we use probabilistic method. The first element a_1 of A is chosen at random among all elements in U with uniform distribution, the second element a_2 is chosen with uniform distribution among the remaining elements and so forth.

We have to show that the statement of the theorem holds with positive probability. To this end note that for every fixed C in $\mathcal{C}$ and for every fixed set of indexes $\{i_1,\dots,i_{t+1}\}\subset\{1,2,\dots,N\}$ the probability that *all* $a_{i_1},\dots,a_{i_{t+1}}$ fall in C is at most $\left(\frac{s}{|U|-t}\right)^{t+1}$. The number of sets of indexes as above is $\binom{N}{t+1}$. By union bound the probability that a random set A does not satisfy the lemma is upper bounded by the left hand side of the displayed inequality. □

We apply the lemma for U consisting of all non-marked n-bit strings, for $N=2^{k-\varepsilon}$ and for $\mathcal{C}$ consisting of all sets in $\mathcal{C}''$ appeared so far. Thus $s=2^{n-k-4}$, $|U|\geqslant 2^{n-1}$, $|\mathcal{C}|=2^{O(n)}$, and we need to show that for some $t=O(n)$ it holds

$$2^{O(n)}\binom{2^{k-\varepsilon}}{t+1}\left(\frac{2^{n-k-4}}{2^{n-1}-t}\right)^{t+1}<1,$$

which easily follows from the inequality $\binom{2^{k-\varepsilon}}{t+1}\leqslant 2^{(k-\varepsilon)(t+1)}$. Theorem 5 is proved.

Theorem 6 is proved similarly to Theorem 5. The only difference is that we change A each time when at least $2^{k-\varepsilon-\delta}$ strings in A receive c- or d-marks. As the result, the number of changes of A will increase 2^{δ} times and the complexity of A will increase by δ.

4 Good Models and their Properties

The next theorem shows that $CT(A|B)$ is close to $C(A|B)$ for all good models A, B. In what follows we will consider models for x of complexity at most $O(C(x))$ (there is no need for more complex models).

Theorem 10 *If A is an ε-strong statistic for x and B is an ε-sufficient statistic for x then $CT(A|B)\leqslant C(A|B)+O(\varepsilon+\log k)$ where $k=C(x)$.*

Proof The idea is the following. Let p be a program witnessing $CT(A|x) \leqslant \varepsilon$. We first show that A has the following feature: there are many strings $x' \in B$ with $p(x') = [A]$. More specifically, at least $2^{-C(A|B)}$ fraction of x' from B have this property. At most $2^{C(A|B)}$ sets A' can have this feature, as each such A' can be identified by the portion of $x' \in B$ with $p(x') = [A']$. Given B and p we are able to find a list of all such A' by means of a short total program. Given B, the set A can be identified by p and its index in that list.

First we show that there are many $x' \in B$ with $p(x') = [A]$ (otherwise B could not be a sufficient statistic for x). Let $D = \{x' \in B \mid p(x') = [A]\}$. Obviously, D includes x and

$$C(D|B) \leqslant C(A|B)$$

(ignoring terms of order $O(\varepsilon + \log k)$). Given B and p the string x can be identified by its index in D, therefore

$$C(x|B) \leqslant C(D|B) + \log|D| \leqslant C(A|B) + \log|D|.$$

On the other hand, $C(x|B) = \log|B|$, as B is ε-sufficient. Hence $\log|D| \geqslant \log|B| - C(A|B)$. Recall that we ignored terms of order $O(\varepsilon+\log k)$. Actually, we have shown that for some constant c we have $\log|D| \geqslant \log|B| - C(A|B) - c(\varepsilon + \log k)$.

Consider now all A' such that

$$\log|\{x' \in B \mid p(x') = [A']\}| \geqslant \log|B| - C(A|B) - c(\varepsilon + \log k).$$

Each such A' can be identified by the portion of $x' \in B$ with $p(x') = [A']$. Thus there are at most $2^{C(A|B)+c(\varepsilon+\log k)}$ different such A's. Given B and $C(A|B)$, p, ε we are able to find the list of all A's. The program that maps B to the list of A's is obviously total. Therefore there is a $C(A|B) + O(\varepsilon + \log k)$-bit total program that maps B to A and $CT(A|B) = C(A|B) + O(\varepsilon + \log k)$. □

We will now derive from this theorem that $CT(A|B)$ is negligible for every strong MSS A and every sufficient statistic B for the same string x. To state this result rigorously we have to define the notion of MSS formally. When trying to do that, we face the following problem: for certain strings x a negligible increase in ε may cause a large decrease of the minimal complexity of ε-sufficient statistics for x.

We overcome this problem as follows. We call a set A a *δ-minimal ε-sufficient statistic* for x if A is an ε-sufficient statistic for x and there is no $\varepsilon + f(C(x))$-sufficient statistic B for x with

$$C(B) < C(A) - \delta.$$

Here $f(C(x))$ denotes a fixed large enough function of $C(x)$ of order $O(\log C(x))$ that will chosen later so that all our results be valid. See Fig. 4.

As before, we think of ε as a large enough value of order $O(\log C(x))$ so that for all x the singleton set $\{x\}$ be an ε-sufficient statistic for x and thus ε-sufficient statistics exist. However existence of ε-sufficient statistics does not imply that a δ-minimal ε-sufficient statistic exists. Indeed, ε-sufficient statistic of the smallest complexity is not necessarily a δ-minimal ε-sufficient statistic: this happens if we can further

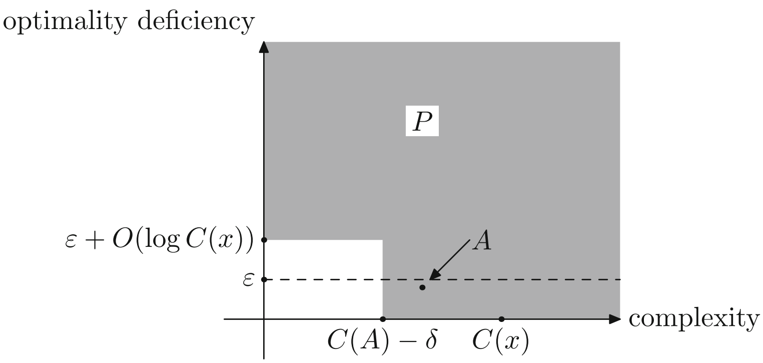


Fig. 4 The set A is δ-minimal ε-sufficient statistic for x. This means that its parameters lie below the *dashed line* and the profile of x is included in the gray set P

decrease the complexity of models for x by δ at the expense of increasing the optimality deficiency by only $f(C(x))$. However, if there is a δ-minimal ε-sufficient statistic for x, then so is any ε-sufficient statistic A for x of the smallest complexity. In that case the complexity of any other δ-minimal ε-sufficient statistic for x is at most δ more than that of A.

Now we are going to show that if A is an MSS for x then we can drop the term $C(A|B)$ in the statement of Theorem 10 and show that $CT(A|B) = O(\varepsilon + \delta + \log k)$. To this end we need a new definition.

Let Ω_m denote the number of strings of complexity at most m. Consider its binary representation, i.e., the sum

$$\Omega_m = 2^{s_1} + 2^{s_2} + \ldots + 2^{s_t}, \text{ where } s_1 > s_2 > \ldots > s_t.$$

According to this decomposition, we may split the list L_m of strings of complexity at most m into groups: first 2^{s_1} elements, next 2^{s_2} elements, etc. (We assume that an algorithm is fixed that, given m, enumerates all strings of complexity at most m in some order. That order is used to partition the list into groups.) Let us denote by $S_{m,s}$ the group of size 2^s from this partition. Notice that $S_{m,s}$ is defined only for s that correspond to ones in the binary representation of Ω_m.

If x is a string of complexity at most m, it belongs to some group $S_{m,s}$, and this group can be considered as a model for x. We may consider different values of m (starting from $C(x)$). In this way we get different models $S_{m,s}$ for the same x. The complexity of $S_{m,s}$ is $m - s + O(\log m)$. Indeed, chop L_m into portions of size 2^s each, then $S_{m,s}$ is the last full portion and can be identified by m, s and the number of full portions, which is less than $\Omega_m/2^s < 2^{m-s+1}$. The size of $S_{m,s}$ is 2^s by definition and thus the sum of complexity and log-cardinality is $m + O(\log m)$. The models $S_{m,s}$ were introduced in [2].

Theorem 11 (Theorem VIII.4(ii) from [7]) *Assume that $x \in A$ and $m \geqslant C(x')$ for all $x' \in A$. Let s be the (unique) integer such that the set $S_{m,s}$ defined above contains x. Then $S_{m,s}$ is simple conditional to A, that is, $C(S_{m,s} \mid A) = O(\log m)$.*

Actually, the authors of [2, 7] used prefix complexity K in place of the plain complexity C and Theorem 11 was stated only for $m = K(A) + \log|A| + O(1)$. It is easy to verify that Theorem 11 holds, as we stated it, with the same proof. In Appendix we present the complete proof.

Corollary 12 *Assume that A is a δ-minimal ε-sufficient statistic for x and B is an ε-sufficient statistic for x. Then $C(A|B) = O(\delta + \log C(x))$.*

Proof Let m be equal to the maximal complexity of elements of $A \cup B$. Let s be the (unique) integer such that the set $S_{m,s}$ defined above contains x. Then $m \leqslant C(x) + \varepsilon + O(\log C(x))$. Thus the sum of complexity and log-cardinality of $S_{m,s}$ is at most $C(x) + \varepsilon + O(\log C(x))$. As A is a δ-minimal ε-sufficient statistic for x, we can conclude that $C(S_{m,s}) \geqslant C(A) - \delta$ (provided the function f in the definition of δ-minimal ε-sufficient statistic is large enough).

By Theorem 11, $C(S_{m,s}|A)$ is negligible, therefore $C(A|S_{m,s})$ is at most $O(\delta + \log C(x))$ (symmetry of information).

Again by Theorem 11, $C(S_{m,s}|B)$ is negligible. As $C(A|B) \leqslant C(A|S_{m,s}) + C(S_{m,s}|B)$, we conclude that $C(A|B)$ is at most $O(\delta + \log C(x))$. □

This corollary and Theorem 10 imply the following

Theorem 13 (on total complexity of any good MSS given any sufficient statistic, formal) *If A is both a δ-minimal ε-sufficient statistic and an ε-strong model for x and B is an ε-sufficient statistic for x then $CT(A|B) = O(\varepsilon + \delta + \log C(x))$.*

The next theorem shows that the profile of any good model for x coincides with the profile of x itself. Recall that two sets are ε-close if each of them is inside ε-neighborhood of the other (with respect to L_2-norm).

Theorem 14 (on the profile of good models, formal) *If A is an ε-good model for x then the sets P_x and $P([A])$ are $O(\varepsilon + \log n)$-close. Here n is the length of x.*

This theorem is an easy corollary of the following lemma that relates the profiles of yz and y in the case when z is a random string conditional to y.

Lemma 18 *Let $x = yz$ where y is a string of complexity k and z is a string of length m with $C(z|y) \geqslant m - \delta$. Then the sets P_x and P_y are $O(\log n + \delta)$-close.*

The proof of this lemma is presented in Appendix.

Proof of Theorem 14 Assume that A is an ε-good model for x. Let z be the $\log|A|$-bit index of x in A. Then x and $[A]z$ are equivalent (the total conditional complexities are at most $\varepsilon + O(\log n)$).

If both total conditional complexities $CT(u|v)$ and $CT(v|u)$ are less than α then the sets P_u and P_v are $O(\alpha)$-close. Indeed, in this case $C(u)$ and $C(v)$ differ by at most $O(\alpha)$. Furthermore, let p witness the inequality $CT(v|u) \leqslant \alpha$. Then the mapping

$A \mapsto \{p(u') \mid u' \in A\}$ is a translation of models of u to models of v showing that P_u is included in the $O(\alpha)$ neighborhood of P_v. In a similar way we can show that P_v is included in the $O(\alpha)$ neighborhood of P_u.

Therefore P_x and $P_{[A]z}$ are $O(\varepsilon + \log n)$-close. On the other hand, $C(z|[A]) \geqslant \log|A| - \varepsilon - O(\log n)$ and by Lemma 15 $P_{[A]z}$ and $P_{[A]}$ are $O(\varepsilon + \log n)$-close. □

5 A Normal String Having a Weak MSS

The next result is a separation of good and weak models in the case when the sets P_x and P_x^ε coincide. Such x will have the form yz where z is a random string conditional to y, as in the Example 1 and Lemma 15.

Theorem 16 (on existence of a normal string having a bad MSS, formal) *For every k there is a string x of length $3k$ and Kolmogorov complexity $2k$ whose set P_x and $P_x^{O(\log k)}$ are $O(\log k)$ close to the set shown on* Fig. 5. *Moreover, the string x has a model B of complexity $C(B) = k$ and log-cardinality $\log|B| = k$ such that $CT(B|x) = k$. All equalities here hold up to $O(\log k)$ additive error term.*

Proof We will let $x = yz$ where $|y| = 2k$, $|z| = k$ and $C(z|y) \geqslant k$. To construct y call a string y of length $2k$ *simple* if it has a model S with $C(S) < k$ and $C(S) + \log|S| < 2k - \log 2k$. Otherwise call y *complex*. The total number of simple strings is strictly less than 2^{2k}. Let y be the lexicographical first complex string of length $2k$.

Now we let

$$B = \{yz' \mid z' \in \{0,1\}^k\} \cup \{x' \mid C(x') < k\}.$$

Let us show that $CT(B|x) \geqslant k$. Assume that p is a total program mapping x to B. Then let u be the first string outside all sets $p(x'')$ for $|x''| = 3k$. Then u is outside B and hence $C(u) \geqslant k$. Hence $|p| > k - O(\log k)$.

By construction we have $C(y) \geqslant k$. On the other hand, we claim that $C(y) \leqslant k + O(1)$. Indeed, y can be found given k and the set of all simple strings. The set

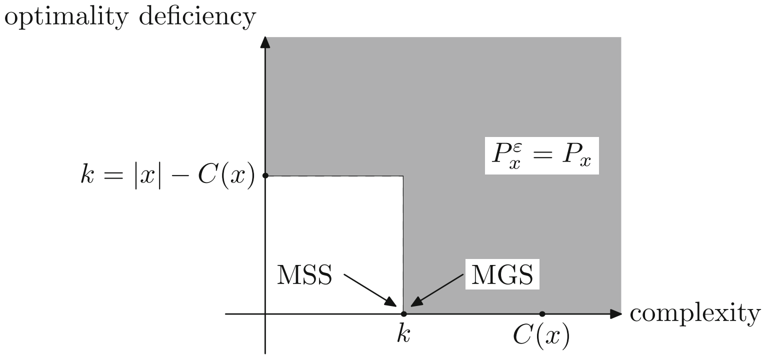


Fig. 5 The sets $P_x = P_x^\varepsilon$ for the string x from Theorem 16

of simple strings can be found given the set of all halting programs of length less than k, which in turn can be identified by the k-bit number N_k of halting programs of length less than k (we run all programs of length less than k until N_k of them halt). The same argument shows that $C(B) \leqslant k + O(1)$.

Obviously $|B| < 2^{k+1}$. It remains to show that x has the required sets P_x and $P_x^{O(\log k)}$. To this end we first show that the set P_y has the form shown on Fig. 5. The models $\{0,1\}^{2k}$ and $\{y\}$ witness that $P_y^{O(\log k)}$ includes the gray set of Fig. 5. On the other hand, by construction P_y is included in the gray set shown on Fig. 5.

The claim about the set P_x now follows from Lemma 15. It remains to show that the set $P_x^{O(\log k)}$ includes the gray set shown on Fig. 5. This is witnessed by the sets $\{0,1\}^{3k}$ and $\{x\}$. □

6 Open Questions

1) Is there a string that has non-equivalent MGS?
2) Is there a string that has a strong MSS but no model of the type $S_{m,s}$ is a strong MSS for x?
3) Is there such a normal string?
4) Is any strong MSS for a normal string normal?

Acknowledgments The work was in part supported by the RFBR grant 12-01-00864 and the ANR grant ProjetANR-08-EMER-008 Moscow State University, National Research University Higher School of Economics (HSE)

Appendix A: Theorem 17 and Deferred Proofs

A.1 Theorem 17

There are strings x for which it is much easier to transform x to a short program x^* for x than the other way around, and, moreover, it is hard to obtain x from *any* string that is significantly shorter than x itself.

Theorem 17 *(1) For all n and all strings x of length n there is a program x^* for x with $CT(x^*|x) \leqslant 2\log n + O(1)$ and $|x^*| \leqslant C(x) + O(1)$.*

(2) For all n there is a string x of length n and complexity at most $n/3 + O(1)$ such that for all strings p of length at most $2n/3$ we have $CT(x|p) \geqslant n/3 - 2$ (in particular this holds for all programs p for x of length at most $2n/3$).

Proof (1) Theorem 1.1 from [1] shows that there exists a total computable function f that for any x produces a list with $O(|x|^2)$ many elements containing a program p for x of length $|p| = C(x) + O(1)$. Then $CT(p|x) \leqslant 2\log n + O(1)$, as we can specify p by its index in the list $f(x)$.

(2) Initialize Q to be the empty set and x to be the lexicographically first string of length n. Run dovetail style all programs q of length at most $n/3 - 2$ for all

inputs p of length at most $2n/3$. Once such a program q halts for all p, we add q in the set Q and find a new candidate for x, which is the first string outside the set $\{q(p) : q \in Q,\ |p| \leqslant 2n/3\}$.

After at most $2^{n/3}$ changes the set Q and the candidate x become stable. As x can be identified by the number of changes written in exactly $n/3$ bits, we have $C(x) \leqslant n/3 + O(1)$. □

A.2 The proof of Lemma 15

In the proof we will ignore additive error terms of order $O(\delta + \log n)$. Considering P_y and P_x with $O(\log n)$ accuracy, we can ignore all models whose complexity or log-cardinality is not of order $O(n)$.

We have $C(x) = C(y) + m$. Thus the models of the type $\{y'z' : y' \in A,\ z' \in \{0,1\}^m\}$ witness that P_y is included in P_x.

To prove that P_x is included into P_y, we have to transform any i, j-description for x into a model for x whose complexity is at most i and the sum of complexity and log-cardinality is at most $i + j - m$.

To this end consider first the projection of A: $\{y' : |z'| = m,\ y'z' \in A\}$. Unfortunately, this set may be as large as A itself. Reduce it as follows. Consider the yth section of A: $A_y = \{z' : |z'| = m,\ yz' \in A\}$. Define l as the natural number such that $2^l \leqslant |A_y| < 2^{l+1}$. Let A' be the set of those y' whose y'th section has at least 2^l elements. Then by counting arguments we have $|A'| \leqslant 2^{j-l}$.

Then improve A' using a result of [7]:

Lemma 18 (Lemma C.4 on page 3285 in [7]) *For every $A' \ni y$ there is $A'' \ni y$ with $C(A'') \leqslant C(A') - C(A'|y) + O(\log(C(A') + \log|A'|))$ and $\log|A''| = \log|A'|$.*

By this lemma we get a $(C(A') - C(A'|y)), (j - l)$-description of y. We claim that

$$C(A') - C(A'|y) \leqslant C(A) - C(A|y). \tag{1}$$

Indeed, $C(A'|A)$ is negligible hence $C(y|A') \geqslant (y|A)$. This implies that

$$C(y) - C(y|A') \leqslant C(y) - C(y|A).$$

By the symmetry of information this inequality is equivalent to (1). Thus A'' is a $(C(A) - C(A|y)), (j - l)$-description of y. The sum of complexity and log-cardinality of this description is at most

$$C(A) - C(A|y) + j - l \leqslant i - C(A|y) + j - l.$$

We claim that this is less than $i + j - m$, that is, $C(A|y) \geqslant m - l$.

To lower bound l, we will relate it to the conditional complexity of z given y and A. Indeed, we have $C(z|A, y) \leqslant l$, as z can be identified by its ordinal number in yth section of A. On the other hand,

$$C(z|A, y) \geqslant C(z|y) - C(A|y) \geqslant m - C(A|y).$$

A.3 Proof of Theorem 11

The set $S_{m,s}$ can be effectively found from m, s and the number $M = \lfloor \Omega_m/2^{s+1} \rfloor$: we enumerate the list L_m of strings of complexity at most m until $M2^{s+1} + 2^s$ strings are enumerated, the 2^s last enumerated strings form the set $S_{m,s}$.

To find the number M from A and m, s, enumerate strings into the list L_m until all strings from A are enumerated. We claim that the index I of the last enumerated enumerated string from A satisfies the equality $\lfloor I/2^{s+1} \rfloor = M$.

Indeed, starting from the first string from $S_{m,s}$, the index I' of every string x' in the list L_m satisfies the equality $\lfloor I'/2^{s+1} \rfloor = M$. Therefore this equality holds also for the index of x (as $x \in S_{m,s}$) and for indexes of all strings in A which were enumerated after x.

References

1. Bauwens, B., Makhlin, A., Vereshchagin, N., Zimand, M.: Short lists with short programs in short time. ECCC report TR13-007. http://eccc.hpi-web.de/report/2013/007/
2. Gács, P., Tromp, J., Vitányi, P.M.B.: Algorithmic statistics. IEEE Trans. Inform. Th. **47**(6), 2443–2463 (2001)
3. Kolmogorov, A.N.: Talk at the Information Theory Symposium in Tallinn. Estonia (1974)
4. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and its Applications, 2nd edn. Springer, New York (1997)
5. Shen, A.Kh.: Discussion on Kolmogorov complexity and statistical analysis. Comput. J. **42**(4), 340–342 (1999)
6. Shen, A.: Game Arguments in computability theory and algorithmic information theory. Proceedings of CiE, pp. 655–666 (2012)
7. Vereshchagin, N.K., Vitányi, P.M.B.: Kolmogorov's structure functions and model selection. IEEE Trans. Inf. Theory **50**(12), 3265–3290 (2004)
8. Vereshchagin, N.: Algorithmic minimal sufficient statistic revisited. In: 5th Conference on Computability in Europe, CiE 2009, Proceedings, LNCS 5635, pp. 478–487, Heidelberg (2009)
9. Vereshchagin, N.: On Algorithmic Strong Sufficient Statistics. In: 9th Conference on Computability in Europe, CiE 2013, Proceedings, LNCS 7921, pp. 424–433, Milan (2013)