The normalized algorithmic information distance can not be approximated

Bruno Bauwens[1] [0000−0002−6138−0591] and Ilya Blinnikov[1]

National Research University Higher School of Economics, 11, Pokrovsky Boulevard, 109028, Moscow. brbauwens@gmail.com

Abstract. It is known that the normalized algorithmic information distance is not computable and not semicomputable. We show that for all $\varepsilon < 1/2$, there exist no semicomputable functions that differ from N by at most ε. Moreover, for any computable function f such that $|\lim_t f(x,y,t) - \mathrm{N}(x,y)| \leq \varepsilon$ and for all n, there exist strings x, y of length n such that $\sum_t |f(x,y,t+1) - f(x,y,t)| \geq \Omega(\log n)$. This is optimal up to constant factors.

We also show that the maximal number of oscillations of a limit approximation of N is $\Omega(n/\log n)$. This strengthens the $\omega(1)$ lower bound from [K. Ambos-Spies, W. Merkle, and S.A. Terwijn, 2019, *Normalized information distance and the oscillation hierarchy*].

Keywords: Algorithmic information distance · Kolmogorov complexity · Computability theory · Oscillation hierarchy.

1 Introduction

The information distance defines a metric on bit strings that in some sense takes all "algorithmic regularities" into account. This distance was defined in [4] as $\mathrm{E}(x,y) = \max\{\mathrm{K}(x\,|\,y), \mathrm{K}(y\,|\,x)\}$, where $\mathrm{K}(\cdot|\cdot)$ denotes conditional prefix Kolmogorov complexity relative to some fixed optimal prefix-free Turing machine; we refer to appendix of the ArXiv version of this paper for the definition and basic properties, and to the books [6, 8] for more background. After minor modifications, this distance satisfies the axioms of a metric, as explained in the appendix of the ArXiv version. We refer to [2] for an overview of many equivalent characterizations.

The distance is not computable. However, conditional Kolmogorov complexity is *upper semicomputable*, which means that there exists a computable function $f\colon \{0,1\}^* \times \{0,1\}^* \times \mathbb{N} \to \mathbb{Q}$ for which $\mathrm{K}(x\,|\,y) = \lim_t f(x,y,t)$, and that is non-increasing in its last argument t. Hence, also E is upper semicomputable.

The distance E is useful to compare strings of similar complexity. However, for strings of different complexity, a normalized variant is often preferable.

Definition 1. *The* normalized *algorithmic information distance of strings x and y is*[1]

$$\mathrm{N}(x,y) \;=\; \frac{\max\{\mathrm{K}(x\,|\,y),\mathrm{K}(y\,|\,x)\}}{\max\{\mathrm{K}(x),\mathrm{K}(y)\}}.$$

This normalized distance has inspired many applications in machine learning, where complexities are heuristically estimated using popular practical compression algorithms such as gzip, bzip2 and PPMZ, see [6, section 8.4]. Within small additive terms, the function N has values in the real interval $[0,1]$ and satisfies the axioms of a metric:

- $0 \le \mathrm{N}(x,y) \le 1 + O(1/\mathrm{K}(x,y))$,
- $\mathrm{N}(x,y) = \mathrm{N}(y,x)$,
- $\mathrm{N}(x,x) \le O(1/\mathrm{K}(x))$,
- $\mathrm{N}(x,y) + \mathrm{N}(y,z) \ge \mathrm{N}(x,z) - O((\log \mathrm{K}(y))/\mathrm{K}(y))$.

See [6, Theorem 8.4.1].[2]

In this paper, we study the computability of N. Note that if Kolmogorov complexity were computable, then also N would be computable. But this is not the case, and in [9] it is proven that N is not upper semicomputable and not lower semicomputable, (i.e. $-\mathrm{N}$ is not upper semicomputable). Below in Lemmas 2 and 4 we present simple proofs. In fact, in [9] it is proven that (i) there exists no lower semicomputable function that differs from N by at most some constant $\varepsilon < 1/2$, and (ii) there exists no upper semicomputable function that differs at most $\varepsilon = (\log n)/n$ from N on n-bit strings. Theorem 1 below implies that (ii) is also true for all $\varepsilon < 1/2$.

By definition, N is the ratio of two upper semicomputable functions, and hence it is *limit computable*, which means that there exists a computable function f such that $\mathrm{N}(x,y) = \lim_t f(x,y,t)$. A function f that satisfies this property is called a *limit approximation* of N.

We define a *trivial limit approximation* f_{tr} of N where $f_{\mathrm{tr}}(x,y,t)$ is obtained by replacing all appearances of $\mathrm{K}(\cdot)$ and $\mathrm{K}(\cdot|\cdot)$ in Definition 1 by upper approximations $\mathrm{K}_t(\cdot)$ and $\mathrm{K}_t(\cdot|\cdot)$, where $(x,t) \mapsto \mathrm{K}_t(x)$ is a computable function satisfying $\lim \mathrm{K}_t(x) = \mathrm{K}(x)$ and $\mathrm{K}_1(x) \ge \mathrm{K}_2(x) \ge \ldots$; and similar for $\mathrm{K}_t(\cdot|\cdot)$. We assume that $\mathrm{K}_1(x\,|\,y)$ and $\mathrm{K}_1(x)$ are bounded by $O(n)$ for all x of length n.

[1] The numerator is nonzero, even if $x = y$. $\mathrm{K}_U(x) \ge 1$ holds for every choice of the optimal prefix-free Turing machine U, because such machines never halt on input the empty string. Indeed, if it halted, then it would be the only halting program by the prefix property, and hence, the machine can not be optimal.

[2] In [6, Exercise 8.4.3] it is claimed that for the prefix variant of the normalized information distance, one can improve the precision of the last item to $O(1/\mathrm{K}(x,y,z))$. However, we do not know a proof of this. If this were true, then with minor modifications of N similar to those in the appendix of the ArXiv version. all axioms of a metric can be satisfied precisely.

Lemma 1. *For all n and strings x, y of length at most n:*

$$\sum_{t=1}^{\infty} |f_{\rm tr}(x,y,t+1) - f_{\rm tr}(x,y,t)| \;\leq\; 2\ln n + O(1).$$

Definition 2. *An* ε-approximation *of a function g is a limit approximation of a function g' with $g - \varepsilon \leq g' \leq g + \varepsilon$.*

For a suitable choice of U, we have $0 \leq \mathrm{N} \leq 1$, and the function defined by $f(x,y,t) = 1/2$ is a $(1/2)$-approximation.[3] We show that for $\varepsilon < 1/2$ and every ε-approximation, the sum in the above lemma is at least logarithmic.

Theorem 1. *Let f be an ε-approximation of* N *with $\varepsilon < 1/2$. For large n:*

$$\max_{x,y\in\{0,1\}^n} \sum_{t=1}^{\infty} |f(x,y,t+1) - f(x,y,t)| \;\geq\; \tfrac{1}{100}\cdot(1-2\varepsilon)^2 \cdot \log n.$$

This result implies that for each $\varepsilon < 1/2$, there exists no upper semicomputable function that differs from N by at most ε.

We now state the main result of [1].

Definition 3. *Let $k \geq 1$. A sequence $a_1, a_2, \ldots$ of real numbers has* at most k oscillations *if the sequence can be written as a concatenation of k sequences ($k-1$ finite and 1 infinite) such that each sequence is either monotonically non-increasing or non-decreasing. The sequence has* 0 *oscillations if $a_1 = a_2 = \ldots$*

The main result of [1] states that no 0-approximation f of N has at most a constant number of oscillations. More precisely, for each k, there exists a pair (x,y) such that $f(x,y,1), f(x,y,2), \ldots$ does not have at most k oscillations.

Let $k\colon \mathbb{N} \to \mathbb{N}$. We say that f *has at least $k(n)$ oscillations*, if for all n there exists a pair (x,y) of strings of length at most n, such that $f(x,y,1), f(x,y,2), \ldots$ does not have at most $k(n)-1$ oscillations. (The proof of) Theorem 1 implies that if $\varepsilon < 1/2$, then any ε-approximation has at least $\Omega((1-2\varepsilon)^2 \log n)$ oscillations.

The trivial 0-approximation $f_{\rm tr}$ has at most $O(n)$ oscillations, because each upper-approximation of Kolmogorov complexity in its definition is bounded by $O(n)$ on n-bit strings, and hence, there can be at most this many updates. Can it be significantly less than n, for example at most $n/100$ for large n?

The answer is positive. For all constants c, there exist optimal machines U in the definition of complexity K for which the number of updates of K_t is at most $n/c + O(\log n)$. For example, one may select an optimal U whose halting programs all have length 0 modulo c. If N is defined relative to such a machine, than the total number of updates is $2n/c + O(\log n)$. Hence, for every constant e there exists a version of N and a 0-approximation that has at most n/e oscillations for large input sizes n. Our second main result provides an almost linear lower bound on the number of oscillations.

[3] For general optimal U, and for $\varepsilon > 1/2$, we can obtain an ε-approximation that is constant in t by choosing $f(x,y,t) = \mathrm{N}(x,y)$ for some finite set of pairs (x,y), and by choosing $f(x,y,t) = 1/2$ otherwise.

Theorem 2. *Every* 0*-approximation of* N *has at least* $\Omega(n/\log n)$ *oscillations.*

In an extended version of this article, we plan to improve the $\Omega(n/\log n)$ lower bound to an $\Omega(n)$ bound. This requires a more involved variant of our proof.

Theorems 1 and 2 both imply that N and hence Kolmogorov complexity is not computable. In fact, they imply something stronger: $\mathrm{K}(\mathrm{K}(x \mid y) \mid x, y)$ can not be bounded by a constant.[4] It has been shown that $\mathrm{K}(\mathrm{K}(x) \mid x) \geq \log n - O(1)$, see [5, 3], and our proofs are related. Like the proof in [3], we also use the game technique. This means that we present a game, present a winning strategy, and show that this implies the result. Using games one often obtains tight results with more intuitive proofs. (Moreover, the technique allows to easily involve students in research, because after the game is formulated, typically no specific background is needed to find a winning strategy.) For more examples of the game technique in computability theory and algorithmic information theory, we refer to [7].

N is not upper nor lower semicomputable

For the sake of completeness, we present short proofs of the results in [9], obtained from Theorem 3.4 and Proposition 3.6 from [1] (presented in a form that is easily accessible to people with little background in the field). A function g is *lower semicomputable* if $-g$ is upper semicomputable.

Lemma 2. N *is not lower semicomputable.*

Proof. Note that for large n, there exist n-bit x and y such that

$$\mathrm{N}(x, y) \;\geq\; 1/2\,.$$

Indeed, for any y, there exists an n-bit x such that $\mathrm{K}(x \mid y) \geq n$. The denominator of N is at most $n + O(\log n)$, and the inequality follows for large n.

Assume N was lower semicomputable. On input n, one could search for such a pair (x, y), and we denote the first such pair that appears by (x_n, y_n). We have $\mathrm{K}(x_n) = \mathrm{K}(n) + O(1)$ and $\max\{\mathrm{K}(x_n \mid y_n), \mathrm{K}(y_n \mid x_n)\} \leq O(1)$. Hence $\mathrm{N}(x_n, y_n) \leq O(1/\mathrm{K}(n))$. For large n this approaches 0, contradicting the equation above.

Remark. With the same argument, it follows that for any $\varepsilon < 1/2$, there exists no lower semicomputable function that differs from N by at most ε. Indeed, instead

[4] Indeed, if this were bounded by c, there would exist an upper approximation f of $\mathrm{K}(\cdot \mid \cdot)$ such that for each pair (x, y), the function $f(x, y, \cdot)$ has only finitely many values. (We modify any upper approximation of complexity by only outputting values k on input x, y, for which $\mathrm{K}(k \mid x, y) \leq c$. There are at most 2^c such k.) Hence, there would exist an approximation f' of N such that for all x, y, the function $f'(x, y, \cdot)$ has only finitely many values. Such functions would have only finitely many oscillations, contradicting Theorem 2, and a finite total update, contradicting Theorem 1.

of $\mathrm{N}(x,y) \geq 1/2$ we could as well use $\mathrm{N}(x,y) \geq 1/2 + \varepsilon$, and search for (x_n, y_n) for which the estimate is at least $1/2$.

To prove that N is not upper semicomputable, we use the following well-known lemma.

Lemma 3. *The complexity function* $\mathrm{K}(\cdot)$ *has no unbounded lower semicomputable lower bound.*

Proof. This is proven by the same argument as for the uncomputability of K, (see appendix the appendix of the ArXiv version): suppose such bound $B(x) \leq \mathrm{K}(x)$ exists. Then on input n, one can search for a string x_n with $n \leq B(x_n)$ and hence $n \leq \mathrm{K}(x_n)$. But since there exists an algorithm to compute x_n given n, we have $\mathrm{K}(x_n) \leq O(\log n)$. This is a contradiction for large n. Hence, no such B exists.

Lemma 4. N *is not upper semicomputable.*

Proof. By optimality of the prefix-free machine in the definition of K, we have that $\mathrm{K}(x \,|\, y) \geq 1$ for all x and y. Thus $1 \leq \mathrm{K}(x \,|\, x) \leq O(1)$, and hence,

$$1/\mathrm{K}(x) \;\leq\; \mathrm{N}(x,x) \;\leq\; O(1/\mathrm{K}(x)).$$

If N were upper semicomputable, we would obtain an unbounded lower semicomputable lower bound of K, which contradicts Lemma 3.

2 Trivial approximations have at most logarithmic total update

Lemma 1 follows from the following lemma for $c \leq O(1)$ and the upper bound $m \leq O(n)$ on the upper approximations of Kolmogorov complexity.

Lemma 5. *Assume* $1 \leq a_1 \leq a_2 \leq \cdots \leq a_m \leq m$, $1 \leq b_1 \leq b_2 \leq \cdots \leq b_m \leq m$ *and* $a_i \leq b_i + c$. *Then,*

$$\sum_{i \leq m} \left| \frac{a_i}{b_i} - \frac{a_{i+1}}{b_{i+1}} \right| \;\leq\; 2 \ln m + O(c^2).$$

Proof. We first assume $c = 0$. We prove a continuous variant. Let $\alpha, \beta \colon [0,m] \to [1,m]$ be non-decreasing real functions with $\alpha(t) \leq \beta(t)$ and $1 \leq \alpha(0) \leq \beta(m) \leq m$. The sum in the lemma can be seen as a special case of

$$\int_{t=0}^{t=m} \left| \mathrm{d} \frac{\alpha(t)}{\beta(t)} \right| \;=\; \int \frac{\mathrm{d}\alpha(t)}{\beta(t)} \;+\; \int \frac{\alpha(t)}{\beta^2(t)} \mathrm{d}\beta(t).$$

The left integral in the sum is maximized by setting $\beta(t)$ equal to its minimal possible value, which is $\alpha(t)$. The right one is maximized for the maximal value of $\alpha(t)$, which is $\beta(t)$. Thus,

$$\leq \int_{u=\alpha(0)}^{u=\alpha(m)} \frac{\mathrm{d}u}{u} \;+\; \int_{u=\beta(0)}^{u=\beta(m)} \frac{\mathrm{d}u}{u} \;\leq\; 2 \ln m.$$

For $c \geq 0$, the minimal value of β is $\max\{1, \alpha - c\}$ and the maximal value of α is $\min\{m, \beta + c\}$. The result follows after a calculation.

3 Oscillations of 0-approximations, the game

For technical reasons, we first consider the *plain length conditional* variant of the normalized information distance N'. For notational convenience, we restrict the definition to pairs of strings of equal length.

Definition 4. *For all n and strings x and y of length n, let*

$$\mathrm{N}'(x,y) \;=\; \frac{\max\{\mathrm{C}(x \,|\, y), \mathrm{C}(y \,|\, x)\}}{\max\{\mathrm{C}(x \,|\, n), \mathrm{C}(y \,|\, n)\}}.$$

If $\mathrm{C}(x \,|\, n) = 0$, let $\mathrm{N}'(x,x) = 0$.

Remarks.
- For $x \neq y$, the denominator is at least 1, since at most 1 string can have complexity zero relative to n.
- The choice of the value of $\mathrm{N}'(x,x)$ if $\mathrm{C}(x \,|\, n) = 0$ is arbitrary, and does not affect Proposition 1 below.
- In the numerator, the length n is already included in the condition, since it equals the length of the strings.
- There exists a trivial approximation of N' with at most $2n + O(1)$ oscillations. Indeed, consider an approximation obtained by defining $\mathrm{C}_t(\cdot|\cdot)$ with brute force searches among programs of length at most $n + O(1)$.
- Again, for every constant e, we can construct an optimal machine and a 0-approximation of N' for which the number of oscillations is at most n/e. We now present a matching lower bound.

Proposition 1. *Every 0-approximation of N' has at least $\Omega(n)$ oscillations.*

In this section, we show that the proposition is equivalent to the existence of a winning strategy for a player in a combinatorial (full information) game. In the last section of the paper, we present such a winning strategy.

Description of game $\mathcal{G}_{n,c,k}$. The game has 3 integer parameters: $n \geq 1$, $c \geq 1$ and $k \geq 0$. It is played on two 2-dimensional grids X and Z. Grid X has size $n \times 2^n$. Its rows are indexed by integers $\{0, 1, \ldots, n-1\}$, and its columns are indexed by n-bit strings. Let X_u be the column indexed by the string u. See figure 1 for an example with $n = 3$. Grid Z has size $n \times \binom{2^n+1}{2}$. The rows are indexed by integers $\{0, \ldots, n-1\}$, and its columns are indexed by unordered pairs $\{u, v\}$, where u and v are n-bit strings, (that may be equal).[5] We sometimes denote unordered pairs $\{u, v\}$ of n-bit strings as uv, and write $\mathrm{Z}_{\{u,v\}} = \mathrm{Z}_{uv}$. Note that $\mathrm{Z}_{uv} = \mathrm{Z}_{vu}$. Let $u \in \{0,1\}^n$. The *slice* Z_u of Z is the 2-dimensional grid of size $n \times 2^n$ containing all columns Z_{uv} with $v \in \{0,1\}^n$. Additionally, Bob

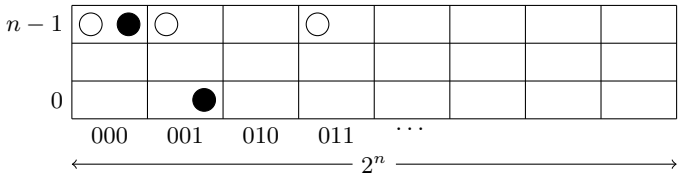

Fig. 1. Example of board X with $n = 3$. Alice has placed 2 tokens in row 2 (white), and Bob has placed 1 token in row 0 and 1 in row 2 (black). The row restrictions for both players are satisfied, since $\max\{1, 3\} \leq 2^2$ and $1 \leq 2^0$. $X_{000} = X_{011} = 2$, $X_{001} = 0$ and $X_{010} = 3$.

must generate a function f mapping unordered pairs of n-bit strings and natural numbers to real numbers.

Two players, Alice and Bob, alternate turns. The rounds are numbered as $t = 1, 2, \ldots$ At each round, Alice plays first. At her turn, she places tokens on cells of the grids. She must place at least 1 token. Afterwards, Bob places zero or more tokens on the grids, and he declares all values $f(uv, t)$ for all unordered pairs $\{u, v\}$, where t is the number of the current round. This terminates round t, and the players start with round $t + 1$.

For each player, for each $i \in \{1, \ldots, n\}$, and for all grids $\mathrm{G} \in \{\mathrm{X}\} \cup \{\mathrm{Z}_u : u \in \{0,1\}^n\}$, the following *row restriction* should be satisfied: *The total number of tokens that the player has placed during the whole game in the i-th row of* G*, is at most* 2^i. If a player does not satisfy this restriction, the game terminates and the other player *wins*. See figure 1. Bob's moves should satisfy 2 additional requirements. If after his turn these requirements are not satisfied, the game terminates and Alice wins.

– Let X_u be the value of column X_u given by the minimal row-index of a cell in X_u containing a token. If X_u contains no tokens, then $X_u = n$. Similar for the value Z_{uv} of column Z_{uv}. For all u and v:

$$\frac{Z_{uv} - 1}{\max\{X_u, X_v\} + c} \;<\; f(uv, t) \;<\; \frac{Z_{uv} + c}{\max\{X_u, X_v\}}. \qquad (\mathtt{c})$$

– For all u and v: $f(uv, 1), f(uv, 2), \ldots$ has at most k oscillations. (k)

Note that for decreasing c and k, it becomes easier for Alice to win.

Discussion. If Alice places a token in a row with small index, Bob has a dilemma: either he can change the function f, or he can place tokens on the other board to restore the ratios in (c). In the first case, he might increase the number of oscillations in (k), while in the second case, he exhausts his limited capacity to place tokens on rows of small indices, (by the row restriction, at most $1 + 2^1 + \ldots + 2^{i-1} = 2^i - 1$ tokens can be placed below row i in each grid G).

[5] Formally, we associate sets $\{u, v\}$ with 2 elements to an unordered pair (u, v), and singleton sets $\{u\}$ to the pair (u, u).

Remark. The game has at most $O(n2^{2n})$ rounds, because in each round, Alice must place at least 1 token, and by the row restriction, Alice can place at most $O(n2^{2n})$ tokens on all grids. Hence, the game above is finite and has full information. This implies that either Alice or Bob has a winning strategy.

Lemma 6. *Let $k\colon \mathbb{N}\times\mathbb{N}\to\mathbb{Z}$ be such that Alice has a winning strategy in the game $\mathcal{G}_{n,c,k(n,c)}$. Then for every 0-approximation of N' there exists a constant c such that for large n, the 0-approximation has more than $k(n,c)$ oscillations on n-bit inputs.*

Proof. The idea of the proof is to use any limit approximation f' to construct a strategy for Bob. By assumption there exists some winning strategy for Alice, and we let it play against this strategy for Bob. Then we show that Bob satisfies the row restrictions and requirement (c). Since Alice's strategy is winning, we conclude that requirement (k) must be violated. Our construction implies that f has fewer oscillations than f', thus also f' has more than $k(n,c)$ oscillations.

It suffices to prove the lemma for the largest function $k(n,c)$ for which Alice wins the game $\mathcal{G}_{n,c,k(n,c)}$. This function k is computable, since the game is finite, and for each value we can determine whether Alice has a winning strategy by brute force searching all strategies.
- Let $\mathrm{C}_s(\cdot|\cdot)$ represent an upper approximation of $\mathrm{C}(\cdot\,|\,\cdot)$.
- Let $\mathrm{C}(u\leftrightarrow v)=\max\{\mathrm{C}(u\,|\,v),\mathrm{C}(v\,|\,u)\}$ and similar for $\mathrm{C}_s(u\leftrightarrow v)$.
- Let f' be a 0-approximation of N'. Without loss of generality, we assume $f'(u,v,t)=f'(v,u,t)$.

For all c and n, we present a run of the game $\mathcal{G}_{n,c,k(n,c)}$. The mapping from c and n to a (transcript of) this run is computable. First, we fix a winning strategy of Alice in the game $\mathcal{G}_{n,c,k(n,c)}$ in a computable way. For example, we may brute force search all strategies and select the first winning strategy that appears. Let $r_0=1$. Consider the game in which Alice plays this strategy, and Bob replies as follows.

Bob's strategy. At round t, Bob searches for a value s with $s>r_{t-1}$ such that for all u and v:

(i) $\mathrm{C}_s(u\,|\,n)<X_u+c$ and $C_s(u\leftrightarrow v)<Z_{uv}+c$,
(ii) $f'(u,v,s)=\frac{\mathrm{C}_s(u\leftrightarrow v)}{\max\{\mathrm{C}_s(u\,|\,n),\mathrm{C}_s(v\,|\,n)\}}$.

If such an s is found, he sets $r_t=s$ and $f(uv,t)=f'(u,v,s)$ for all u and v. For all u he places a token in column X_u at row $\mathrm{C}_s(u\,|\,n)$. For all unordered pairs $\{u,v\}$, he places a token in column Z_{uv} at row $\mathrm{C}_s(u\leftrightarrow v)+1$. *End of Bob's strategy.*

We first show that if Bob does reply, he satisfies the row restriction. For $\mathrm{G}=\mathrm{X}$ this holds because there are at most 2^i programs of length i, and hence, at most 2^i strings u with $\mathrm{C}_s(u)=i$ for some s. For $\mathrm{G}=\mathrm{Z}_u$, this holds because $\mathrm{C}_s(u\leftrightarrow v)=i$ implies $\mathrm{C}(v\,|\,u)\leq i$, and there are less than 2^{i+1} such v.

Assuming that Bob plays in round t, requirement (c) holds. Indeed, after Bob's move and for $s = r_t$, condition (i) implies:

$$X_u \;\leq\; \mathrm{C}_s(u \,|\, n) \;<\; X_u + c \qquad \text{and} \qquad Z_{uv} - 1 \;\leq\; \mathrm{C}_s(u \leftrightarrow v) \;<\; Z_{uv} + c.$$

Together with (ii) and $f(t, u, v) = f'(s, u, v)$, this implies requirement (c).

We show that for large c, there always exists an s such that (i) and (ii) are satisfied, and hence, Bob plays in each round. Since f' is a 0-approximation, requirement (ii) is true for large s, and this does not depend on c. We show that (i) is also satisfied. To prove the left inequality, we first construct a Turing machine M. The idea is that the machine plays the game above, and each time Alice places a token in a cell of column X_u with row index i, it selects an unassigned i-bit string, and assigns to it the output u. Thus on input a string p and integers c, n, it plays the game, waits until the p-th token is placed in the row with index equal to the length of p, and it outputs the column's index, (which is an n-bit string). The row restriction implies that enough programs are available for all tokens. Hence, $\mathrm{C}_M(u \,|\, n, c) \leq i$, whenever Alice places a token in X_u at height i. By optimality of the Turing machine in $\mathrm{C}(\cdot \,|\, \cdot)$, we have $\mathrm{C}(u \,|\, n, c) \leq X_u + O(1)$ for all u, and hence,

$$\mathrm{C}(u \,|\, n) \;\leq\; X_u + O(\log c).$$

For large c, this is less than $X_u + c$. By a similar reasoning, we have $\mathrm{C}(u \leftrightarrow v) < Z_{uv} + c$, because each time Alice places a token in row i of column Z_{uv}, we assign 2 programs of length i: one that outputs u on input v, n, c, and one that outputs v on input u, n, c. Thus, for large s, also requirement (i) is satisfied, and Bob indeed plays at any given round, assuming he played in all previous rounds.

Recall that Alice plays a winning strategy, and that Bob satisfies the row restriction and requirement (c). Hence, requirement (k) must be violated, i.e., for some pair (u, v), the sequence $f(uv, 1), f(uv, 2), \ldots$ has more than $k(n)$ oscillations. Since r_t is increasing in t, this sequence is a subsequence of $f'(u, v, 1), f'(u, v, 2), \ldots$, and the latter must also have more than $k(n)$ oscillations. This implies the lemma.

To prove Theorem 2 we need a version of the previous lemma for the prefix distance.

Lemma 7. *Under the assumption of Lemma 6, every 0-approximation of* N *has more than* $k(n, 5 \log n)$ *oscillations on* n*-bit inputs for large* n*.*

Proof. As a warm up, we observe that

$$\mathrm{K}(x) \;\leq\; \mathrm{C}(x \,|\, n) + 4 \log n + O(1).$$

Indeed, we can convert a program on a plain machine that has access to n, to a program on some prefix-free machine without access to n, by prepending prefix-free codes of the integers n and $\mathrm{C}(x \,|\, n)$. Each such code requires $2 \log n + O(1)$ bits, and hence the inequality follows.

We modify the proof above by replacing all appearances of $\mathrm{C}(x \mid n)$ by $\mathrm{K}(x)$, of $\mathrm{C}(x \mid y)$ by $\mathrm{K}(x \mid y)$, and similarly for the approximations $\mathrm{C}_s(\cdot \mid \cdot)$. We also set $c = 5 \log n$ and assume that f' is a 0-approximation of N. In Bob's strategy, no further changes are needed.

The row restriction for Bob is still satisfied, because the maximal number of halting programs of length i on a prefix-free machine is still at most 2^i. Requirement (c) follows in the same way from items (i) and (ii) in Bob's strategy. It remains to prove that for large c and s, these conditions (i) and (ii) are satisfied. Item (ii) follows directly, since f' is a 0-approximation of N.

For item (i), we need to construct a prefix-free machine M'. This is done in a similar way as above, by associating tokens in row i to programs of length i, but we also need to prepend 3 prefix-free codes: for the row index, for n, and for c. This implies

$$\mathrm{K}(u) \leq X_u + 4 \log n + O(\log c).$$

Recall that $c = 5 \log n$. Hence, this is at most $X_u + c$ for large n. The lemma follows from the violation of requirement (k) in the same way as before.

4 Total update of ε-approximations, the game

We adapt the game for the proof of Theorem 1.

Description of game $\mathcal{H}_{n,\varepsilon,a}$, where $\varepsilon > 0$ and $a \geq 0$ are real numbers. The game is the same as the game of the previous section, except that requirements (c) and (k) are replaced by:

– For all u and v with $\max\{X_u, X_v\} \geq \sqrt{n}$:

$$\left| f(u,v,t) - \frac{Z_{uv}}{\max\{X_u, X_v\}} \right| \;\leq\; \varepsilon. \qquad (\epsilon)$$

– For all u and v with $\max\{X_u, X_v\} \geq \sqrt{n}$:

$$\sum_{s=1}^{t-1} |f(u,v,s) - f(u,v,s+1)| \;\leq\; a. \qquad (\mathtt{a})$$

Remarks.
- We call the sum in (a), the *total update* of f. Similar for the total update of an ε-approximation.
- The threshold $\sqrt{n}$ is chosen for convenience. Our proof also works with any computable threshold function that is at least super-logarithmic and at most n^α for some $\alpha < 1$.

Lemma 8. *Let $a \colon \mathbb{N} \to \mathbb{R}$. Suppose that for large n, Alice has a winning strategy in the game $\mathcal{H}_{n,\varepsilon,a(n)}$. Fix $\varepsilon' < \varepsilon$, and an ε'-approximation f' of either* N′ *or* N. *Then, for large n, there exist n-bit inputs for which the total update of f' exceeds $a(n)$.*

Proof. We first consider an ε'-approximation f' of N', and at the end of the proof we explain the modifications for N. The proof has the same high-level structure as the proof of Lemma 6: from f' we obtain a strategy for Bob that is played against Alice's winning strategy. Then, from the violation of (a) we conclude that the total update of f' exceeds $a(n)$.

Let n be large such that Alice has a winning strategy in the game $\mathcal{H}_{n,\varepsilon,a(n)}$. We consider a run of the game where Alice plays a computably generated winning strategy and Bob's replies are as follows.

Bob's strategy. He searches for an $s > r_{t-1}$ such that for all u and v with $\max\{\mathrm{C}_s(u), \mathrm{C}_s(v)\} \geq \sqrt{n}$:

(i) $\mathrm{C}_s(u \,|\, n) \leq X_u + c$ and $C_s(u \leftrightarrow v) \leq Z_{uv} + c$,
(ii) $\left| f'(u,v,s) - \frac{\mathrm{C}_s(u \leftrightarrow v)}{\max\{\mathrm{C}_s(u \,|\, n), \mathrm{C}_s(v \,|\, n)\}} \right| \leq \varepsilon'$,

If such an s is found, let $r_t = s$. Bob chooses $f(uv,t) = f'(u,v,s)$ for all u and v. For all u he places a token in column X_u at row $\mathrm{C}_s(u \,|\, n)$. For all unordered pairs $\{u,v\}$, he places a token in column Z_{uv} at row $\mathrm{C}_s(u \leftrightarrow v) + 1$. *End of Bob's strategy.*

For similar reasons as above, we have that for some c and for large s, requirements (i) and (ii) are satisfied. This implies that for some c, Bob always reacts.

We now verify that for large n, requirement (ϵ) holds. Recall that we need to check the inequality when the denominator is at least $\sqrt{n}$. After Bob's move we have again that

$$X_u \leq \mathrm{C}_s(u \,|\, n) < X_u + c \qquad \text{and} \qquad Z_{uv} - 1 \leq \mathrm{C}_s(u \leftrightarrow v) < Z_{uv} + c. \quad (*)$$

Since $\mathrm{N}' \leq e$ for some constant e, we may also assume that $f' \leq e$, because truncating f' can only decrease the number of oscillations. This and item (ii) imply that if n is large enough such that

$$(c+1)\frac{e+1}{\sqrt{n}} \leq \varepsilon - \varepsilon', \quad (**)$$

inequality (ϵ) is indeed satisfied.

Because Bob loses, requirement (a) must be violated. Since the total update of f is at least the total update of f' as long as the $\sqrt{n}$-threshold is not reached, this implies that every ε'-approximation has total update more than $a(n)$. The statement for N' is proven.

The modifications for N are similar as in the previous section. Instead of choosing c to be a constant, we again choose it to be $5 \log n$, and for the same reasons as above, this makes (*) true if we replace conditional plain complexity by (conditional) prefix complexity. This increase from constant to logarithmic c increases the minimal value of n in (**) only by a factor $O(\log^2 n)$. Otherwise, nothing changes in the above argument. The lemma is proven.

5 Conclusion

We have proven that statements about the incomputability of the normalized information distance are equivalent to the existence of winning strategies in a game. To prove the main results, we need to present these winning strategies. This is done in an extended version of the paper that is available on ArXiv.

References

1. Klaus Ambos-Spies, Wolfgang Merkle, and Sebastiaan A Terwijn. Normalized information distance and the oscillation hierarchy. *arXiv preprint arXiv:1708.03583*, 2017.
2. Bruno Bauwens. Information Distance Revisited. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science (STACS 2020)*, volume 154 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 46:1–46:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
3. Bruno Bauwens and Alexander Shen. Complexity of complexity and maximal plain versus prefix-free Kolmogorov complexity. *Journal of Symbolic Logic*, 79(2):620–632, 2013.
4. Charles H. Bennett, Péter Gács, Ming Li, Paul M.B. Vitányi, and Wojciech H. Zurek. Information distance. *IEEE Transactions on information theory*, 44(4):1407–1423, 1998.
5. Peter Gács. On the symmetry of algorithmic information. *Soviet Math. Dokl.*, 15:1477–1480, 1974.
6. Ming Li and Paul M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th edition*. Springer, 2019.
7. Andrei A. Muchnik, Ilya Mezhirov, Alexander Shen, and Nikolay Vereshchagin. Game interpretation of Kolmogorov complexity. unpublished, mar 2010.
8. Alexander Shen, Vladimir A. Uspensky, and Nikolay Vereshchagin. *Kolmogorov complexity and algorithmic randomness*, volume 220. American Mathematical Soc., 2017.
9. Sebastiaan Terwijn, Leen Torenvliet, and Paul M.B. Vitányi. Nonapproximability of the normalized information distance. *Journal of Computer and System Sciences*, 77:738–742, 2011.