RaCAF report

RaCAF group; edited by A. Shen, responsable scientifique

1 Introduction

The word "random" can be understood in many different ways. When we say that something is "random", we could mean a lot of different things. Restricting ourselves to science, we see two use cases: we can speak about *processes* or about *objects*. For example, a notion of random variable refers to processes: the outcome of a coin tossing is a random variable that has two equiprobable outcomes "head" and "tail", and it makes no sense to ask whether the outcome "heads" is random or not. On the other hand, a table of random numbers [28] printed in 1955 as a book and now available on the Internet, is a specific sequence of 1 000 000 digits. The authors make a claim that this sequence is an outcome of some random process; however, the users get a specific object (sequence) whose usefulness should be based on some properties of this object itself, not on the claims about its origin. These two meainings of the word "random" are connected: When we test a statistical hypothesis (e.g., "null hypothesis") about some random process and want to decide whether it is consistent with observations, this decision is made looking at an individual object (the experimental data). If we restrict ourselves to the second meaning (random objects), there still is a wide gap that ranges from purely theoretical notions like incompressibility (in the sense of Kolmogorov complexity, see [32, 34]) to quite practical issues related to statistical evidence, validity of research papers, and pseudorandomness (see below). The goal of our project is to approach this wide gap from different sides. In the following sections we try to describe the context for some project work/papers and then the work itself.

2 Statistical tests and p-values

As we have said, the connection between randomness of a process and randomness of an object appears in different contexts. In statistics it is called *hypotheses testing*. We assume some distribution (ofter called a *null hypothesis*) on the experimental results, and observe some specific outcome of the experiment (a value of a random

variable that is a model of our process according to the hypothesis). We need to decide whether this value is consistent with the hypothesis.

Looking at the example we considered (the hypothesis of a fair coin that assigns probabilities 1/2 to both outcomes "head" and "tail"), one may conclude that our setting makes no sense: none of the two outcomes can be considered as a confirmation (or refutation) of a hypothesis. However, in some other situations the question looks more reasonable. For example, if we observe 10000 coin tosses and see (say) 8121 heads, it makes the fair coin hypothesis looking bad. How is this possible? Recall that all 2^{10000} sequence of zeros and ones (heads and tails) are equiprobable according to the fair coin hypothesis, so the sequence we observed has the same probability as any other possible outcome.

This paradox plays a central role both in mathematical statistic (where it is a very practical question) and in algorithmic information theory (where it is considered from a more abstract viewpoint). Superficially, the approaches provided by these two fields, look quite different. Let us describe them shortly. In statistics, we consider some function defined on all possible outcomes, with numerical values. In our case the function may be the number of heads in the sequence. We reject (or at least question) the statistical hypothesis (model) if the value of the function is improbably high. More precisely, if this test function is f and the actual value (for the observed outcome) is c, we compute the probability of the event $f(X) \geq c$ according to the assumed distribution of the random variable X. It is called a "p-value"; in our example with 8121 heads the p-value is less than 10^{-913}. Following the advice of Borel who said " je suis arrivé à la conclusion qu'on ne devrait pas craindre d'employer le mot de *certitude* pour désigner une probabilité qui differe de l'unité d'une quantité suffisamment petite", we may reject the assumption of the fair coin.

Of course, usually the p-values are not so small as in this example; the values like 0.01 or 0.05 are often used to show that the observation is statistically significant (say, to convince the reader that a new medicine is really useful and that the improvement is not just a random fluctuation). There are many foundational problems related to this practice.

- *How to choose a test?* There are many possible functions that can be used as a test. For some of them the p-value is small, but not for others. By a suitable choice of a test function (tailored to the outcome) we can make the p-value small for every outcome. For example, consider the function f that is equal to 1 for the outcome we observed and equals 0 for all other inputs. Then the p-value will be the probability of this specific outcome (in our case, 2^{-10000}), and it this way almost any outcome can be used to reject almost any statistical hypothesis. To prevent this obvious cheating, we may require that the test function is fixed before the experiment is made. However, this is hard to guarantee: if you read a paper that reports some experimental results, how can you

be sure that the choice of a test function was made before the experiment? In other cases this requirement is not very practical: if somebody discovers some regularities in the existing experimental data (like Kepler did for planets' data), some tests are needed to check that it is not a random coincidence, but these tests cannot be fixed before the experiment, it is too late. (For planets one may wait for the new observations, but it is not always possible or practical.)

- *Publication selection.* A related (but different) problem appears when many researchers do a lot of experiments and use some p-value threshold to decide whether their observations were statistically significant. Assume that in fact null hypothesis is always true. Still, if 0.01 is used as significance threshold for the p-value, one would expect that 1% of all researchers will conclude that their observations are statistically significant and send their papers to a journal. So the journal receives a stream of false papers, and at the same time none of the authors did anything wrong and there is no reason to reject any of the papers.

These problems are quite practical. The American Statistican Association even prepared and released an official statement ("ASA Statement on Statistical Significance and P-Values", [2]) where the dangers of careless use of p-values are explained. The notion of p-value from the theoretical view point is analyzed in [18]. In [17] even a more dangerous practice is considered when p-values (or similar estimates) are used in court as evidence ("such and such combination of events cannot happen randomly"); the authors note that the arguments of this type are often used without clear methodology that consistently assigns probabilities for different arguments of that type, and without clear and justified thresholds for these probabilities (and people summoned to the jury service have no statistical training and still have to make their own opinion on the validity of evidence). One may also note a recent publication of 72 authors [9] that suggests to replace one rather arbitrary threshold for p-values (0.05) by another one (0.005).

3 The notion of randomness deficiency

Many questions discussed in the previous section are outside the scope of our project. Still they are motivations to analyze the notion of "test" as it is defined in algorithmic information theory, and the corresponding notion of randomness deficiency. Informally speaking, the idea of a "test fixed in advance" is replaced in algorithmic information theory by the idea of a "simple test", and the notion of simplicity is understood in terms of Kolmogorov complexity. See for details the Appendix 1 in [34]. The technical difference is that now we consider "calibrated" tests. There are two

notions of calibration that are close to each other but not equivalent. Consider a non-negative function t. The first calibration requirement, called "probability-bounded", says that the probability of the event $f(x) \geq c$ is at most $1/c$. This corresponds to p-values (the difference is that $1/f$ is used instead of f); this approach in a slightly different version was considered by Martin-Löf in his first paper about randomness (1966, see, e.g., [34] for the historical information). Another approach, developed by Levin and Gács in 1970s (see [34] for the references) is called "expectation-bounded"; it requires the expected value of f to be bounded by 1. The Markov inequality shows that it is a stronger requirement. The notions of tests were more or less forgotten after 1980; only recently they were studied again. In particular, RaCAF visitor G. Novikov [27] has shown that the difference between these two notions (and some intermediate notions) behaves in a rather subtle way (which is still not really understood). Similar effects (in a different settings) were observed in [3].

Another line of research related to tests and deficiencies can be called the "quantitative algorithmic randomness theory". The notion of randomness (for an infinite sequence of zeros and ones) was introduced by Martin-Löf in 1966 using tests. Unlike finite sequences where the sharp dividing line between random and non-random objects is impossible, for infinite sequences we can draw such a line and divide all sequences into random and non-random ones. A random sequence remains random if we change one bit. It is a natural property, but it implies that random sequence can start with a long sequence of zeros. This is reflected in the value of a function called *randomness deficiency*: it is finite for random sequences and infinite for non-random ones, and for a random sequence that starts with many zeros the randomness deficiency is large. Informally speaking, randomness deficiency is a negated logarithm of the p-value for the universal randomness test.

Therefore, the notion of randomness deficiency allows us to transform a "qualitative" statements in algorithmic randomness theory into "quantitative" ones. It is easier to do this for an expectation-bounded version of randomness deficiency. For example, a classical van Lambalgen theorem says that a pair (α, β) is random if and only if two conditions are true: (1) α is random; (2) β is random and remains random even with oracle α (see [34] for the exact definitions and statements). The quantitaive version of this result (see [8]) says that

$$d(\alpha, \beta) = d(\alpha) + d^{\alpha}(\beta),$$

where the left side is the deficiency of the pair (and is finite if and only if (α, β) is random), and in the right hand side we have the deficiencies of α and of β; in the latter case we use α as an oracle (which increases the deficiency). Left-hand side is finite if and only if *both* terms in the right hand side are finite. So we get the classical van Lambalgen's theorem plus some quantitative information for the case when the pair is random.

A similar quantitative result for image randomness is proven in [15]: it says that the randomness deficiency of a sequence equals the minimal randomness deficiency of its preimages.

4 Tests and random number generators

Here again we use statisticals tests but from different perspective. In natural science, we have some experimental data and choose a probabilistic model (distribution) that fits the data. Tests are used to check whether the model seems to be OK for the data; if not, we search for another model that fits better the existing data. Now, for the random number generators, the distribution is fixed (e.g., the fair coin distribution, the Bernoulli distribution with independent trials and equiprobable outcomes). We have some sequence (e.g., obtained by a physical coin tossing) and want to check whether it looks plausible (one could believe that the sequence is a result of the corresponding random process). For example, if the coin is not really symmetric, or the tossing is performed not high enough, or (in a more realistic settings) the parameters of the noise used to get random bits differ from the expected ones, such a test will tell us about the problems.

Again, we have the same problem. If a sequence is given, it is easy to construct a test that shows that the sequence is not random. On the other hand, if we fix some battery of tests in advance, then it is quite possible that some evidently non-random sequence passes all the tests, which is also undesirable. The algorithmic information theory provides a universal test where bad sequences are compressible ones. More precisely, a bit string x of length n has randomness deficiency at least d if $C(x) < n - d$. Then the fraction of strings that have deficiency greater than d is at most 2^{-d}, so we can construct a probability-bounded test based on this deficiency function. This test has many theoretical advantages; it is universal in the sense that a sequence that passes it at level d, will also pass any computable test at level $d + O(1)$, where $O(1)$-constant dependes on the choice of the other test. On the other hand, the complexity function (as defined by Kolmogorov) is non-computable, and the constant in $O(1)$ can be large, and these two problems make the complexity approach not practical. Still the standard compressors can be considered as randomness tests: from a practical viewpoint, if a bit sequence can be compressed by bzip (or gzip, or zip) by more than 50 bytes, it is definitely non-random (the corresponding p-value is about 2^{-400}, since 50 bytes = 400 bits).

To understand the real situation, one may look at the statistical tests that are really used to test random number generators (including physical ones). We have selected a classical battery of tests, called "Marsaglie Diehard tests", created in 1990s by George Marsaglia (1924–2011). Typical test from this collection works as follows. There is some random process that uses random bits. For example, one of the tests

fills a matrix of size 32×32. Then a rank of this matrix (a number between 0 and 32) is computed. This is repeated 40 000 times, and four numbers are computed: how many matrices have rank 32, 31, 30 and ≤ 29. (Marsaglia explains that matrices of rank less than 29 are rare, and he decided to combine them with matrices of rank 29.) These probabilities can be computed, so we can then apply, say, χ^2-criterion as a test (we have independent trials with finite number of outcomes and known probability).

Unfortunately, for many other tests in the collection one cannot compute the ditribution explicitly. In this cases Marsaglia uses some approximation for the distribution. Even if the approximation is proven, we lose the possibility for secondary testing (Section 5); moreover, in many cases there is no proof at all. One may suggest to use the "truly random generators" to certify the tests for which there is no proof. The problem, however, is that we get a vicious circle (to certify a generator we need to certify a test and vice versa) and, moreover, the nature and value of this certification is unclear.

As the linux man page for `diehard` puts it, "Lacking a source of perfect random numbers to use as a reference, validating the tests themselves is not easy and always leaves one with some ambiguity ... During development the best one can usually do is to rely heavily on these "presumed good" random number generators." We return to this question and formulate our suggestions in Section 6.

5 Secondary testing: independence needed

Secondary testing is sound if (1) we know the exact distribution of the test function; (2) the function has large min-entropy, i.e., each value has a negligible probability. Then, if we compute p-value as a function of this test, it will have distribution that is very close to the uniform distribution in $[0, 1]$. (If test function is injective, we get a uniform distribution on the set of all fractions with denominator N, where N is the number of possible outcomes.) Then we may use the test repeatedly (on fresh random bits) and get some sample from the uniform distribution (under the null hypothesis); some other tests (like Kolmogorov–Smirnov one) could be then used to check it.

One should mention that `diehard` test suite sometimes does not implement this approach correctly. The description of the "birthday spacing test" says: "...The first test uses bits 1–24 (counting from the left) from integers in the specified file. Then the file is closed and reopened. Next, bits 2–25 are used to provide birthdays, then 3–26 and so on to bits 9–32. Each set of bits provides a p-value, and the nine p-values provide a sample for a KSTEST [Kolmogorov–Smirnov test]". Here the *same* bits (in the overlapping range) are used in different tests, so there is no reason to expect that p-values will be uniformly distributed.

6 How to discredit a bad sequence in a reliable way

There is an approach (that we plan to test) to overcome the difficulties mentioned. Imagine that we have some test for which only some heuristic for approximating a p-value is conjectured. If we repeat this test using non-overlapping group of bits, we get some distribution. Imagine that this distribution is far from the uniform distribution in $[0, 1]$, so we suspect that the string we are testing is bad. How can we confirm this conclusion without any unproven assumptions?

First approach is mentioned above: compare the distribution of p-values with the similar distribution for "truly random" bits (say, using Kolmogorov–Smirnov test for checking whether two samples are drawn from the same distribution). If they differ significantly, this difference shows that our sequence is bad. In other words, we consider a "randomized randomness test", where the bits used for this "randomization" should be "truly random".

It seems at first (see the citation abowe) that this approach depends on the access to "truly random bits". Even if we have some physical generator that is believed to be good (passes other tests, for example), there is no way to prove it.

To overcome the difficulty we suggest the following trick. Assume that we have to test a sequence of bits and split it into N blocks of length M:

$$B = B_1 B_2 \ldots B_N$$

The we apply the (unproven) test to all the blocks and get some quasi-p-values $t(B_1), \ldots, t(B_N)$. If we had some truly random generator, we could generate random sequence

$$R = R_1 R_2 \ldots R_N$$

and then apply the Kolmorogov–Smirnov test to two samples $t(B_1), \ldots, t(B_N)$ and $t(R_1), \ldots, t(R_N)$. Now we want to do the same *without access to a certified source of random bits.* For that, we first use the generator that is believed to be good and get the sequence $R_1, \ldots, R_N$ as before. For testing we now need a twice longer sequence; we split it into $2N$ blocks of the same size:

$$B' = B_1 B_2 \ldots B_N B_{N+1} B_{N+2} \ldots B_{2N}.$$

Then we apply Kolmogorov–Smirnov test to two samples

$$t(B_1), t(B_2), \ldots, t(B_N) \qquad (*)$$

and

$$t(R_1 \oplus B_{N+1}), t(R_2 \oplus B_{N+2}), \ldots, t(R_N \oplus B_{2N}). \qquad (**)$$

What is the advantage of this approach? If the generator used to get R_i is really good, we are in the same situation as before with the same chances to catch the non-randomness of B' (though using twice as many bits). On the other hand, without

any assumptions on the sequence $R_1R_2 \ldots R_N$ we get a *provably valid* test: if indeed B' is generated with a correct distribution, both sequences $(*)$ and $(**)$ are drawn from the same distribution!

7 Testing physical randomness

We have acquired a device generating physical random bits, in which the correcting apparatus has been removed. Our study plan is twofold: fist to make the device go through a batch of statistical randomness test to understand the natural bias, second to measure its randomness through the use of algorithmic tests.

8 Randomness as a type of warranty

A notion of randomness appears in many contexts and could have different (but probably related) meanings in different contexts. We can use the notion of expectation-bounded test in the framework of game-theoretic approach to probability theory developed by Vovk and Shafer [39].

In the spirit of this theory, we consider the interaction between two players introduced in [31]: "randomness producer" (P) and "randomness consumer" (C). The producer provides a bit sequence of some fixed length N, in exchange for a 1 dollar payment. The consumer, not seeing this sequence, provides an expectation-bounded test, i.e., some non-negative function t defined on all N-bit strings with the average at most 1. (This producer does not know this function, so this is a game with incomplete information.) After the sequence R and test t are chosen, the producer pays back $t(R)$ dollars to the consumer.[1] In the Vovk–Shafer framework the randomness of the bits means that P is ready to participate in the game, i.e., randomness is a type of warranty for the bits that are shipped.

For example, if C plans to use the random bits bought from P for some randomized algorithm (that, say, generates a prime number with probability 0.999), then the function t would be equal to 1000 on sequences that lead to a composite numbers, and 0 elsewhere. The proof of the correctness of the algorithm guarantees that the average of this function is at most 1. So, bringing this function when buying the sequence, C hedges her risks: if she gets in trouble because the number turns out to be composite, at least she can get a \$1000 monetary compensation from P.

The framework suggested by Vovk and coauthors is related not only to randomness, but also to complexity. Recently he and Pavlovic suggested a new representation for randomness testing [38]. It turned out that the basic notion introduced by

[1] This corresponds to a "non-commercial" production of random bits; a commercial producer could charge a bit more, say, \$1.01 instead of 1 to cover its costs.

them (it can be called the *stopping time complexity*) can be obtained as a special case of conditional complexity with structured conditions introduced long ago (see [34] for the historical account). This connection and other properties of the stopping time complexity are studied in [4].

9 Weak Complexity Models

One of the obstacles for a practical use of Kolmogorov complexity (and the related notion of incompressibility) is its non-computability. This non-computability is inavoidable if we do not restrict the computational power of decompressors (description modes) used in the definition. Therefore, a restricted versions of Kolmogorov complexity could be interesting.

One extreme restriction is to consider finite memory and almost real-time computations, as provided by the finite automata computation models. It was known for a long time that (morally) finite-state incompressibility corresponds to Borel normality (all blocks of given lengths should have the same limit frequency in a random sequence). However, this result did not fit the general framework for complexity and compressibility as used in the other versions of Kolmogorov complexity. We show that this one can overcome this difficulty and define the class of description modes using finite-state memory and show that normality corresponds to the incompressibility in this class. As a byproduct we get simple proofs for many results about normal number (the equivalence between aligned and non-aligned definitions, Wall's theorem, Agafonov's theorem, Piatetski-Shapiro's theorem). See [33] for details.

10 Randomness extractors: one or multiple sources

From the practical viewpoint, the randomness extraction deals with the following problem. It is quite easy to implement some physical process that is believed to be "random" in the weak sense of unpredictability (white noise and radioactive decay are two well known examples of such a process). In other words, it is rather simple to make a physical device that produces a sequence of bits with high Shannon entropy (or min-entropy, or some other reasonable measure of "randomness"). However, it is virtually impossible to guarantee that the outcome of a real-world physical device is "perfect", i.e., provides a sequence of truly independent and uniformly distributed bits. An interesting challenge is therefore the problem of *extraction of randomness*, i.e., post-processing of random data that improves the "quality" of random bits. In practice this task is well known; for example, Marsaglia himself noted that some sequence obtained from a special device fails several random test and tried to correct

it by xor-ing with some good pseudorandom generator.[2]

A more formal question is what kind of transform should be applied to a sequence of bits with high enough measure of randomness to get a result that would be much closer (in some formal sense) to the uniform distribution. Such a transform may shrink the size of the data (the number of random output bits can be significantly less than the number of input bits), but improve the "quality" of randomness in these data.

Most known constructions of randomness extraction can be subdivided into two classes: extractors that use a short supplementary random input (which supposed to be almost perfect) and extractors that take two (or more) imperfect but mutually independent inputs.[3]

The methods of randomness extraction was substantially developed in theoretical computer science (mostly due to purely theoretical aspects of derandomization in computational complexity). In the last two decades the techniques of derandomization have made an impressive progress, the researchers working in this area have successfully employed very nontrivial mathematical and algorithmic techniques. The known constructions of randomness extractors use different tools, including spectral graph theory (e.g., random walks on expander graphs), coding theory (e.g., error correcting codes with efficient list decoding), see [30, 36] for a survey. However, in most natural settings there remains a large gap between "theoretical" results (those which characterize the parameters of randomness extractors that can be achieved in theory, if the computational complexity is not an issue), and more "practical" constructions that can be implemented by polynomial-time algorithms.

Though the reunification of the theoretical results for randomness extractors and the practical efficient algorithms remains elusive, the existing methods may be powerful enough to make progress in various problems of computer science and information theory. We plan to apply the techniques of randomness extraction on the two following (quite opposite) offshoots of the theory of randomness. The questions of the first type are very practical: how successful/efficient are the known constructions of randomness extractors (first of all, based on the random walk on expander graphs) when the raw data is the physical noise (obtained, e.g., from a standard sound card or a more specialized gadget). We plan to experiment with the following scheme: subsequence blocks in a weakly random sequences are used for a "weakly random walk" in a graph with expander-like properties. Another planned experiment is to

[2]Later a possible reason for the failure was found: CR-LF/LF (DOS/Unix) conversion that happens while copying files in text mode could be applied to the data.

[3]It is easy to see that one cannot hope to find a function that maps a "somehow random" long bit string x into one "almost perfect" bit b (not using any auxiliary information). Indeed, such a function should have preimages of 0 and 1 of approximately the same size, and a random variable uniformly distributed in one of them will have large entropy, min-entropy etc., and still will be mapped into a constant random variable.

try to use time-separated data for extraction from independent sources: it is much more plausible that some physical device has no memory (knowing today's noise hardly changes the distribution of tomorrow's noise for a memory-less device) than any assumption on the specific distribution.

The questions of the second type are rather theoretical and appear in algorithmic information theory (theory of Kolmogorov complexity): which information quantities can be extracted from a tuple of correlated strings? In a series of recent works it was shown (see the surveys [35, 40]) that the randomness extractors combined with standard hashing techniques help to "materialize" some quantities known in information theory. For instance, it turns to be possible to extract from a string x an almost shortest description of x conditional on another string y, which is a sort of operational interpretation of the conditional Kolmogorov complexity $C(x|y)$. Some other results of this type was obtained earlier by the participants of the project [29]. We suppose to use similar methods to find an operational characterization of more elusive information quantities like the mutual information $I(x:y) = C(x) + C(y) - C(x,y)$. Csiszár and Narayan suggested in [16] that such a characterization can be found in cryptographical settings. Assume for example that two remote parties (Alice and Bob) hold correlated data (Alice is given a string x, Bob is given another string y, and they both know an approximate value of $I(x:y)$). Can Alice and Bob use the given data to agree on a common secret key by communicating over an open channel? It seems that this problem can be solved using randomness extractors and some nonconventional information inequalities. Csiszár and Narayan have shown this in the framework of Shannon's information theory, and we are working on a similar result for the algorithmic framework.

11 Local rules enforcing high complexity

The following result deals more with the theoretical side of randomness notions. It shows, roughly speaking, that local rules are enough to force a uniquely defined and complex global structure. (One often considers local rules with complex global structure as a theoretical models for quasicrystals.) A technique based on computer science and logic tools (Kleene recursion theorem) was developed earlier by the participants of the project [14]. It was used to show that local rules can enforce a maximally complex structure (in a subspace of codimension 1): for a tiling of a plane one can construct a finite set of local rules (subshift of finite type in the terminology of the ergodic theory) that guarantees that one-dimensional sections of any tiling are "almost random", have complexity close to maximal ($\Omega(n)$ for n-bit substring).

In 2017 a new step in this direction was made. In the general case, a set of local rules can admit completely different tilings, with completely different and *a priori* non-uniform structural properties. This freedom is restricted if we consider *minimal*

or *quasiperiodic subshifts of finite type*. The minimality requirement says that every two tilings have one and the same set of finite patterns: if some finite pattern appears in one tiling, then it must appear (in fact, infinitely often) in every other one. The quasiperiodicity means that every valid structure is uniform in some sense: every finite pattern that appear in the structure at least once, must appear there infinitely often, with about the same density everywhere. In [13] we described the structures that can be obtained as "cross sections" (restrictions to a subspace of codimension 1) of minimal subshifts of finite type; we also constructed quasiperiodic subshifts of finite type with maximally complex structure in all subspace of codimension 1.

12 Algorithmic statistics

A systematic research of the relation between statistical hypotheses (models) and individual objects is a topic of *algorithmic statistics*. It appeared in the first papers (and talks) by Kolmogorov on algorithmic information theory; later many people (Rissanen, Bennett, Koppel, Antunes and others) suggested similar ideas in different (and not always clear) form. It turned out that all these approaches (Kolmogorov structure functions, (α, β)-stochasticity, minimal description length principle, logical depth, computational depth, sophistication) are essentially equivalent (see [23, 24] and [37] for a survey). One could say that "computability theory" part of algorithmic statistics is well understood now.

One can describe the main goal of algorithmic statistics as follows. We observe some data x (say, binary string) and have some statistical model for x (a finite distribution on binary strings, or, for simplicity, a finite set A with assumed uniform distribution on A). When A is a good model for x ($=x$ is a "typical", or "random" element of A)? The model A should be simple (have small complexity), and x should have small randomness deficiency (see above) in A. Vitanyi and Vereshchagin noted that these requirements are closely related to good two-part descriptions (when x is specified by A and the ordinal number of x in A). These notions are also closely related to resource-bounded complexity. However, the resource bounds used here are huge (busy beaver numbers, see [3]), so this remains more a computability theory that complexity theory even if we speak about bounded time (or space) computations.

To make these results (a bit) closer to practice, one should consider much more reasonable bounds, and some new techniques, like pseudo-random generator of Nisan–Wigderson, are needed. At that complexity level one should distinguish between time and space bounds (for busy beaver type bounds this difference does not matter). The space-bounded version of algorithmic statistics is considered in [25]. In [26] a time-bounded version is considered. The authors suggest three definitions of a plausible statistical hypothesis with polynomial time bounds, which are

called acceptability, plausibility and optimality. Roughly speaking, a probability distribution m 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 m). 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 m is called an optimal explanation for x if $m(x)$ is large. It turns out that (under some complexity-theoretic assumptions) for acceptability and plausibility there are infinitely many non-stochastic objects, i.e. objects that do not have simple plausible (acceptable) explanations, and 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).

13 Randomness and Continuous Time Processes

The previous discussion was mostly considering randomness as being generated by digital algorithms or machines. However, practical generation of randomness often relies on continuous (in time and space) processes. This includes the most emblematic example of randomness generation: rolling a dice. While the dynamic of a dice is governed by a purely deterministic law of evolution, its output is generally considered as random.

From a mathematical point of view, its evolution is described by some polynomial ordinary differential equation $y' = p(y)$ starting from some initial data $y(0) = y_0$. Mathematical theories and results have been invented to model this phenomenon and explain how a purely deterministic process can generate randomness.

They include *ergodic theory* which studies dynamical systems with an invariant measure, initially motivated by problems of statistical physics. One of its main concerns is the behavior of dynamical systems (and the corresponding measure-preserving transformations) when they are allowed to run for a long time. From this viewpoint, the randomness of a dice is explained by the incertitude present in the initial condition y_0, modeled as random, and its dynamics is seen as a process preserving this initial randomness. Computable versions of several theorems of ergodic theory have been proven.

However, another approach, though probably less common in 2017 (but not at the time of first computers that were analog machines), is however very natural: a dice can be considered as a process computing a value in $\{1, 2, \dots, 6\}$. More generally, every system modeled by an ordinary differential equation can be considered

as doing a computation in a similar sense. If we want to simulate a given continuous system, we could build an analog computer, that is, a mechanical or on electronic process, whose evolution corresponds to the system to be modeled. The term 'analog' in "analog computation" comes historically from the idea of computing by analogy, though in the recent years it is mostly understood as being the antonimous of 'digital': indeed, these simulation machines were working with continuous quantities such as angles or voltage (not bits as modern computers do).

Recent results showed that polynomial ordinary differential equations have many similarities with Turing machines. In particular, they can simulate Turing machines (and vice versa). The class of solutions of polynomial ordinary differential equations have many closure properties rather similar to the properties of computable functions. A recent breakthrough, obtained by the PhD thesis of Amaury Pouly was to establish that time is corresponding, in some deep sense, to the length of solutions. The results of this type open a way to define computability or complexity notions related to continuous functions internally, without any reference to digital concepts such as Turing machines (using only basic concepts like length, distance, etc.). The PhD thesis of Amaury Pouly, co-supervised by Olivier Bournez and Daniel Graça, has been awarded in August 2017 the selective Ackermann PhD Award. These statements have also been accepted to the selective *Journal of the ACM* [5].

Coming back to our example: One sees that it is possible to generate a random number $\{1, 2, \dots, 6\}$ using a polynomial ordinary differential equation. Namely, consider the one corresponding to the dice dynamics. Can one generate any function? We answered positively to this question through the result published in ICALP'17 [6], solving an open question from Rubel in 81. This has been remarked and a post in the well-known blog *Gödel's Lost Letter and P=NP* have been devoted to our result [19].

To be more concrete and precise: An astonishing fact was established by Lee A. Rubel (1981): there exists a fixed non-trivial fourth-order polynomial differential algebraic equation (DAE) such that for any positive continuous function φ on the reals, and for any positive continuous function $\epsilon(t)$, it has a $\mathcal{C}^\infty$ solution with $|y(t) - \varphi(t)| < \epsilon(t)$ for all t. Lee A. Rubel provided an explicit example of such a polynomial DAE. Other examples of universal DAE have later been proposed by other authors. However, while these results may seem very surprising, their proofs are quite simple and are frustrating, and can be interpreted more as the fact that (fourth-order) polynomial algebraic differential equations are too loose as a model compared to classical ordinary differential equations.

The question whether one can require the solution that approximates φ to be the unique solution for a given initial condition is a well known open problem [Rubel 1981, page 2], [Boshernitzan 1986, Conjecture 6.2]. In [6], we solve it and show that Rubel's statement holds for polynomial ordinary differential equations (ODEs), and

since polynomial ODEs have a unique solution given an initial data, this positively answers Rubel's open problem. More precisely, we show that there exists a *fixed* polynomial ODE such that for any φ and $\epsilon(t)$ there exists some initial condition that yields a solution that is ϵ-close to φ at all times.

These results were established in the purpose of trying to define concepts similar to Kolmogorov complexity for continuous functions. Of course, these concepts can be defined by transferring the questions back to the digital world, but we believe they can possibly be defined in an implicit/internal way, without any reference to concepts such as Turing machine. For example, by replacing the concept of "simplest program" by simplest ordinary differential equation/initial condition, one may expect theorems similar to the one known in the classical settings. The most fundamentals definitions and statements in the classical settings relies indeed on the fact that there exists universal Turing machines. Results established in [6] are clearly a first substantial step towards this, as they lead a way to talk about universal (polynomial) ordinary differential equations.

14 Higher randomness and computations

To understand the philosophical notion of randomness, one may study more powerful computational models. In particular, in order to understand the power provided by a random set, we have investigated computations to which as been added a generic ultrafilter [11]. This is not an easy task for any computational model, it comes more naturally with cellular automata. We add a generic ultrafilter to this model by constructing a shift-invariant interval-unbounded ultrafilter on $\mathbb{Z}$. The construction is interesting in itself, as different hypothesis are needed depending on the ground model considered. For example, starting from a model satisfying the continuum hypothesis, one can construct such an ultrafilter, whereas we need to use Martin's axiom in a model with larger continuum. We show that the power of these augmented cellular automata is strictly more powerful: such an enhanced automaton can use the ultrafilter to decide the totality of a given Turing Machine. All the ultrafilter constructed in this manner are essentially generic as they have been built via a forcing construction.

We have also studied the power of another enhanced model, the Infinite Time Turing Machines [10]. These machines behave like classical Turing Machines, but have special features to handle infinite ordinal time. The running times of these machines present gaps, and we proved that their gap distribution is both rich and complex. Some ordinals play an important role in this study : they are strongly closed ordinals called admissibles. Of those admissibles, some begin gaps and others are properly contained in a gap. We have given a comprehensive description of this later case.

In the study of these Infinite Time Turing Machines, we have embarked on the quest for rich specific infinite time behaviours of Turing Machines. We have defined infinite time variants of Tibor Rado's busy beaver fonctions and given a comprehensive study of the first machine classes, in addition to providing a theoretical outlook on the algorithmic complexity of these busy beaver functions [12]

References

[1] Chronological report about RaCAF progress, including references and texts of RaCAF-related papers, http:www.lirmm.fr/~ashen/racaf.html

[2] R. Wasserstein, N. Lazar, Tha ASA's Statement on *P*-Values: Context, Process and Purpose. ASA Statement on Statistical Significance and *P*-Values, *The American Statistician*, **70**(2), 129–133.

[3] M. Andreev, Busy Beavers and Kolmogorov Complexity, *Pursuit of the Unversal, proceedings of the 12th CiE conference, 2016*, Springer, Lecture Notes in Computer Science, 9709, p. 195–204.

[4] M. Andreev, G. Posobin, A. Shen, Plain stopping time and conditional complexities revisited, preprint, https://arxiv.org/abs/1708.08100

[5] O. Bournez, D.S. Gracça, A. Pouly, Polynomial Time corresponds to Solutions of Polynomial Ordinary Differential Equations of Polynomial Length, *Journal of the ACM*, accepted for publication.

[6] O. Bournez, A. Pouly, A Universal Ordinary Differential Equation, *International Colloquium on Automata, Language and Programming*, ICALP'2017, 116:1–116:14

[7] R. Brown, D. Eddelbuettel, D. Bauer, *Dieharder: A Random Number Test Suite. Version 3.31.1.* http://webhome.phy.duke.edu/~rgb/General/dieharder.php Linux man page: https://linux.die.net/man/1/dieharder

[8] B. Bauwens, A. Shen, H. Takahashi, Conditional Probabilities and van Lambalgen's Theorem Revisited, *Theory of Computing Systems*, 2017, doi:10.1007/s00224-017-9789-2

[9] D. Benjamin et al. (72 authors in total), Redefine Statistical Significance, *Nature Human Behaviour*, 2017, https://www.nature.com/articles/

[3]Papers that involve the RaCAF-related work of the participants are shown in blue in this list.

s41562-017-0189-z.epdf. See also: doi:10.17605/OSF.IO/MKY9J, https://psyarxiv.com/mky9j/ [Absract: We propose to change the default *P*-value threshold from 0.05 to 0.005 for statistical significance for claims of new discoveries from 0.05 to 0.005.]

[10] M. Carl, B. Durand, G. Lafitte, S. Ouazzani, Admissible in Gaps, *CiE 2017: Unveiling Dynamics and Complexity, Proceedings*, Lecture Notes in Computer Science, 10307, Springer, 2017, 175–186. https://doi.org/10.1007/978-3-319-58741-7_18

[11] J. Cervelle, G. Lafitte, On shift-invariant maximal filters and hormonal cellular automata, *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, 1–10, .https://doi.org/10.1109/LICS.2017.8005145

[12] O. Defrain, B. Durand, G. Lafitte, Infinite Time Busy Beavers, *CiE 2017: Unveiling Dynamics and Complexity, Proceedings*, Lecture Notes in Computer Science, 10307, Springer, 2017, 221–233. https://doi.org/10.1007/978-3-319-58741-7_22

[13] B. Durand, A. Romashchenko, On the Expressive Power of Quasi-Periodic SFT, *Mathematical Foundations of Computer Science*, 2017, to be published in the Proceedings

[14] B. Durand, A. Romashchenko, A. Shen, Fixed-point tile sets and their applications, *Journal of Computer and System Sciences*, **78**(3), 731–764, 2012.

[15] L. Bienvenu, M. Hoyrup, A. Shen, Layerwise Computability and Image Randomness, *Theory of Computing Systems*, 2017, doi:10.1007/s00224-017-9791-8

[16] I. Csiszár, P. Narayan, Secrecy capacities for multiple terminals. *IEEE Transactions on Information Theory*, **50**(12), 3047–3061 (2004).

[17] Impugning Randomness, Convincingly, *Studia Logica: An International Journal for Symbolic Logic*, Springer, **100**(1–2), February–April, 2012, 193-222, https://www.jstor.org/stable/41475223.

[18] Yu. Gurevich, V. Vovk, *Test statistics and p-values*, https://arxiv.org/abs/1702.02590

[19] R.J. Lipton, K.W. Regan, *Modeling Reality*, blog post, https://rjlipton.wordpress.com/2017/08/09/modeling-reality/

[20] Guilhem Marion, *Le hasard et sa production*, report de stage, LIRMM, see [1].

[21] G. Marsaglia, Diehard CDROM. Was available at http://stat.fsu.edu/pub/diehard/ as wikipedia page says; now (June 6, 2017) both the original link and the link to copy given there https://wayback.archive.org/web/20160120104710/http://www.stat.fsu.edu/pub/diehard/ do not work, but these files are still available in the internet, e.g., http://ftpmirror.your.org/pub/misc/diehard/cdrom/, and it seemed consistent with some other snapshots at wayback.archive.org

[22] G. Marsaglia, W. Tsang, Some difficult to pass test of randomness, *Journal of Statistical Software*, **7**(3), doi:10.18637/jss.v007.i03 (2002), https://www.jstatsoft.org/article/view/v007i03.

[23] A. Milovanov, Algorithmic Statistics: Normal Objects and Universal Models, *Computer Science in Russia 2016*Б Lecture Notes in Computer Science, v. 9691 (2016), 280–293.

[24] A. Milovanov, Some Properties of Antistochastic Strings, *Theory of Computing Systems*, published online 21 June 2016, DOI 10.1007/s00224-016-9695-z.

[25] A. Milovanov, On Algorithmic Statistics for space-bounded algorithms. In *Proceedings of 12th International Computer Science Symposium in Russia* (CSR 2017) LNCS, vol. **10304**, pp. 232–234, 2017.

[26] A. Milovanov, N. Vereshchagin, Stochasticity in Algorithmic Statistics for Polynomial Time, *32nd Computational Complexity Conference*(CCC 2017) proceedings (Leibniz International Proceedings in Informatics, LIPIcs), doi:10.4230/LIPIcs.CCC.2017.17, 17:1–17:18

[27] G. Novikov, Randomness Deficiences, *CiE 2017: Unveiling Dynamics and Complexity, Proceedings*, Lecture Notes in Computer Science, 10307, Springer, 2017, 338–350. https://link.springer.com/chapter/10.1007/978-3-319-58741-7_32

[28] A Million Random Digits with 100,000 Normal Deviates. The Free Press, 1955 (Reprinted by RAND, 2001). https://www.rand.org/pubs/monograph_reports/MR1418.html

[29] A. Romashchenko, *Coding in the fork network in the framework of Kolmogorov complexity*, preprint, arXiv:1602.02648.

[30] R. Shaltiel, An introduction to randomness extractors. In *International Colloquium on Automata, Languages, and Programming* , Springer, 2011, 21–41.

[31] A. Shen, Randomness as a type of warranty, *Varieties of Algorithmic Information conference talks*, Heidelberg, https://vai2015.sciencesconf.org/conference/vai2015/pages/talk.pdf

[32] A. Shen, Algorithmic Information Theory, book section in *The Routledge handbook of philosophy of information*, Routletge, 2016, 37–43.

[33] A. Shen, Automatic Kolmogorov complexity and normality revisited, *FCT 2017 Conference, Bordeaux, France, Proceedings*, full version: https://arxiv.org/pdf/1701.09060.pdf

[34] A. Shen, V. Uspensky, N. Vereshchagin, Kolmogorov Complexity and Algorithmic Randomness. A book accepted for publication (in 2017) by the American Mathematical Society. Draft: http://www.lirmm.fr/~ashen/kolmbook-eng.pdf

[35] J. Teutsch, M. Zimand, A brief on short descriptions, *ACM SIGACT News*, **47**(1), 42–67 (2016).

[36] S.P. Vadhan, Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, **7** (1–3), 1–336.

[37] N. Vereshchagin, A. Shen, *Algorithmic statistics: forty years later*. Book chapter in *Computability and Complexity. Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*. Lecture Notes in Computer Science, v. 10010, Springer, 2017, p. 669–737.

[38] V. Vovk, D. Pavlovic, Universal probability-free conformal prediction. In: Alex Gammerman, Zhiyuan Luo, Jesus Vega, and Vladimir Vovk, editors, *Proceedings of the Fifth International Symposium on Conformal and Probabilistic Prediction with Applications* (COPA 2016), v. 9653 of Lecture Notes in Artificial Intelligence, pages 40-47, Switzerland, 2016, Springer. See also: https://arxiv.org/pdf/1603.04283.pdf (March 2016)

[39] V. Vovk, G. Shafer, *Probability and Finance: It's Only a Game.* Wiley, 2001.

[40] M. Zimand, *Distributed compression through the lens of algorithmic information theory: a primer*, preprint, arXiv:1706.08468 (2017).