Electronic Colloquium on Computational Complexity, Report No. 43 (2017)

Stochasticity in Algorithmic Statistics for Polynomial Time

Alexey Milovanov* and Nikolay Vereshchagin†

National Research University Higher School of Economics

Abstract

A fundamental notion in Algorithmic Statisticsis that of a stochastic object, i.e., an object having a simple plausible explanation. Informally, a probability distribution is a plausible explanation for x if it looks likely that x was drawn at random with respect to that distribution. In this paper, we suggest three definitions of a plausible statistical hypothesis for Algorithmic Statistics with polynomial time bounds, which are called *acceptability, plausibility and optimality*. Roughly speaking, a probability distribution μ is called an acceptable explanation for x, if x possesses all properties decidable by short programs in a short time and shared by almost all objects (with respect to μ). Plausibility is a similar notion, however this time we require x to possess all properties T decidable even by long programs in a short time and shared by almost all objects. To compensate the increase in program length, we strengthen the notion of 'almost all'—the longer the program recognizing the property is, the more objects must share the property. Finally, a probability distribution μ is called an optimal explanation for x if $\mu(x)$ is large (close to $2^{-\mathrm{C}^{\mathrm{poly}}(x)}$).

Almost all our results hold under some plausible complexity theoretic assumptions. Our main result states that for acceptability and plausibility there are infinitely many non-stochastic objects, i.e. objects that do not have simple plausible (acceptable) explanations. We explain why we need assumptions—our main result implies that P $\neq$ PSPACE. In the proof of that result, we use the notion of an *elusive set*, which is interesting in its own right. Using elusive sets, we show that the distinguishing complexity of a string x can be super-logarithmically less than the conditional complexity of x with condition r for almost all r (for polynomial time bounded programs). Such a gap was known before, however only in the case when both complexities are conditional, or both complexities are unconditional.

It follows from the definition that plausibility implies acceptability and optimality. We show that there are objects that have simple acceptable but implausible and non-optimal explanations. We prove that for strings whose distinguishing complexity is close to Kolmogorov complexity (with polynomial time bounds) plausibility is equivalent to optimality for all simple distributions, which fact can be considered as a justification of the Maximal Likelihood Estimator.

*Supported in part by Young Russian Mathematics award.

†The work of both authors was in part supported by the RFBR grant 16-01-00362 and by the Russian Academic Excellence Project '5-100'.

ISSN 1433-8092

1 Introduction

Acceptable statistical hypotheses

Example 1. Assume we are given an n-bit natural number x which is a square and has no singularities. Which statistical hypotheses we would accept for x? An acceptable hypothesis is the following: the number x was obtained by random sampling in the set of all n-bit squares, where all numbers have equal chances to be drawn (the hypothesis μ_1). An the following hypothesis μ_2 is clearly not acceptable: the number x was obtained by random sampling in the set of *all* n-bit numbers. On what grounds we refute hypothesis μ_2? Because we can exhibit an easily checked property (to be a square) possessed by x and not possessed by a vast majority of all n-bit strings.

The reader can object this line of reasoning by noting that on these grounds we can reject the hypothesis μ_1 as well. Indeed, we exhibit the property "to be equal to x", which is also shared by a negligible fraction of numbers with respect to μ_1. However, in contrast to the property "to be a square", this property is not simple, as it has no short program recognizing the property in a short time. And for the property "to be a square", there is such a program.

Generalizing this example, we will define the notion of an acceptable statistical hypothesis x. A probability distribution μ over the set of binary strings will be called an *acceptable hypothesis* for a binary string x if there is no simple set $T \ni x$ with negligible $\mu(T)$. We will call a set T *simple* if there is a short program to decide membership in T in a short time, as in Example 1.

A string will be called *stochastic*, if it has a simple acceptable hypothesis. How will we measure simplicity of a probability distribution μ? In the same way as we measure the simplicity of a refutation set T: a probability distribution will be called simple, if it can be generated by a short probabilistic machine with no input in a short time. We say that such a machine *generates a distribution* μ, if for all x the probability of the event "M outputs x" equals $\mu(x)$. The running time of M is defined as the maximum of M's running time over all outcomes of its coin tossing.

Of course in a rigorous definition of an acceptable hypothesis μ, we have to specify three parameters: the upper bound α for the length of a program that recognizes T, the upper bound t for the running time of that program, and the upper bound ε for $\mu(T)$ (how small should be $\mu(T)$ to be qualified as "negligible"). The larger these parameters are, the stronger the notion of an acceptable hypothesis is. An in a rigorous definition of a simple distribution μ, we have to specify two parameters: the upper bound α' for the length of a program generating μ and the upper bound t' for the running time of that program. The smaller these parameters are, the stronger the notion of a simple distribution is. Thus in the notion of stochasticity we have 5 parameters, α', t' and α, t, ε. It seems natural to choose $\alpha > \alpha'$ and $t > t'$, that is, to give more resources to those who want to refute a hypothesis μ than the amount of resources needed to generate μ (as it was in Example 1).

Also in the definition of an acceptable hypothesis the parameter ε should be much smaller than $2^{-\alpha}$. In this case the notion of an acceptable distribution

satisfies *The Majority Principle: for every probability distribution μ for almost all (w.r.t. μ) strings x the distribution μ is an acceptable hypothesis for x* (Proposition 2 below). We believe that any notion of a plausible statistical hypothesis should satisfy this principle.

The main question we are interested in is the following: for which values of parameters there are strings that have no simple acceptable explanations? Such strings will be called *non-stochastic*. Our main result states that under assumption NE $\neq$ RE *there are infinitely many non-stochastic strings x for $t, t', 1/\varepsilon = poly(n)$ and $\alpha, \alpha' = O(\log n)$, where $n = |x|$ (Theorem 3).*

In Section 5 we explain why we need complexity theoretic assumptions to prove the main result: we prove that existence of non-stochastic strings for such parameters implies that P$\neq$PSPACE. To prove Theorem 3, we introduce the notion of an *elusive set*. Using that notion, we establish that there is a super-logarithmic gap between Kolmogorov complexity and Distinguishing complexity with polynomial time bounds (similar questions were addressed in [3]). We also study the following two notions of a good statistical hypothesis.

Plausible statistical hypotheses

Example 2. Let $G : \{0,1\}^n \to \{0,1\}^{2n}$ be a Pseudo Random Number Generator (the precise definition is given in Assumption 4 below). Consider a string $x = G(s)$ of length $2n$. Would we accept the uniform distribution over all strings of length $2n$ as a good statistical hypothesis for x? We do not like this hypothesis, as the fraction of x' of length $2n$ for which $x' = G(s')$ for some s' is negligible. However it is impossible to check this property by a short program in short time—for almost all s the uniform distribution over all strings of length $2n$ is an acceptable hypothesis for $G_n(s)$ (Theorem 10). However for every fixed s the property $G_n(s) = x$ can be decided by a *long* program (of length n), into which s is hard-wired.

Let us give up the requirement that the program recognizing T in a short time is short. In a compensation, let us decrease the threshold for $\mu(T)$: we will now think that $\mu(T)$ is negligible if $\log_2 \mu(T)$ is much less than the negative length of the program recognizing T. Notice that in Example 2 we have $\log_2 \mu(T) = -2n$, which is by n less than the negative length of the program recognizing T. Probability distributions satisfying this requirement are called *plausible hypotheses for x*. The definitions imply that every plausible hypothesis is acceptable (Proposition 1). The converse is false (Theorem 10). And again the notion of plausibility satisfies the Majority Principle (Proposition 2).

As plausibility implies acceptability, our main result implies that there are infinitely many strings that have no simple plausible explanations. The existence of such strings can be proved also under other assumptions. Indeed, under Assumption 2 (below) only strings whose distinguishing complexity is close to Kolmogorov complexity can have simple plausible explanations (Proposition 9). And strings with a large gap between these complexities exists under assumption FewP $\cap$ SPARSE $\not\subseteq$ P [3].

Optimal statistical hypotheses

In practice, it is hard to decide whether a given probability distribution μ is plausible or acceptable for a given string x, as there are many possible "refutation sets" T and for a given T it is very hard to check whether it indeed refutes μ or not. Ideally, we would like to have a sound notion of a good hypothesis such that for a given simple distribution μ and a given string x, we could decide whether μ is good for x in a short time. Or, at least to refute μ in a short time, if μ is not good for x.

There is a natural parameter measuring how good is μ as an explanation for x, namely $\mu(x)$. Let us try to use this parameter instead of "refutation sets". According to the new definition, a simple probability distribution μ is a good explanation for x if $\mu(x)$ is large. How large? We will compare $\mu(x)$ with $2^{-\mathrm{C}^t(x)}$, where $\mathrm{C}^t(x)$ denotes Kolmogorov complexity with time bounded by t, where t is large enough compared to the running time of the short probabilistic program generating μ. We will call μ an *optimal hypothesis for* x if $\mu(x) \approx 2^{-\mathrm{C}^t(x)}$.

There are three arguments to justify this definition. Firstly, whatever t we choose, the Majority Principle holds true (Proposition 6). Second, under some complexity theoretic assumption if t is large enough compared to the running time of probabilistic machine generating μ then $\mu(x)$ cannot significantly exceed $2^{-\mathrm{C}^t(x)}$, therefore, if $\mu(x)$ is close to this value, then μ is optimal indeed. And third, given μ, x we can prove in a short time that μ is not an optimal hypothesis for x, if this is the case—it suffices to produce a program p for x such that $\mu(x) \ll 2^{-|p|}$.

Relations between the introduced notions

It follows from the definitions that plausibility implies acceptability and optimality. (To prove the second implication, we let $T = \{x\}$ in the definition of plausibility.) All other statements in the remainder of this section hold true under some assumptions (that are specified later).

For strings x with $\mathrm{CD}^{\mathrm{poly}(n)}(x) \ll \mathrm{C}^{\mathrm{poly}(n)}(x)$ there are no plausible explanations at all (Proposition 9). For such strings we are not aware about any relations between plausibility and optimality.

On the other hand, for strings x with $\mathrm{CD}^{\mathrm{poly}(n)}(x) \approx \mathrm{C}^{\mathrm{poly}(n)}(x)$, the picture is clear: Plausibility = Optimality $\Rightarrow$ Acceptability, and the converse implication does not hold (Example 2, Theorem 10 and Remark 2). The equivalence of plausibility and optimality (Theorem 11) for such strings is a good news, as it provides a justification to the Maximal Likelihood Estimator. Indeed, imagine that x was drawn at random w.r.t. a simple but unknown probability distribution μ. Then with high μ-probability all $\mathrm{C}(x), \mathrm{C}^t(x), \mathrm{CD}^t(x)$ are close to each other and are close to $-\log \mu(x)$ [1] and μ is an acceptable and plausible hypothesis for x (Propositions 2, 6, 5). Given x, we want to find μ or any other plausible or at least acceptable statistical hypothesis for x. Using the Maximal Likelihood Estimator, we choose among all simple hypotheses μ the one

[1] provided that t is larger than certain polynomial of the time needed to generate μ

that maximizes $\mu(x)$. Theorem 11 guarantees the success to this strategy—the chosen hypothesis μ is both acceptable and plausible.

In the next section we provide the rigorous definitions and formulations to all informal definitions and statements mentioned in this Introduction.

2 Our results and their comparison to the previous ones

2.1 Existence of non-stochastic strings

By a technical reason we consider only probability distributions μ over $\{0,1\}^n$ for some n and assume that $\mu(x)$ is a rational number for all x.

Definition 1. Let t, α be natural numbers and ε is a number of the form 2^{-k} for some natural k. A t, α, ε-*acceptable* statistical hypothesis (or *explanation*) for a string x of length n is a probability distribution μ such that $\mu(T) \geqslant \varepsilon$ for all $T \ni x$ recognized by a deterministic program of length less than α in at most t steps for all inputs of length n.

In this definition we are talking about running time of a program on its input. To define rigorously the notion of a program and its running time, we have to fix a *universal Turing machine*. This technical part of the paper is deferred to Section 6.

The larger t, α, ε are, the stronger the notion of t, α, ε-acceptable hypothesis becomes. For every x the distribution concentrated on x is a $*, *, 1$-acceptable hypothesis for x (the asterisk for the time parameter means that the time can be arbitrary large as long as the program always halts). However, we are interested in simple explanations.

Definition 2. A probability distribution μ is called t, α-*simple* if it can be generated by a probabilistic program (with no input) of length less than α in time at most t. [2] (Recall that a machine M generates μ in time t if for all x the probability of event "M outputs x" equals $\mu(x)$ and the running time of M is at most t for all outcomes of coin tossing.)

Strings that have t', α'-simple t, α, ε-acceptable for small t', α' and large t, α, ε are informally called *stochastic* and otherwise *non-stochastic*. The smaller t', α' and the larger t, α, ε are, the stronger the notion of stochasticity is and the weaker the notion of non-stochasticity is.

Definition 3. A probability distribution μ is called a t, ε-*plausible* hypothesis for a sting x of length n, if for any set $T \ni x$ recognized by a program of length l whose running time on all inputs of length n is at most t we have $\mu(T) \geqslant 2^{-l}\varepsilon$.

The following proposition is a straightforward corollary from definitions:

Proposition 1. *Every t, ε-plausible hypothesis for x is $t, \alpha, \varepsilon 2^{-\alpha}$-acceptable for x for any α.*

[2] In Theorem 11 we will need simplicity in another sense: we will need that the function $x \mapsto \mu(x)$ can be computed by a program of length α in time t.

Remark 1. Notice that if μ is t,α-plausible for x then $\mu(x) > 0$. In contrast, t,α,ε-acceptability of μ for x does not imply that in general. However, if the set $T = \{x \mid \mu(x) = 0\}$ can be recognized by a program of length α in time t, then t,α,ε-acceptability for x implies $\mu(x) > 0$. Another reason for not stipulating $\mu(x) > 0$ in the definition of acceptability is that we can achieve this almost 'for free'. Indeed, for every t,α-simple distribution μ the distribution μ' that is the arithmetic mean of μ and the uniform distribution over the set of all strings of length n is t',α'-simple for t',α' close to t,α. For all x of length n we have $\mu'(x) > 0$. If μ is t,α,ε-acceptable for x, then μ' is $t,\alpha,\varepsilon/2$-acceptable for x.

The next statement shows the Majority Principle is valid for t,α,ε-acceptability provided $\varepsilon \ll 2^{-\alpha}$ and for t,ε-plausibility provided $\varepsilon \ll 1/n$.

Proposition 2 (Majority Principle)**.** *For every probability distribution μ over binary strings of length n and all α,ε we have*

$$\mu\{x \mid \mu \text{ is not } *,\alpha,\varepsilon\text{-acceptable for } x\} < \varepsilon 2^{\alpha},$$
$$\mu\{x \mid \mu \text{ is not } *,\varepsilon\text{-plausible for } x\} < \varepsilon(n + O(1)).$$

This proposition as well as all other statements in this section will be proved in Section 4.

Our main result shows that there are infinitely many *non*-stochastic strings x for polynomial values of $t,t',1/\varepsilon$ and logarithmic values of α,α'. This result holds under the following

Assumption 1. For some language L in NP over the unary alphabet there is no probabilistic polynomial time machine that for each string x in L finds a certificate for membership of x in L with probability at least $1/2$. Equivalently, for some language L in NE (the class of languages accepted in time $2^{O(n)}$ by non-deterministic Turing machines) there is no probabilistic machine that for each string x in L in time $2^{O(|x|)}$ finds a certificate for membership of x in L with probability at least $1/2$.

This assumption follows from the assumption NE $\neq$ RE, where RE denotes the class of languages recognized in time $2^{O(n)}$ by probabilistic Turing machines that err with probability at most $1/2$ for all strings in the language and do not err for strings outside the language. It is unknown whether these to assumptions are equivalent or not (see [5]).

Theorem 3. *Under Assumption 1 for some constant d for all c for infinitely many n there is a string of length n that has no $n^c, c\log n$-simple n^d, d, n^{-c}-acceptable hypotheses.*

In other words, for the strings x from this theorem, for every $n^c, c\log n$-simple μ there is $T \ni x$ recognized by a program of length d in time n^d with $\mu(T) < n^{-c}$. The values of parameters in this theorem are chosen so that the Majority Principle holds amply: for any candidate μ the fraction of strings for which μ is not acceptable is less than $2^d n^{-c}$ which is negligible for large c and n. And the resources n^d, d needed to refute a candidate μ can be even smaller than resources $n^c, c\log n$ allowed to generate the candidate μ, as c can be much larger than d.

Later we will compare this result to known results on non-existence of stochastic strings. The latter exists only for $t = t' = *$.

2.2 Super-logarithmic gap between distinguishing complexity and Kolmogorov complexity

In the proof of Theorem 3 we use the notion of an elusive set, which is interesting in its own right.

Definition 4. A language T is called *elusive* if it is decidable in polynomial time and for all c for infinitely many n the following holds. T contains at least one word of length n, however there is no probabilistic machine M without input with program of length at most $c \log n$ and running time at most n^c that with probability at least n^{-c} produces a string of length n from T.

We show that under Assumption 1 there exists an elusive set (Theorem 15). Then we prove that any elusive set has infinitely many non-stochastic strings. There is another interesting corollary from existence of elusive sets: there are infinitely many pairs x, r with $\mathrm{CD}^{\mathrm{poly}(n)}(x|r) \ll \mathrm{C}^{\mathrm{poly}(n)}(x|r)$ (the definition of conditional distinguishing complexity and conditional Kolmogorov complexity is given in Section 6). More specifically, the following holds.

Theorem 4. *Under Assumption 1 for some constant d for all c there are infinitely many strings x with*

$$\mathrm{CD}^{n^d}(x|r) \leqslant \mathrm{C}^{n^c}(x|r) - c \log n$$

for 98% of r's of length n^d. Here n stands for the length of x. Moreover, under Assumption 2 (see below), in the left hand side *of the last inequality, we can replace the conditional complexity by the unconditional one:*

$$\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x|r) - c \log n.$$

Assumption 2. There is a set that is decidable by deterministic Turing machines in time $2^{O(n)}$ but is not decidable by Boolean circuits of size $2^{o(n)}$ for almost all n.

The existence of pairs x, r satisfying the first part of Theorem 4 is known to be equivalent to the impossibility to separate in polynomial time non-satisfiable Boolean formulas from those having the unique satisfying assignment [3]. The latter statement (denoted by $(1SAT, SAT) \notin \mathrm{P}$) follows from the assumption NP $\neq$ RP, which is weaker than Assumption 1, using Valiant and Vazirani Lemma [11].[3] For unconditional complexity, previously it was known that there are strings with $\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x) - c \log n$ under the assumption FewP $\cap$ SPARSE $\not\subseteq$ P [3]. Thus the first part of Theorem 4 is not new, however its second part is.

[3] Thus if $(1SAT, SAT) \in \mathrm{P}$ then there are no elusive sets. Is the inverse true?

2.3 A comparison of the notions of acceptability, plausibility and optimality

Definition 5. A probability distribution μ is called t,ε-*optimal for* x, if

$$\mu(x) > \varepsilon 2^{-\,\mathrm{C}^t(x)}.$$

The larger t,ε are, the stronger the notion of t,ε-optimality is. Assume that the distribution μ is t',α-simple for a small α. We will explain that this definition makes sense for values of t which are larger than some polynomial of $|x|+t'$ and does not make any sense if, conversely, t' is larger than some polynomial of $|x|+t$.

Consider the following

Assumption 3. There is a set which is decidable by deterministic Turing machines in time $2^{O(n)}$ but is not decidable by deterministic Turing machines in space $2^{o(n)}$ for almost all n.

Proposition 5. *Under Assumption 3 the following holds. There is a constant d such that for all t,α-simple probability distribution μ for all strings x of length n,*

$$\log_2 \mu(x) \leqslant -\,\mathrm{C}^{(n+t)^d}(x) + \alpha + d\log(n+t).$$

Assume that μ is a t,ε-optimal t',α-simple hypothesis for x and $t > (n+t')^d$ where d is the constant from Proposition 5. Then $\log_2 \mu(x)$ differs from the maximal possible value of $\log_2 \mu'(x)$ for t',α-simple hypotheses μ' by at most $\alpha+\log(1/\varepsilon)+d\log(n+t')$. This fact provides some justification for the notion of optimality. Another justification for the definition is the validity of the Majority Principle:

Proposition 6. *For some constant c for all n and all strings x of length n for all probability distributions μ we have*

$$\mu\{x \mid \mu \textit{ is not } *,\varepsilon\textit{-optimal for } x\} < \varepsilon(n+c).$$

Conversely, if t' is larger than some polynomial of $|x|+t$ then *for all strings* there is a simple optimal hypothesis (and thus the notion of optimality becomes trivial).

Proposition 7. *There is a constant c such that for all t every string x of length n has a $(n+t)^c, c\log(n+t)$-simple $t,1$-optimal hypothesis.*

Letting $T=\{x\}$ in the definition of plausibility we can see that plausibility implies optimality:

Proposition 8. *For all strings x and for all t,ε-plausible hypotheses μ for x we have* $\log_2 \mu(x) \geqslant -\,\mathrm{CD}^t(x) + \log_2 \varepsilon \geqslant -\,\mathrm{C}^t(x) + \log_2 \varepsilon - O(1)$.

By Proposition 5 the first inequality in this proposition implies the following

Proposition 9. *Under Assumption 3 there is a constant d such that for every string x of length n that has a t_1,α-simple t_2,ε-plausible hypothesis we have*

$$\mathrm{C}^{(n+t_1)^d}(x) \leqslant \mathrm{CD}^{t_2}(x) + \alpha + \log_2(1/\varepsilon) + d\log(n+t_1).$$

Therefore strings with large gap between distinguishing complexity and Kolmogorov complexity do not have simple plausible explanations. From the result of [3] cited above it follows that (under some complexity theoretic assumptions) for some d for every c there are infinitely many strings x without $n^c, c\log n$-simple n^d, n^{-c}-plausible hypotheses (where n denotes the length of x).

Thus plausibility implies acceptability and optimality. Is there any implication in the reverse direction? Assuming existence of a Pseudo Random Number Generator $G : \{0,1\}^n \to \{0,1\}^{2n}$ we can show that acceptability does not imply neither plausibility, nor optimality.

Assumption 4. (Existence of PRNG) There is a polynomial time computable function $G : \{0,1\}^* \to \{0,1\}^*$, such that $|G(s)| = 2|s|$ and for every sequence of Boolean circuits $\{C_n\}$ with $2n$ inputs and 1 output such that the size of C_n is bounded by a polynomial of n, the difference of probabilities of events $C_n(G(s)) = 1$ and $C_n(r) = 1$ tends to 0 faster than every inverse polynomial of n (that is, for any polynomial p for all sufficiently large n the difference of probabilities is less than $1/p(n)$). We assume here the uniform distributions over strings s and r of length n and $2n$, respectively.

Theorem 10. *Assume that there is PRNG $G : \{0,1\}^n \to \{0,1\}^{2n}$, as in Assumption 4. Then for all c for all sufficiently large n for 99% of strings s of length n the uniform distribution over strings of length $2n$ is a $n^c, c\log n, n^{-c}/200$-acceptable hypothesis for $G_n(s)$.*

Remark 2. Note that the uniform distribution is neither optimal ($\mathrm{C}^{\mathrm{poly}(n)}(x) \leqslant n + O(1)$, and $\log \mu(x) = -2n$), nor plausible (recall Example 2) hypothesis for x. By counting arguments for almost all s for $x = G_n(s)$ it holds $\mathrm{CD}^{\mathrm{poly}(n)}(x) \approx \mathrm{C}^{\mathrm{poly}(n)}(x) \approx \mathrm{C}(x) \approx n$. Therefore, there are strings satisfying Theorem 10 and having the latter property.

Finally, for simple hypotheses μ and for strings with $\mathrm{CD}^{\mathrm{poly}}(x) \approx \mathrm{C}^{\mathrm{poly}}(x)$ optimality implies plausibility and hence acceptability. However this time we need that μ can be computed rather than generated in a short time by a short program.

Theorem 11. *Under Assumption 2 there is a constant c such that the following holds true. Let μ be a probability distribution μ such that the function $x \mapsto \mu(x)$ can be computed by a program of length α in time t. Assume further that $\mu(x) > \varepsilon 2^{-\mathrm{CD}^{(n+t+t_1)^c}(x)}$, where n is the length of x and t_1 an arbitrary number. Then μ is a $t_1, \varepsilon 2^{-\alpha - c\log n}$-plausible hypothesis for x.*

Notice that in this theorem instead of $(n + t + t_1)^c, \varepsilon$-optimality we use a stronger condition $\mu(x) > \varepsilon 2^{-\mathrm{CD}^{(n+t+t_1)^c}(x)}$ (with distinguishing complexity in place of Kolmogorov complexity). However for strings x and t_2 with $\mathrm{C}^{t_2}(x) \leqslant \mathrm{CD}^{(n+t+t_1)^c}(x) + \beta$ we can replace that condition by the condition of $t_2, \varepsilon 2^{\beta}$-optimality of μ for x. Informally speaking, if $\mathrm{C}^{\mathrm{poly}}(x) \approx \mathrm{CD}^{\mathrm{poly}}(x)$ then optimality for x implies plausibility for x.

2.4 Non-stochastic strings in classical Algorithmic Statistics

In Algorithmic Statistics without resource bounds [4, 6, 7, 8, 12, 13] plausibility of a statistical hypothesis μ for x is measured by one parameter $-\log\mu(x) - \mathrm{C}(x|\mu)$, called *randomness deficiency of x w.r.t. μ*. Probability distributions can be represented by the lists of pairs (a string, its probability) ordered in a specific way. Thus we can talk on conditional Kolmogorov complexity $\mathrm{C}(x|\mu)$ and of Kolmogorov complexity $\mathrm{C}(\mu)$ of μ itself. Up to an additive constant $\mathrm{C}(\mu)$ coincides with the length of the shortest program generating μ (assuming that the program always halts).

Neglibibility of randomness deficiency is similar to all three our definitions of a good hypothesis. More specifically the inequality $-\log\mu(x) - \mathrm{C}(x|\mu) < \beta$ is similar to saying that μ is $*, \alpha, 2^{-\beta}$-acceptable, $*, 2^{-\beta}$-plausible and $*, \alpha, 2^{-\beta}$-optimal for x. However there is an important difference. The inequality $-\log\mu(x) - \mathrm{C}(x|\mu) < \gamma$ implies that μ is $*, \gamma + O(1)$-optimal for x, but not the other way around. If $-\log\mu(x) - \mathrm{C}(x|\mu) < \gamma$ then for every set $T \ni x$ *accepted by a non-deterministic* program p we have $\mu(T) > 2^{-|p|-\gamma}$. Conversely, if $-\log\mu(x) - \mathrm{C}(x|\mu) \geqslant \gamma$, then there is a set $T \ni x$ *accepted by a short non-deterministic* program (of length about $\mathrm{C}(\mu)$) with $\mu(T) \leqslant 2^{-\gamma}$.

In contrast, both the notion of $*, \gamma$-plausibility and the notion of $*, \alpha, \varepsilon$-acceptability are defined by means of *deterministic* recognizing machines. Thus $-\log\mu(x) - \mathrm{C}(x|\mu) < \gamma$ implies $*, \gamma$-plausibility but not the other way around (with logarithmic accuracy: the inequality $-\log\mu(x) - \mathrm{C}(x|\mu) < \gamma$ implies only $*, \gamma + O(\log n)$-plausibility.)

A string x is called *Kolmogorov α, β-stochastic* if there is a probability distribution μ with $\mathrm{C}(\mu) \leqslant \alpha$ and $-\log\mu(x) - \mathrm{C}(x|\mu) \leqslant \beta$. As we have just explained, Kolmogorov α, β-stochasticity implies the existence of a $*, \alpha$-simple $*, \varepsilon$-plausible (and hence $*, \alpha_2, \varepsilon + \alpha_2$-acceptable for all α_2) hypothesis. (Again we ignore logarithmic terms.)

Shen proved the existence of Kolmogorov α, β-non-stochastic string for α, β that are linear in n:

Theorem 12 ([10])**.** *For some constant c for all n and all α, β, with $2\alpha + \beta < n - c\log n$, there is a Kolmogorov α, β-non-stochastic string of length n.*

As we have mentioned, this statement does not imply the existence of non-stochastic strings in our sense (even for very large values of time parameters). However the techniques of [10] can be used to prove the following:

Theorem 13. *For all n and all α, β with $\alpha + \beta < n$ there is a string of length n that has no $*, \alpha$-simple $*, \alpha + O(\log n), 2^{-\beta}$-acceptable hypotheses.*

It is not hard to see that Theorem 13 implies Theorem 12. Later the term 2α in Theorem 12 was replaced with α:

Theorem 14 ([13])**.** *For some constant c for all n and all α, β, with $\alpha + \beta < n - c\log n$, there is a Kolmogorov α, β-non-stochastic string of length n.*

This result is optimal up to logarithmic terms. Indeed, for all x of length n and all $\alpha \leqslant n$ the uniform distribution μ over strings of length n that have

the same α first bits as x, has complexity about α and randomness deficiency at most $n - \alpha$:

$$-\log \mu(x) - \mathrm{C}(x|\mu) = n - \alpha - \mathrm{C}(x|\mu) \leqslant n - \alpha.$$

So using the known methods we can show the existing of strings of length n that have no $*, \alpha_1$-simple $*, \alpha_2, \varepsilon$-acceptable hypotheses for $\alpha_1, \log(1/\varepsilon) = \Omega(n)$ and for α_2 which are only logarithmically larger than α_1. It is essential for those methods that the running time can be arbitrary large and hence they cannot be used in the case when the running time is bounded by a polynomial of the length.

The notion of an optimal hypothesis is also borrowed from the classical Algorithmic Statistics. A distribution μ with small Kolmogorov complexity is called optimal if $\log \mu(x)$ is close to $-\mathrm{C}(x)$, which is equivalent to saying that the randomness deficiency is small. However, optimality was studied also for distribution μ with large Kolmogorov complexity, in which case optimality was defined as $\log \mu(x) \approx \mathrm{C}(\mu) - \mathrm{C}(x)$. Using the Symmetry of Information, we can show that the randomness deficiency always does not exceed the 'optimality deficiency' $\mathrm{C}(\mu) - \mathrm{C}(x) - \log \mu(x)$, but not the other way around [13]. However in the definition of Kolmogorov stochasticity, we can use the optimality deficiency instead of randomness deficiency: for a string of length n there is an $*, \alpha$-simple hypothesis with optimality deficiency less than β if and only if the string is Kolmogorov α, β-stochastic. More accurately, both directions 'if' and 'only if' hold up to adding some terms of order $O(\log n)$ to parameters α, β [13].

3 Open questions

Question 1. *Under which other assumptions (different from NP $\neq$ RP) there are non-stochastic strings and elusive sets? Under which other assumptions (different* $(1SAT, SAT) \in \mathrm{P}$ *and* P =PSPACE*) there are no elusive sets and all strings are stochastic?*

Question 2. *Let us replace in the definitions of a plausible and acceptable hypothesis deterministic machines by non-deterministic ones. Do the notions of a plausible and acceptable hypothesis and of stochastic string become stronger?*

Question 3. *Are there strings that do not possess simple optimal hypotheses?*

Question 4. *How acceptability is related to optimality for strings x with* $\mathrm{CD}^{poly}(x) \ll \mathrm{C}^{poly}(x)$*?*

Question 5. *Are there non-stochastic strings with polynomial bounds for time and linear bounds for program length: is it true that for some c and $\varepsilon < 1$ for all d and all $\delta < 1$ for all but finitely many n every string x of length n has an $n^c, \varepsilon n$-simple $n^d, \delta n, n^{-c}$-acceptable hypothesis?*

4 The proofs

4.1 Proof of Proposition 2

The first inequality: the number of sets recognized by a program of length less than α is less than 2^α and each such set contains a fraction at most ε of all n-bit strings w.r.t. μ.

The second inequality: w.l.o.g. we may consider only sets T recognized by programs of length less than $n + c$ (for some constant c). Indeed, assume that a set $T \ni x$ witnesses implausibility of μ for x and is recognized by a program of length $l > n + c$. Then $\mu(x) \leqslant \mu(T) \leqslant \varepsilon 2^{-n-c}$. Thus the set $\{x\}$, whose complexity is less than $n + c$, witnesses implausibility of μ for x (if c is large enough). Then we can repeat the arguments from the previous paragraph: for every fixed l any set T recognized by a program of length l refutes a fraction at most $\varepsilon 2^{-l}$ of all strings and the number of programs of length l is 2^l, thus all together they refute a fraction at most ε of strings of length n.

4.2 Proof of Theorem 3

Theorem 15. *Under Assumption 1 there exists an elusive set.*

Proof. Fix a language L over the unary alphabet $\{1\}$ that belongs to $\mathrm{NP} \setminus \mathrm{RP}$. Since $L \in \mathrm{NP}$ it can be represented in the form

$$L = \{1^k \mid \exists x \in \{0,1\}^{k^c}\ R(1^k, x)\},$$

where $c > 0$ is a natural number and R a relation decidable in time $\mathrm{poly}(k)$.

Consider the set

$$T = \{x \in \{0,1\}^{k^c} \mid R(1^k, x)\}.$$

Obviously T can be recognized in polynomial time. Let us show that T is elusive.

Let d be any constant. For the sake of contradiction assume that for some m for all $k > m$ with $1^k \in L$ there is a program M_k of length $d \log k^c$ that, with probability at least k^{-cd}, in time k^{cd} prints a string from T of length k^c. To obtain a contradiction we construct the following probabilistic algorithm that recognizes L with one-sided bounded error in polynomial time:

The algorithm. On input 1^k we run all randomized programs of length $d \log k^c$ in time k^{cd}. Each program is run k^{cd} times. For each string x output by any of those programs we check the equality $R(1^k, x) = 1$. If the equality holds true for at least one of those x's, we say that $1^k \in L$, and otherwise we say that $1^k \notin L$. (The end of the Algorithm.)

This algorithm runs in polynomial time and can err only on inputs $1^k \in L$. Let us bound the probability of error on any such inputs. We assume that for all $1^k \in L$ with $k > m$ there is a randomized program of length $d \log k^c$ that produces a string from T with probability at least k^{-cd}. The probability that k^{cd} times its output falls outside T is less than $(1 - k^{-cd})^{k^{cd}} \leqslant 1/e$. Therefore for all $k > m$ the algorithm errs with probability at most $1/e$. Hence $L \in \mathrm{RP}$, which is a contradiction. □

Proof of Theorem 3. By Theorem 15 there is an elusive set T. For some d there is a machine with program of length at most d recognizing T in time n^d.

Let $T^{=n}$ denote the set of all strings of length n from T. For every n such that $L^{=n} \neq \emptyset$ pick any string x_n from $L^{=n}$. We claim that for any constant c for infinitely many n the string x_n does not have $n^c, c\log n$-simple n^d, d, n^{-c}-acceptable hypotheses.

For the sake of contradiction assume that for some m for all $n > m$ there is such hypothesis μ_n. As $x_n \in T$ and T is recognized by a program of length at most d in time n^d we have $\mu(T) \geqslant n^{-c}$. Thus for each such n the probabilistic program of length less than $c\log n$ generating the distribution μ_n in time n^c produces a string from T with probability at least n^{-c}, which contradicts the assumption that T is elusive. □

4.3 Proof of Theorem 4

Proposition 16. *Assume that L is an elusive set. Then for all constants c there is a constant d such that there infinitely many $x \in L$ with*

$$\mathrm{CD}^{n^d}(x|r) \leqslant \mathrm{C}^{n^c}(x|r) - c\log n$$

for 99% of strings r of length n^d. Here n denotes the length of x.

This proposition follows from Sipser's lemma.

Lemma ([9])**.** *For every language L recognizable in polynomial time there is a constant d such that for all n for 99% of strings r of length n^d and all $x \in L^{=n}$ we have*

$$\mathrm{CD}^{n^d}(x|r) \leqslant \log|L^{=n}| + d\log n.$$

Proof Proposition 16. Let d be the constant from Sipser's Lemma applied to the given elusive language L. Let c be an arbitrary constant. By Sipser's lemma it suffices to show that $\log|L^{=n}| + d\log n$ is less than the right hand side of the inequality we have to prove. More precisely, we have to show that

$$\log|L^{=n}| + d\log n \leqslant \mathrm{C}^{n^c}(x|r) - c\log n$$

for infinitely many $x \in L$ and for 99% of r of length n^d.

For the sake of contradiction assume that for some m for all $n > m$ for all $x \in L^{=n}$ we have

$$\mathrm{C}^{n^c}(x|r) < \log|L^{=n}| + (c+d)\log n$$

for at least 1% of r's. For any such n consider the program M_n of probabilistic machine that samples a random string w of length less that $\log|L^{=n}| + (c+d)\log n$ (all such strings are equiprobable) and a string r of length n^d. Then M_n considers w as a program of a string conditional to r, runs that program in n^c steps and outputs its result (if any). Thus M_n outputs every $x \in L^{=n}$ with probability at least $1/(100|L^{=n}|n^{c+d})$. And hence for all $n > m$ with non-empty $L^{=n}$ the output M_n falls in $L^{=n}$ with probability at least

$$|L^{=n}|/(100|L^{=n}|n^{c+d}) = 1/100n^{c+d}.$$

This contradicts the assumption that L is an elusive set, as M_n runs in time $\mathrm{poly}(n)$ and its program length is $O(\log n)$. □

The first part Theorem 4 directly follows from Theorem 15 and Proposition 16. Let us prove the second part of Theorem 4. In [14], it was shown that under Assumption 2 we can replace in Sipser's lemma conditional complexity by the unconditional one.

Theorem 17 (Theorem 3.2 in [14]). *Under Assumption 2 for all $L \in$ PSPACE/*poly *there is a constant d such that for all $x \in L^{=n}$ we have*

$$\mathrm{CD}^{n^d, L^{=n}}(x) \leqslant \log |L^{=n}| + d \log n.$$

Moreover the constant d depends only on the length of the advice string for L and on the space bound for L.

In the notation $\mathrm{CD}^{n^d, L^{=n}}(x)$ the superscript $L^{=n}$ means that the distinguishing program is granted the access to an oracle for $L^{=n}$. If L is decidable on polynomial space we can drop this superscript.

Combining this theorem with the proof of Proposition 16 and Theorem 15 we obtain the proof of the second part of Theorem 4.

Remark 3. In [3], a weaker result is derived from an assumption that is not comparable with our one:

Theorem 18 ([3]). *Assume that* $\mathrm{FewP} \cap \mathrm{SPARSE} \not\subseteq \mathrm{P}$. *Then for some constant d for all c for infinitely many x we have*

$$\mathrm{CD}^{n^d}(x) < \mathrm{C}^{n^c}(x) - c \log n.$$

Here n denotes the length of x.

Remark 4. In [3], the following relation between (1 SAT, SAT) and distinguishing complexity was discovered:

Theorem 19 ([3]). *The following are equivalent:*
(1) $(1SAT, SAT) \notin \mathrm{P}$.
(2) *For some d for all c there are x and y with*

$$\mathrm{CD}^{(|x|+|y|)^d}(x|y) \leqslant \mathrm{C}^{(|x|+|y|)^c}(x|y) - c \log(|x| + |y|).$$

From Theorem 19 and Theorem 4 we obtain the following implication $\mathrm{NE} \not\subseteq \mathrm{RE} \Rightarrow (1SAT, SAT) \notin \mathrm{P}$, which is not surprising since $(1SAT, SAT) \in \mathrm{P}$ implies NP = RP using the Valiant–Vazirani Lemma.

4.4 Proof Proposition 5

Definition 6. A probability distribution σ over $\{0,1\}^*$ is called P-samplable, if there is a program of randomized machine that generates this distribution in time bounded by a polynomial of the length of the output.

Theorem 20 (Lemma 3.2 in [1]). *Under Assumption 3 for every* P*-samplable probability distribution σ there is a d such that for all x of length n,*

$$\mathrm{C}^{n^d}(x) \leqslant -\log \sigma(x) + d \log n.$$

Proof of Proposition 5. Assume that μ is generated by a program q of length less than α in time t. Assume that $\alpha < n$ as otherwise the statement is obvious (the complexity of x with a polynomial time bound does not exceed its length).

Consider the following P-samplable probability distribution σ: we choose a random t with probability proportional to $1/t^2$, then we choose a random program q' of a randomized machine with probability proportional to $2^{-|q'|}/|q'|^2$, run that program in in t steps and output the triple $(1^t, q', x)$, where x is the result of q' (if any, and the empty string otherwise). The triple $(1^t, q', x)$ is encoded in such a way that the code length be polynomial in $t + |q'| + |x|$. By Theorem 20

$$\mathrm{C}^{|y|^d}(y) \leqslant -\log \sigma(y) + d \log |y|$$

for some constant d and all y. Letting $y = (1^t, q, x)$, we obtain

$$\begin{aligned}\mathrm{C}^{|(1^t,q,x)|^d}(1^t, q, x) &\leqslant -\log \sigma(1^t, q, x) + c \log |(1^t, q, x)| \\ &\leqslant 2 \log t + \alpha + 2 \log \alpha - \log \mu(x) + O(\log(t + |x|)).\end{aligned}$$

Since the complexity of any entry of a tuple does not exceed the complexity of the tuple itself, we get the sought inequality. □

4.5 Proof of Proposition 6

Indeed, if μ is not $*, \beta$-optimal for x, then $\mu(x) < 2^{-\beta - K(x)}$. The sum of probabilities of all such words is less than ε times the sum of $2^{-\mathrm{C}(x)}$ over all x of length n. The latter sum is less than $n + O(1)$, since $\mathrm{C}(x) \leqslant n + O(1)$ for all x of length n and for all fixed k the sum of $2^{-\mathrm{C}(x)}$ over all x with $\mathrm{C}(x) = k$ is at most 1 (there are at most 2^k such x's).

4.6 Proof of Proposition 7

Consider the machine that chooses a random program of length $\mathrm{C}^t(x)$ (with uniform distribution), runs it in t steps and outputs its result (if any). The program of this machine has length $O(\log(n + t))$ and its running time is bounded by a polynomial in $t + n$. With probability at least $2^{-\mathrm{C}^t(x)}$ that machine prints x hence it generate a probability distribution μ that is $\mathrm{poly}(n + t), 1$-optimal for x.

4.7 Proof of Theorem 10

Fix an arbitrary constant c. For any set T_n of strings of length $2n$ recognizable by a program of length less than $c \log n$ in time n^c we can construct a Boolean circuit C_n recognizing that set whose size is bounded by a polynomial of n (that polynomial depends only on c). Therefore there is a function $\varepsilon(n)$ that tends to 0 faster than any inverse polynomial of n and such that the probabilities of events $G_n(s) \in T_n$ and $r \in T_n$ differ at most by $\varepsilon(n)$ for any such set T_n.

For the sake of a contradiction assume that for infinitely many n for 1% of strings s of length n there is a set T_s recognizable by a program of length less than $c \log n$ in time n^c with $\Pr[r \in T_s] < n^{-c}/200$. Consider the union $\mathcal{T}_n$ of all

such sets. Since $G_n(s) \in \mathcal{T}_n$ for all such s, the probability of event $G_n(s) \in \mathcal{T}_n$ is at least $1/100$. On the other hand, the probability of event $r \in \mathcal{T}_n$ is less than $2^{c \log n}$ (the number of sets T_s) times $n^{-c}/200$, which equals $1/200$. Thus the difference of probabilities of events $r \in \mathcal{T}_n$ and $G_n(s) \in \mathcal{T}_n$ is greater than $1/200$.

Recall now that for each n probabilities of events $G_n(s) \in T_s$ and $r \in T_s$ differ by at most $\varepsilon(n)$, which tends to 0 faster than any inverse polynomial of n. The number of T_s is less than $2^{c \log n}$. Thus we obtain the inequality $2^{c \log n}\varepsilon(n) > 1/200$ for infinitely many n, which is a contradiction.

4.8 Proof of Theorem 11

First we derive a corollary from Theorem 17.

Corollary 21. *Under Assumption 2 for some constant d for all n for every program q of length at most $3n$ that recognizes a set $T \subset \{0,1\}^n$ in time t, for all $x \in T$ we have*

$$\mathrm{CD}^{(n+t)^d}(x) \leqslant \log |T| + |q| + d \log(n+t).$$

Proof. Fix any sequence of strings q_n with $q_n \leqslant 3n$. Let

$$L = \bigcup_{n,t} T_{n,t}, \text{ where } T_{n,t} = \{0^{[n,t]-n}1x \mid |x| = n,\ q_n(x) = 1 \text{ in time } t\}.$$

Here $[n,t]$ — denotes a polynomial computing a 1-1-mapping from pairs of natural numbers to natural numbers such that $[n,t] \geqslant n,t$. Given the length of any word from $T_{n,t}$ we can compute n and t in polynomial time. Therefore $L \in \mathrm{P}/3n$ and $L^{=([n,t]+1)} = T_{n,t}$. Hence we can apply Theorem 17 to L and conclude that

$$\mathrm{CD}^{([n,t]+1)^d, q_n}(0^{[n,t]-n}1x) \leqslant \log |T_{n,t}| + d \log(n+t)$$

for all t, for all x of length n and all sequences $\{q_n\}$ as above. The constant d does not depend on $\{q_n\}$, therefore this inequality holds for all n,t, for all x of length n and all $q \leqslant 3n$. Plug into this inequality t, q, n, x from the conditions of theorem. We obtain

$$\mathrm{CD}^{([n,t]+1)^d, q}(0^{[n,t]-n}1x) \leqslant \log |T| + d \log(n+t).$$

It remains to append to the program of this length distinguishing $0^{[n,t]-n}1x$ from other strings the information about n,t and q. In this way we get a distinguishing program for x of length $\log |T| + |q| + O(\log(n+t))$ with running time $\mathrm{poly}(n,t)$ that does not need an oracle for T. □

Proof of Theorem 11. Assume the contrary: there is $T \ni x$ recognizable by a program of length l in time t_1 with $\mu(T) < \varepsilon 2^{-l-\alpha-c \log n}$ (where the constant c will be chosen later). Then consider the set $T' = \{x' \in T \mid \mu(x') \geqslant 2^{-i}\}$ where $-i$ stands for the integer part of the binary logarithm of $\mu(x)$. This set has at most $\mu(T)2^i \leqslant 2\mu(T)/\mu(x)$ strings (of length n) and can be recognized in

time $t+t_1$ by a program of length $\alpha + l + O(\log\log(1/\mu(x)))$. W.l.o.g. we may assume that $\mu(x) \geqslant 2^{-n}$ and that $\alpha, l \leqslant n$. Thus the length of that program is less than $3n$. Corollary 21 implies that for some constant d

$$\mathrm{CD}^{(n+t+t_1)^d}(x) \leqslant \log\mu(T) - \log\mu(x) + \alpha + l + d\log n,$$

that is,

$$\begin{aligned}\log\mu(x) &\leqslant -\mathrm{CD}^{(n+t+t_1)^d}(x) + \alpha + l + \log\mu(T) + d\log n\\ &\leqslant -\mathrm{CD}^{(n+t+t_1)^d}(x) + \alpha + l + \log\varepsilon - l - \alpha - c\log n + d\log n\\ &= -\mathrm{CD}^{(n+t+t_1)^d}(x) + \log\varepsilon - c\log n + d\log n.\end{aligned}$$

Let $c = d+1$. Then the last inequality implies that

$$\log\mu(x) < -\mathrm{CD}^{(n+t+t_1)^d}(x) + \log\varepsilon \leqslant -\mathrm{CD}^{(n+t+t_1)^c}(x) + \log\varepsilon,$$

which contradicts the condition of the theorem. □

4.9 Proof of Theorems 13 and 12

Proof of Theorem 13. For every μ generated by a program of length $< \alpha$ consider the set of all x' satisfying the inequality $\mu(x') \geqslant 2^{-\beta}$. For any fixed μ there at most 2^β such x' (otherwise the sum of their probabilities would exceed 1). Therefore the total number of strings in all such sets is less than

$$2^\alpha 2^\beta < 2^n.$$

Here the first factor is an upper bound for the number of μ and the second factor the number of x' for a fixed μ.

Let x by the lex first string of length n outside all such sets. Its Kolmogorov complexity is at most $\alpha + O(\log n)$, as we can find it from the number N of distributions μ generated by a program of length $< \alpha$ and from parameters α, β (from α and N we can find all such distributions by running in parallel all programs of length less than α until we find N distributions; then for every of the distributions μ we can find the set of strings x' with $\mu(x') \geqslant 2^{-\beta}$).

Let us show that x does not possess $*, \alpha$-simple $*, \alpha + O(\log n), 2^{-\beta}$-acceptable hypotheses. Assume that μ is generated by a program of length $< \alpha$. Consider the set $T = \{x\}$. Its complexity is at most $\alpha + O(\log n)$ and μ-probability is less than $2^{-\beta}$ by construction. Hence the set T witnesses that μ is not acceptable for x. □

Proof of Theorem 12. Assume that α', β' satisfy the inequality $2\alpha' + \beta' + c\log n < n$ where c is the constant hidden in the O-notation in Theorem 13 (actually a little larger). Let in Theorem 13 $\alpha = \alpha'$ and $\beta = \beta' + \alpha' + c\log n$. If the word x existing by Theorem 13 were Kolmogorov α', β'-stochastic, then it would have $*, \alpha'$-simple $*, \alpha_2, 2^{-\beta'-\alpha_2}$-acceptable hypothesis for all α_2. Letting $\alpha_2 = \alpha + c\log n$, we would derive that x has an $*, \alpha$-simple $*, \alpha + c\log n, 2^{-\beta}$-acceptable hypothesis, which contradicts the statement of Theorem 13. □

5 Non-stochastic strings and P = PSPACE

In this section we show why we need some complexity-theoretic assumption in Theorem 3—its statement implies P $\neq$ PSPACE.

Theorem 22. *Assume that* P=PSPACE. *Then for every c there is d for which every string of length n has $n^d, 2\log n$-simple $n^c, c\log n, n^{-d}$-acceptable hypothesis.*

Proof. Fix a constant c. Define sets $A_0, A_1, \ldots$ by the recursion: $A_0 = \{0,1,\}^n$ and for $i > 0$ let

$$A_i = \{x \in A_{i-1} \mid (\exists\ n^c, c\log n, n^{-c}\text{-simple } T \ni x)\ |T \cap A_{i-1}| \leqslant 2^n n^{-i-c}\}.$$

The definition of A_i implies that it has at most $2^n n^{-i}$ words. Indeed, the number of $n^c, c\log n$-simple sets is less than 2^c and each of them contributes at most $2^n n^{-i-c}$ strings to A_i.

Since A_n is empty, for every string x of length n there is $i \leqslant n$ with $x \in A_i \setminus A_{i+1}$. For a given x fix that i and consider the distribution μ_i generated as follows. Sample a random $j \leqslant 2^n n^{-i}$ and output jth in the lexicographical order word from A_i (if there is no such word, then the last one, say).

Assume that there is an $n^c, c\log n$-simple $T \ni x$ with $\mu_i(T) < n^{-d}$. The probability of each string from A_i is at least $2^{-n}n^i$ hence we have

$$|A_i \cap T| < n^{-d}2^n n^{-i} \leqslant 2^n n^{-(i+1)-c}$$

(the last inequality holds if $d \geqslant c+1$). Hence $x \in A_{i+1}$, which contradicts the choice of i.

It remains to show that μ_i is $n^d, 2\log n$-simple provided d is large enough. The distribution μ_i can be identified by numbers n, i, hence there is a program of length $2\log n$ generating μ_i. Given the index of a string x in A_i we can find x on the space polynomial in n and n^c . Under assumption P=PSPACE, we can do it in time polynomial in n and n^c and hence μ_i is $n^d, 2\log n$-simple for some d. □

In this theorem the time (n^d) to generate distribution μ which is an $n^c, c\log n, n^{-d}$-acceptable hypothesis for x can be much larger than the time (n^c) allowed to refute μ. Does a similar statement hold for d that does not depend on c? The next theorem answers this question in the negative.

Theorem 23. *Assume that* P=PSPACE. *Then for some constant e for every n, α, t there is a string of length n that has no t, α-simple $(\alpha + t + n)^e, \alpha + 2\log t + 2\log n, 2^{-n+\alpha}$-acceptable statistical hypotheses.*

Plugging $t = n^d$ and $\alpha = d\log n$ we get, for each n, a string of length n with no $n^d, d\log n$-simple $n^{ed}, (2d+2)\log n, 2^{-n+d\log n}$-acceptable hypotheses. Thus for any d for some $c = O(d)$ there are infinitely many strings which have no $n^d, 2\log n$-simple $n^c, c\log n, n^{-d}$-acceptable hypotheses.

Proof. Let μ_p^t denote the probability distribution generated by a program p in time t. Consider the arithmetic mean of all t, α-simple distributions: $\mu(x) = 2^{-\alpha} \sum_{|p|<\alpha} \mu_p^t(x)$. Let x stand for the lex first string of length n such that $\mu(x) \leqslant 2^{-n}$. This string can be found on space $\mathrm{poly}(n+t+\alpha)$. Using the assumption P=PSPACE we conclude that x can be found in time $p(n+t+\alpha)$ from t, α, n, where p is a polynomial.

We claim that x has has no t, α-simple $p(\alpha+t+n), \alpha+2\log t+2\log n, 2^{-n+\alpha}$-acceptable statistical hypotheses. For the sake of contradiction assume that μ_p^t is such a hypothesis for x. By construction we have $\mu_p^t(x) \leqslant 2^{-n+\alpha}$ and hence the singleton set $T = \{x\}$ has small probability. It can be recognized in time $p(\alpha+t+n)$ by a program of length less than $\alpha + 2\log t + 2\log n$, consisting of t and n in the self-delimiting encoding followed by p. □

Remark 5. Assume that, in the definitions of a simple and acceptable hypothesis, we would restrict space instead of time. Then in Theorems 22 and 23 we would not need any assumptions.

6 The universal machine

Fix a deterministic one-tape Turing machine U that inputs three binary strings p, x, y and outputs one binary string and satisfies the following condition:

> For any other deterministic one-tape Turing machine V there is a constant c and a polynomial f such that for all p there is q with $|q| < |p| + c$ for which $U(q, y, r) = V(p, y, r)$ (for all y, r) and the running time of $U(q, y, r)$ is bounded by
>
> $$f(|y| + |r| + \text{ the running time of } V(p, y, r))$$
>
> and the similar inequality for the space holds as well.

This machine will be called *universal*. Using the universal machine we can define the Kolmogorov complexity (with or without time or space bounds) and the notions of programs and their running times for deterministic and randomized machines.

Kolmogorov complexity: $\mathrm{C}^t(x|y)$ is the minimal length of p such that $U(p, y, \Lambda) = x$ in time t. Similarly, $\mathrm{CS}^m(x|y)$ is the minimal length of p such that $U(p, y, \Lambda) = x$ on space s. If $U(p, y, \Lambda) = x$ in time t, we say that p is a *program for x conditional to y* (or simply a *program for x*, if $y = \Lambda$), and we call t *the running time of p* on input y.

The distinguishing complexity: $\mathrm{CD}^t(x|y)$ is the minimal length of p such that

- $U(p, x, y) = 1$;
- $U(p, x', y)$ halts in t steps for all x' of the same length as x;
- $U(p, x', y) = 0$ for all $x' \neq x$.

Programs of deterministic machines: we say that a *program p outputs y on input x in time t* if $U(p, y, \Lambda)$ in time t.

Programs of randomized machines: Considering the uniform probability distribution over r's, we obtain a universal randomized machine. More specifically, a *program of a randomized machine* is a pair (p, m). A machine with program (p, m) on an input string y tosses a fair coin m times and then outputs $U(p, y, r)$, where r denotes the outcome of the tossing. We can fix $y = \Lambda$ thus obtaining the notion of a program of a randomized machine without input. The length of the program (p, m) is defined as $|p| + \log_2 m$, and the running time (space) as the maximum over all $r \in \{0, 1\}^m$ of the running time (space) of $U(p, y, r)$.

References

[1] Antunes, L.; and Fortnow, L., Worst-Case Running Times for Average-Case Algorithms. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 298-303. IEEE, 2009.

[2] Buhrman, H.; Fortnow, L.; and Laplante, S., Resource-Bounded Kolmogorov Complexity Revisited. *SIAM Journal on Computing*, 31(3): 887-905. 2002.

[3] Fortnow, L., Kummer M., On resource-bounded instance complexity, *Theoretical Computer Science*, Volume **161**, Issues 1–2, 15 July 1996, Pages 123-140

[4] Gács P., Tromp J., Vitányi P.M.B., Algorithmic statistics, *IEEE Transactions on Information Theory*, v. 47, no. 6, 2001, p. 2443–2463

[5] R. Impagliazzo, G. Tardos. Decision versus search problems in superpolynomial time. 30th Annual Symposium on Foundations of Computer Science, 1989, pp. 222–227.

[6] A.N. Kolmogorov, The complexity of algorithms and the objective definition of randomness. Summary of the talk presented April 16, 1974 at Moscow Mathematical Society. *Uspekhi matematicheskikh nauk*, Russia, **29**(4[178]), 155 (1974).

[7] M. Li, P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, 3rd ed., Springer, New York, 2008.

[8] A. Shen, V. Uspensky, N. Vereshchagin *Kolmogorov complexity and algorithmic randomness*. MCCME, 2013 (Russian). English translation: http://www.lirmm.fr/~ashen/kolmbook-eng.pdf

[9] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th ACM Symposium on the Theory of Computing*, pages 330-335, 1983.

[10] Shen A., The concept of (α, β)-stochasticity in the Kolmogorov sense, and its properties. *Soviet Math. Dokl.*, v. 28, no. 1, 1983, p. 295–299

[11] Valiant, L., Vazirani, V., NP is as easy as detecting unique solutions. Theoretical Computer Science. 47 (1986) 85–93.

[12] Nikolay K. Vereshchagin, Alexander Shen, Algorithmic Statistics: Forty Years Later. *Computability and Complexity* 2017: 669-737

[13] Vereshchagin N. K, and Vitányi P. M. B, Kolmogorov's structure functions with an application to the foundations of model selection, *IEEE Transactions on Information Theory*, v. 50 (2004), no. 12, p. 3265–3290.

[14] Vinodchandran, N.V. and Zimand, M., On Optimal Language Compression for Sets in PSPACE/poly, *Theory of Computer Systems* (2015) **56**: 581.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il