PREDICTABILITY AND COMPRESSIBILITY OF INFINITE BINARY STRINGS

TOMISLAV PETROVIĆ

PhD Thesis Proposal

Boston University
Computer Science Department
111 Cummington Mall, Boston, MA
tomislav@bu.edu

Keywords:
Algorithmic Information Theory, predictability, randomness, Martin-Löf randomness, Kolmogorov-Loveland randomness, computable randomness, martingale

Introduction

The proposed research is in the field of algorithmic information theory, in particular studying the nature of randomness. This issue is important as different applications require very different qualities of randomness (or pseudo-randomness). There is a long history of this question, even the formal study goes back to at least von Mises [4]. A natural way of defining randomness is via a kind of a betting game, a martingale, first used by J. Ville [7]. Winning in a game against infinite string can be viewed as kind of predictability of a string. Martin-Löf randomness can be shown to be equivalent to a kind of incompressibility of a string. There are some important problems in this area, open for four or five decades. We propose to attack at least one of these, the question of the strength of the Kolmogorov-Loveland style of betting. We developed some methods that are promising and have already led to a new kind of result on a very natural style of betting, the sequence-set betting strategies. In contrast to Martin-Löf tests, there is no one universal strategy that wins against all ML-random strings, but we proved there is a pair of them that do. This result has raised substantial interest among the experts.

Predictability is formally defined in terms of betting games and computable strategies. A string is said to be predictable iff there is a computable strategy that, starting with unit capital, by successive betting, wins an unbounded amount of capital. Compressible strings are non-Martin-Löf random strings. A compressible string has a sequence of initial segments (prefixes) s.t. the difference between the length of some program that outputs the prefix and the length of a prefix is

unbounded[1]. The length of the shortest program that outputs a prefix p is called Kolmogorov complexity[2] of the string, $Km(p)$.

The first betting game that we introduce is prefix-betting, a game where the player bets on a prefix, and if the string she's playing against starts with that prefix, she wins some money, if not, the wagered amount is lost. Since prefix-betting is a fair game, if the segment she bets on has length l, if she is right she wins 2^l times the wagered amount. It can be shown that a computable betting strategy can be constructed that predicts all compressible strings. Note however, that in the case of a correct guess, we learn something about the string we are playing against, namely the next segment of its prefix. But in case we were wrong, we learn very little, only that it doesn't begin with the prefix we had bet on. This asymmetry is twofold. Firstly, if the guess was correct, the measure of the set that we know contains the sequence reduces by a factor of 2^{-l}, and if the guess was incorrect it is reduced by a factor of only $(1-2^{-l})$. Secondly, if we make an infinite number of correct guesses, we learn all of the bits of the string. On the other hand, if we make an infinite sequence of wrong guesses, we still might not learn a single bit of the string.

Next, we introduce a different betting game, the sequence-set-betting. This is a game where the player, initially starting with the set of all strings and unit capital, partitions the set of strings into two sets of equal measure in a computable way, and bets on one of them. It can be shown that no such computable betting strategy can predict all compressible strings. On the other hand, we show that it is possible to construct two such strategies s.t. every compressible string is predicted by at least one of them. Unlike the prefix-betting, in a sequence-set-betting game, even if we have a succession of wrong guesses, the set of possible strings is halved with each bet and we know that the string we are playing against is a member of a small set. However, like for the prefix-betting, we can show that in this game too, in case we are winning, we learn much more about the bits of the string we are betting against, than in the case we are losing.

In [2] the non-monotonic-betting game is introduced. In this game, the player bets on the bits of the string, each bet consisting of the index (position) of the bit, wagered amount of capital and the value of the bit she is betting on. The strings for which there is no computable non-monotonic betting strategy that can predict them are called Kolmogorov-Loveland random. Whether Kolmogorov-Loveland random sequences are a proper subset of Martin-Löf random sequences is unknown, and is considered a major open problem in the field of algorithmic information theory [1, 12, 13, 14, 15, 16]. The sequence-set-betting is a generalization of non-monotonic-betting and there is no single non-monotonic strategy that predicts all compressible strings, but it is not known if two or more of non-monotonic strategies could predict all compressible strings. On the other hand, it was shown in [2] that for every positive unbounded computable function g there are two strategies that can predict all strings whose prefixes p of length $\ell(p)$ have $Km(p) \leq \ell(p) - g(\ell(p))$. In other words, the compressible strings that are not predicted by those strategies have such prefixes that the difference between their length and the length of the shortest

[1]On the monotone universal machine. An input for a monotone machine is an infinite binary string and it reads it sequentially, bit by bit. The output is a (possibly infinite) binary string, written sequentially.

[2]We use the monotone variant of Kolmogorov complexity. For a thorough exposition on Kolmogorov complexity and randomness see [1]

programs outputing them increases uncomputably slowly, i.e. the non-monotonic strategies predict all "highly" compressible strings.

[3] introduced the monotonic betting game. In this game, the player bets on the value of the next bit. Contrary to the non-monotonic game, for monotonic betting, there exist "highly" compressible unpredictable strings. It was shown in [2] that for every positive unbounded nondecreasing computable function g there is a string whose prefixes have $Km(p) \leq g(\ell(p))log(\ell(p))$ and no partial or total computable monotone strategy is able to predict it. In fact, for total computable monotonic strategies there is an unpredictable string whose prefixes satisfy $Km(p) \leq g(\ell(p))$. Interestingly, the sequences that can be predicted by partial computable non-monotonic betting strategies can also be predicted by total ones [12]. These results imply that computably random sequences are a proper subset of partial-computably random sequences which in turn are a proper subset of Kolmogorov-Loveland random sequences.

In contrast to both prefix and sequence-set betting, in both monotonic and non-monotonic betting, regardless of whether the prediction was correct, the player learns the same information about the string she is playing against (a single bit).

More generally, these betting games are a type of a broad class of "hat" games, which have been used to solve problems ranging from set theory to coding theory. However, the betting games haven't yet been studied in this context[3].

The thesis will present results on prefix-betting and sequence-set-betting games, advance on the open problem of whether Kolmogorov-Loveland random is the same as Martin-Löf random and study the betting games within the context of hat games.

Background

In 1919., Richard von Mises defined the random strings in terms of "collectives" [4]. An infinite binary string would be random (a collective in von Mises' terminology) if there is no "admissible" selection function which selects a subsequence with a frequency of ones different from $\frac{1}{2}$ in the limit. A selection function reads all of the bits of the sequence in succession, and each time after it reads a bit, decides whether to select the next unread bit. Von Mises left open the question of which functions should be admissible. If we allow for all functions to be admissible, then, for every collective, we have a function that selects a subsequence of only ones (or only zeros). In [5] A. Wald proposed to consider an arbitrary *countable* set of functions as admissible, in which case the collectives do exist. In [6] A. Church proposed that the set of admissible functions is chosen to be the set of computable selection functions. The strings for which every computable selection function selects a substring with the frequency of ones equal to $\frac{1}{2}$ in the limit are variously called Mises-Wald-Church random or computably stochastic.

In 1939., J. Ville [7] showed that there are Mises-Wald-Church random strings for which every finite initial segment has frequency of ones greater than $\frac{1}{2}$. This doesn't correspond with the intuition that a random string should look like a string obtained by tossing a fair coin. In that same paper he considers using martingales (betting strategies in the monotonic betting game) and shows that a set has measure 0 iff there is a betting strategy that wins unbounded capital when betting against a string from that set. Computable betting strategies were not studied until much

[3] to the knowledge of the author

later, by C.P. Schnorr [3]. The sequences which no computable monotonic betting strategy can predict are called computably random.

In general, the frequentist (or stochastic) approaches to defining randomness, like the one proposed by von Mises, have their stronger counterparts in the prediction by betting, like the one proposed by J. Ville. The difference is that the player in a betting game has additional freedom to express her certainty about the value she is betting on.

In 1960s Kolmogorov [8] and Loveland [9] independently proposed a relaxation of the von Mises' requirement that the selection function decides whether to select the *next* bit before reading it. The Kolmogorov-Loveland selection function can choose the position of the bit that is to be read, and before reading it chooses whether this bit will be selected. The sequences for which every computable selection function selects a subsequence with frequency of ones being $\frac{1}{2}$ in the limit are called Kolmogorov-Loveland stohastic.

P. Martin-Löf gives his definition of randomness in terms of computably enumerable statistical tests in [10]. His seminal paper marks a point of departure from the unpredictability paradigm towards the incompressibility paradigm in defining the random strings. To define incompressibility, we'll use the monotone variant of the Turing machine [1]. The reason for using monotone instead of a regular Turing machine is that it will simplify some of our expressions and it allows for both infinite inputs and outputs. The monotone machine is just like the regular machine, but it has an infinite input tape, an infinite output tape and an infinite work tape. The input tape is read-only, the output tape is write-only and the bits are read and written in succession, i.e. there are only two I/O operations, "read bit" and "write bit". The monotone Kolmogorov complexity of a finite string p, $Km(p)$, is defined as the length of the shortest prefix of some input string which the monotone machine reads before it outputs p. Denoting with $\ell(p)$ the length of p, we have that for any p, $Km(p) \leq \ell(p) + c$ (consider as input a program "print(p)"). An incompressible infinite string is s.t. for some constant c and all of its prefixes p, $Km(p) \geq \ell(p) - c$. It can be shown that Martin-Löf random strings are precisely the incompressible ones.

In [3] Schnorr gives his critique of Martin-Löf randomness for being too strong, and argues that randomness should be concerned with defeating computable strategies and not computably enumerable ones. In that paper Schnorr introduces what we'll call computable monotonic betting strategies.

In 1998., An. A. Muchnik describes a non-monotonic betting game [2] (in the paper it's just called the Game). The strings which are not predictable by non-monotonic betting strategies are called Kolmogorov-Loveland random.

In [17] a stronger type of martingales is considered, the martingale processes, and in [18] it is proven that there is a single computable martingale process that predicts all Martin-Löf random strings. The martingale processes are similar to the prefix-betting game, with the difference being that martingale processes can bet on any set of prefixes, whereas prefix-betting strategies bet on a single prefix. Since Martin-Löf random strings do have a characterization in terms of betting strategies, we can view all of the mentioned randomness notions in terms of betting games. One could argue that since we did find a total computable betting strategy that predicts all compressible strings, Schnorr's critique of Martin-Löf randomness

doesn't hold. However, there is a fundamental asymmetry built in the rules of the prefix-betting game.

To see this, one can imagine betting on strings as betting on the contents of enumerated boxes, each box containing one bit. In prefix-betting the player makes a statement about the content of some boxes and declares the degree to which she is certain that the statement is correct by wagering some portion of her capital. Then the player looks away while a judge (or a measuring device) opens those boxes, determines whether the player's guess was correct and then closes again the boxes. In case the player's guess was correct, the player learns a lot about the string she is betting against, namely its prefix. On the other hand, if the player's guess was wrong, the only thing she learns is that the string she is betting against *doesn't* start with the guessed prefix. Since the prefixes of compressible strings form a small set, even after a long sequence of unsuccessful bets she might learn very little about the string. In stark contrast, in the non-monotonic betting game, the player determines which box she wants to test and wagers some capital on the content of that box. Once she opens the box, it cannot be closed again. The testing of the content of the box is irreversible and regardless of the guessed content the player learns the same information from the test, i.e. whether the tested box contains 0 or 1.

Proposed Research

We'll introduce the prefix-betting game and show that there is a single total computable prefix-betting strategy C that predicts all compressible strings. In a prefix-betting game, a bet partitions a set of strings S into two sets, one of them containing only the infinite strings beginning with prefix p we bet on, and the other, a clopen set of infinite strings $S \backslash p\{0,1\}^\infty$. As the prefix we bet on is (possibly) long, a bet divides the set of strings into two sets of different Lebesgue measure, the small one containing the strings starting with the prefix and the large one containing the remaining strings. It is easy to see that the prefix-betting strategy in which every bet divides the set into two sets of equal measure is in fact a monotone-betting strategy, and as such cannot predict all compressible strings. We can also show that our strategy C makes bets in such a way that even if we make an infinite sequence of wrong guesses we learn that the string we are playing against is in a set that has measure greater than $\frac{1}{2}$. On the other hand, if we make an infinite number of correct guesses, not necessarily in sequence, we learn the whole string.

In order to remove this asymmetry in measure between the sets produced by the bet, we introduce a new betting game, the sequence-set-betting. In sequence-set-betting the bet partitions the set of strings into two clopen sets of equal measure and wagers some amount of capital on one of them. Now, after a sequence of l bets, we'll know that the string we are playing against is in a set of measure 2^{-l}, regardless of the correctness of our predictions. However, partitioning the set into halves comes at a cost - there cannot be a single computable betting strategy that wins on all compressible strings. It is easy to see this, at each bet choose the half on which the strategy loses and obtain a computable sequence of nested sets, each having a measure of $\frac{1}{2}$ the previous set. The strings in the intersection of these sets are compressible since we obtained them by a computable procedure, have measure 0, and, by construction, the strategy doesn't predict them. Similarly to non-monotonic-betting strategies we can show that the set of strings predicted by

partial computable sequence-set-betting strategies is the same as the set of strings predicted by the total ones. More importantly, we show how to construct two total computable sequence-set-betting strategies, A and B, s.t. every compressible string is predicted by at least one of them. It is perhaps interesting to note that the class of sequence-set-betting functions doesn't contain a single function that is equivalent in power to the universal one, but there are two of them which are, when combined. Even though the sequence-set-betting strategies are symmetric in the measure-theoretic sense, for our strategies A and B, we can show that there is still an asymmetry in information they learn about the string they bet against. In case of a correct guess they learn more about the bits of the string they are betting against then in the case of an incorrect guess. In fact, A and B can be constructed in such a way that in case of an infinite sequence of wrong guesses, they learn no bits of the string, and in case of an infinite number of correct guesses they learn all of the bits.

The key in our construction of sequence-set-betting strategies A and B is keeping them as independent as possible in the sense that the sets of strings on some losing path of betting outcomes for one strategy have a large intersection with the sets on as many losing paths of the other strategy as possible. The winning paths are determined by the opponent strategy C. When A and B wager their bets cooperatively they together can predict all of the compressible strings.

This suggests that the key to winning against non-monotonic strategies would be in forcing them to lose independence. While this is not possible to achieve when playing against sequence-set betting strategies, the paths of non-monotonic ones "exclude" each other whenever they pick the same position to bet on.

The other part of research will be studying the betting games in the context of hat (or voting) games, which to the knowledge of the author has not yet been done. The hat games are a broad family of games which have been used in areas ranging from set theory to coding theory [19, 20]. In particular, a somewhat famous [21] finite hat game related to the error correcting codes was used in [22] to show some results on autoreducibility of incompressible strings. Broadly speaking, a string is autoreducible if there is a way to deduce the unknown bits of the string from the known ones. This property is related to the non-monotonic game in the sense that non-monotonic strategies predict the strings which are in a way autoreducible. In order to win against non-monotonic strategies we would like to have sets of strings where the unknown bits cannot be deduced from the known ones. One way of achieving this in a finite setting uses partitioning strings of chosen length l into disjoint sets according to the remainder when the number of ones in a string is divided by a chosen number n. We have that if l is somewhat greater than n, these sets contain about the same number of words, and if you don't know several bits of a string, you don't know in which set the string is. Now suppose that you choose one of those sets, you would need to read most of the bits of a string in order to recognize that the string is from a chosen set. Extending this finite result to infinite binary strings would lead to forcing the non-monotonic betting strategies to have their bets coincide on many positions, thus losing their independence and providing a way to resolve the Kolmogorov-Loveland vs. Martin-Löf randomness question.

References

[1] Ming Li, Paul M. B. Vitányi: An Introduction to Kolmogorov Complexity and Its Applications, Third Edition. Texts in Computer Science, Springer 2008, ISBN 978-0-387-33998-6, pp. i-xxiii, 1-790

[2] Andrei A. Muchnik, Alexei L. Semenov, Vladimir A. Uspensky: Mathematical Metaphysics of Randomness. Theor. Comput. Sci. 207(2): 263-317 (1998)

[3] Claus-Peter Schnorr: A Unified Approach to the Definition of Random Sequences. Mathematical Systems Theory 5(3): 246-258 (1971)

[4] R. von Mises. Grundlagen der Wahrscheinlichkeitsrechnung. Mathematische Zeitschrift, 5:52–99, 1919. [xviii, 229]

[5] A. Wald. Die Wiederspruchsfreiheit des Kollektivbegriffes der Wahrscheinlichkeitsrechnung. Ergebnisse eines mathematischen Kolloquiums, 8:38–72, 1937.

[6] A. Church. On the concept of a random sequence. Bull. Amer. Math. Soc., 46:130–135, 1940.

[7] J. Ville. Etude Critique de la Notion de Collectif. Gauthier-Villars, Paris, 1939.

[8] A.N. Kolmogorov. On tables of random numbers. Sankhya , The Indian Journal of Statistics, Ser. A, 25:369–376, 1963.

[9] D.W. Loveland. A new interpretation of von Mises' concept of a random sequence. Z. Math. Logik und Grundlagen Math., 12:279–294, 1966.

[10] P. Martin-Löf. The definition of random sequences. Inform. Contr., 9:602–619, 1966.

[11] W. Merkle. The Kolmogorov-Loveland stochastic sequences are not closed under selecting subsequences. The Journal of Symbolic Logic, 68:1362–1376, 2003.

[12] Wolfgang Merkle, Joseph S. Miller, André Nies, Jan Reimann, Frank Stephan: Kolmogorov-Loveland randomness and stochasticity. Ann. Pure Appl. Logic 138(1-3): 183-210 (2006)

[13] Rodney G. Downey, Denis R. Hirschfeldt: Algorithmic Randomness and Complexity. Theory and Applications of Computability, Springer 2010, ISBN 978-0-387-95567-4, pp. 1-766

[14] Joseph S. Miller, André Nies: Randomness and Computability: Open Questions. Bulletin of Symbolic Logic 12(3): 390-410 (2006)

[15] K. Ambos-Spies and A. Kučera. Randomness in computability theory. In P. Cholak, S. Lempp, M. Lerman, and R.A. Shore, editors, Computability Theory and Its Applications: Current Trends and Open Problems, vol- ume 257 of Contemporary Mathematics, pages 1–14. American Math. Society, 2000.

[16] Rodney G. Downey, Denis R. Hirschfeldt, André Nies, Sebastiaan Terwijn: Calibrating Randomness. Bulletin of Symbolic Logic 12(3): 411-491 (2006)

[17] J.M. Hitchcock and J.H. Lutz. Why computational complexity requires stricter martingales. Theory Comput. Syst., 39(2):277–296, 2006.

[18] Wolfgang Merkle, Nenad Mihailovic, Theodore A. Slaman: Some Results on Effective Randomness. Theory Comput. Syst. 39(5): 707-721 (2006)

[19] Christopher S. Hardin, Alan D. Taylor: An introduction to infinite hat problems. The Mathematical Intelligencer 30(4): 20-25 (2008)

[20] James Aspnes, Richard Beigel, Merrick L. Furst, Steven Rudich: The Expressive Power of Voting Polynomials. Combinatorica 14(2): 135-148 (1994)

[21] S. Robinson, Why mathematicians now care about their hat color, New York Times, April 10, 2001.

[22] Todd Ebert, Wolfgang Merkle, Heribert Vollmer: On the Autoreducibility of Random Sequences. SIAM J. Comput. 32(6): 1542-1569 (2003)