Electronic Colloquium on Computational Complexity, Report No. 178 (2013)

Randomized communication complexity of approximating Kolmogorov complexity

Nikolay Vereshchagin [*] Email: ver@mech.math.msu.su

Moscow State University, Higher School of Economics, Yandex.

Abstract. The paper [Harry Buhrman, Michal Koucký, Nikolay Vereshchagin. Randomized Individual Communication Complexity. *IEEE Conference on Computational Complexity* 2008: 321-331] considered communication complexity of the following problem. Alice has a binary string x and Bob a binary string y, both of length n, and they want to compute or approximate Kolmogorov complexity $C(x|y)$ of x conditional to y. It is easy to show that deterministic communication complexity of approximating $C(x|y)$ with precision α is at least $n - 2\alpha - O(1)$. The above referenced paper asks what is *randomized* communication complexity of this problem and shows that for r-round randomized protocols its communication complexity is at least $\Omega((n/\alpha)^{1/r})$. In this paper, for some positive ε, we show the lower bound $0.99n$ for (worst case) communication length of any randomized protocol that with probability at least 0.01 approximates $C(x|y)$ with precision $\varepsilon n/\log n$ for all input pairs.

1 Introduction

Kolmogorov complexity of x conditional to y is defined as minimal length of a program (for a universal machine) that given y as input prints x. Assume that Alice has x and Bob has y, which are strings of length n. Is there a communication protocol to transmit y to Bob (i.e. to compute the function $I(x, y) = x$) that communicates about $C(x|y)$ bits for all input pairs (x, y)?

The trivial upper bound for communication complexity of this problem is n (Alice sends her input to Bob). If Alice knew y, she could do better: she could find $C(x|y)$ bit program transforming y to x and send it to Bob. However, without any prior knowledge of y it seems impossible to solve the problem in about $C(x|y)$ communicated bits, and the paper [2] confirms this intuition for deterministic protocols. Moreover, for deterministic protocols even testing equality $x = y$ might require much more than $C(x|y)$ bits of communication. Indeed, for every deterministic protocol that tests equality there is an input pair (x, x) on which the protocol communicates at least n bits (see e.g. [4]). On the other hand, we have $C(x|x) = O(1)$.

Surprisingly, the situation changes when we switch to randomized communication protocols. The paper [3] shows that for every positive ε there is a randomized communication protocol with public randomness that for all input pairs

[*] The work was in part supported by the RFBR grant 12-01-00864.

ISSN 1433-8092

(x, y) communicates at most $C(x|y) + O(\sqrt{C(x|y)}) + \log(1/\varepsilon)$ bits and computes $I(x, y) = x$ with error probability at most ε. That protocol runs in $O(\sqrt{C(x|y)})$ rounds.

The paper [3] asks whether it is possible to reduce the number of rounds (keeping the communication close to $C(x|y)$) or to decrease the surplus term $O(\sqrt{C(x|y)})$ in communication length. Both questions are related to the communication complexity of approximating the conditional complexity $C(x|y)$. Indeed, assume that there is a randomized communication protocol that finds $C(x|y)$ with precision α in $r(x, y)$ rounds and communicates at most $l(x, y)$ bits. Then the following randomized communication protocol computes $I(x, y) = x$ in $r(x, y) + 1$ rounds with additional error ε and communicates at most $C(x|y) + l(x, y) + \alpha + \log(1/\varepsilon)$ bits. Alice and Bob first run the given protocol to approximate $C(x|y)$. Assume that the protocol outputs an integer k. Then Alice communicates to Bob the value of randomly chosen linear mapping $A : \{0, 1\}^n \to \{0, 1\}^{k+\alpha+\log(1/\varepsilon)}$ on her x. Bob finds any x' in the set $S = \{x' \mid C(x'|y) < k + \alpha\}$ such that $Ax' = Ax$ and outputs it (we consider protocols with public randomness thus Bob knows A). By union bound the additional error probability of this protocol is at most $2^{k+\alpha}2^{-k-\alpha+\log\varepsilon} = \varepsilon$ (here $2^{k+\alpha}$ is an upper bound for the cardinality of S and $2^{-k-\alpha+\log\varepsilon}$ is the probability that $Ax' = Ax$ for any fixed $x' \neq x$).

The paper [3] shows that the worst case randomized communication complexity of approximating $C(x|y)$ with precision α in r rounds is $\Omega((n/\alpha)^{1/r})$ and asks what happens when the number of rounds is not bounded. In this paper we prove that for some positive ε every randomized protocol that for all input pairs with probability at least 0.01 approximates $C(x|y)$ with precision $\varepsilon n/\log n$ must communicate $0.99n$ bits for some input pair. That is, randomized communication complexity of approximating $C(x|y)$ is close to trivial upper bound n unless the precision is very bad (more than $\varepsilon n/\log n$).

2 Preliminaries

All logarithms in this paper have the base 2.

2.1 Kolmogorov complexity

Let U be a partial computable function that maps pairs of binary strings to binary strings. Kolmogorov complexity of a binary string x conditional to a binary y with respect to U is defined as

$$C_U(x|y) = \min\{|p| \mid U(p, y) = x\}.$$

The notation $|p|$ refers to the length of p.

We call U *universal* or *optimal* if for any other partial computable function V there is a constant c such that

$$C_U(x|y) \leqslant C_V(x|y) + c$$

for all x, y.

By Solomonoff–Kolmogorov theorem universal partial computable functions exist [5]. We fix a universal U, drop the subscript U and call $C(x|y)$ *the Kolmogorov complexity of x conditional to y*. We call U also a "universal machine". If $U(p, y) = x$ we say that "program p outputs x on input y".

Kolomogorov complexity of a string x is the minimal length of a program that prints x on the empty input Λ:

$$C(x) = C(x|\Lambda) = \min\{|p| \mid U(p, \Lambda) = x\}.$$

Kolmogorov complexity of other finite objects (like pairs of strings) is defined as follows: we fix a computable encoding of the objects in question by binary strings and declare Kolmogorov complexity of an object to be Kolmogorov complexity of its code.

For the properties of Kolmogorov complexity we refer to the textbook [5]. Actually, in this paper we do not need many of them. The first property we will need is an upper bound the number of string of small complexity: for every y and k there are less than 2^k strings x with $C(x|y) < k$. We will use also the following obvious inequality $C(x) \leqslant |x| + O(1)$. Also we will use the inequality for the complexity $C(x, y)$ of the pair of strings x, y:

$$C(x, y) \leqslant 2C(x) + C(y) + O(1),$$

which is almost obvious: a short program to print the pair (x, y) can be identified by the shortest program to print x encoded in a prefix free way (the easiest prefix free encoding doubles the length) concatenated with the shortest program to print y. Finally, we will implicitly use the fact that algorithmic transformations do not increase complexity: $C(A(x)) \leqslant C(x) + O(1)$ for every algorithm A and all x (the constant $O(1)$ depends on A but not on x).

2.2 Communication protocols

In this paper we use standard notions of a deterministic communication protocol and of a communication protocol with public randomness, as in the textbook [4]. Assume that Alice and Bob want to compute a function $f : X \times Y \to Z$ where the input $x \in X$ is given to Alice, and the input $y \in Y$ to Bob.

A deterministic communication protocol to compute such a function is identified by a rooted finite binary tree whose inner nodes are labeled with letters A (Alice) and B (Bob), labels indicate the turn to move. Additionally, each A-marked node is labeled by a function from X to $\{0, 1\}$ (different nodes may be labeled by different functions). This function identifies how the bit sent by Alice in her turn depends on her input. Similarly each B-marked node is labeled by a function from Y to $\{0, 1\}$. Each leaf of the tree is labeled by an element of Z (the output of the protocol).

Each node of the tree represents the state of the computation according to the protocol, which is the sequence of bits sent so far. The root is the initial state (no bits sent yet), the left son of a node u represents the state obtained after sending

0 in the state u and the right son of a node u represents the state obtained after sending 1 in the state u. When the current node is a leaf the computation halts, and the label of that leaf is considered as the result of protocol, which should be equal to the value of the function f on the input pair.

The depth of the protocol tree is the worst case length of communication according to the protocol.

We will consider also randomized communication protocols. A randomized communication protocol of depth d with public randomness is a probability distribution $\mathcal{P}$ over deterministic communication protocols of depth d. We say that a randomized protocol $\mathcal{P}$ computes a function f with success probability p if for all input pairs (x, y) the protocol P drawn at random with respect to $\mathcal{P}$ computes $f(x, y)$ with probability at least p.

3 Results

3.1 Deterministic protocols

Theorem 1. *If a deterministic protocol P computes $C(x|y)$ with precision α then its depth d is at least $n - 2\alpha - O(1)$.*

Proof. Indeed, let $P(x, y)$ denote the output of P on input pair (x, y). The protocol P defines a partition of the set $\{0,1\}^n \times \{0,1\}^n$ into at most 2^d rectangles such that $P(x, y)$ is constant on every rectangle from the partition [4].

Let (y, y) be a diagonal input pair, $A \times B$ the rectangle in the partition containing it and k the value of P on that rectangle. As $C(y|y) = O(1)$, we have $k \leqslant \alpha + O(1)$. Since the rectangle $A \times B$ includes $A \times \{y\}$, we have $C(x|y) \leqslant 2\alpha + O(1)$ for all $x \in A$, which implies that $|A| \leqslant 2^{2\alpha+O(1)}$. Hence the number of diagonal pairs (y', y') in $A \times B$ is at most $2^{2\alpha+O(1)}$. As the total number of diagonal pairs is 2^n, it follows that the partition should have at least $2^{n-2\alpha-O(1)}$ rectangles hence $d \geqslant n - 2\alpha - O(1)$.

3.2 Randomized protocols

For randomized protocols it is much harder to derive lower bounds for communication complexity of our problem. For fixed number of rounds a lower bound was shown in [3].

Theorem 2 ([3]). *Assume that a randomized r round protocol with shared randomness for every $(x, y) \in \{0,1\}^n \times \{0,1\}^n$ communicates at most d bits and with probability at least $p > 1/2$ produces a number k such that $k \leqslant C(x|y) < k + \alpha$. Then $d \geqslant \Omega((n/\alpha)^{1/r})$. The constant in Ω-notation depends on r and p.*

We strengthen this theorem by removing the dependence of the lower bound on r. Our lower bound holds even for protocols whose success probability p may approach 0. By technical reason we switch from conditional complexity $C(x|y)$ to complexity of the pair $C(x, y)$. By the symmetry of information [5], we have

$$|C(x, y) - (C(y) + C(x|y))| \leqslant 4 \log n + O(1).$$

As Bob can find $C(y)$ privately and transmit it to Alice in $\log n$ bits, approximating $C(x,y)$ and $C(x|y)$ with more than logarithmic precision are almost equivalent. If a protocol approximates $C(x|y)$ with precision α then it can approximate $C(x,y)$ with precision $\alpha+4\log n+O(1)$ by communicating extra $\log n$ bits, and the other way around.

Our main result shows that approximating $C(x,y)$ is hard for randomized communication protocols.

Theorem 3. *Assume that a randomized protocol of depth d with shared randomness for every $(x,y)\in\{0,1\}^n\times\{0,1\}^n$ with probability at least p produces a list of α numbers containing $C(x,y)$. Then*

$$d \geqslant n - O(\log n)(\alpha/p).$$

Here $O(\log n)$ means $c_1\log n+c_2$ where c_1 is an absolute constant and c_2 is a constant depending on the universal machine in the definition of Kolmogorov complexity and on the chosen computable encoding of pairs.

Corollary 1. *For some positive ε for all large enough n there is no randomized protocol of depth $0.99n$ that for all input pairs with probability at least 0.01 approximates $C(x,y)$ with precision $\varepsilon n/\log n$. The same statement holds for $C(x|y)$ in place of $C(x,y)$.*

Proof (Proof of Theorem 3). First notice that it suffices to prove the statement for $\alpha=1$. Indeed, if a protocol computes a list with α entries containing $C(x,y)$ with probability p then a randomly chosen entry of the list equals $C(x,y)$ with probability p/α. Thus we will assume that $\alpha=1$.

Assume that there is a randomized protocol of depth d that computes $C(x,y)$ for every input pairs (x,y) with success probability at least p. By Yao's principle [6], it follows that for any probability distribution μ on pairs $(x,y)\in\{0,1\}^n\times\{0,1\}^n$ there is a deterministic protocol of depth d that computes $C(x,y)$ on a fraction at least p of input pairs with respect to μ. Thus it suffices to find a distribution μ such that every deterministic protocol that computes $C(x,y)$ on a fraction at least p of input pairs with respect to μ has large depth.

On the top level the construction of μ is the following. We find a family of distributions μ_i where $i=l+1,\dots,2n$ for some $l<2n$. Each μ_i will have the following properties:

(1) for a random pair (x,y) chosen at random with respect to μ_i with high probability $C(x,y)$ is close to i;
(2) the μ_i-probability of any rectangle $A\times B\subset\{0,1\}^n\times\{0,1\}^n$ is not much larger than its uniform measure $|A\times B|/2^{2n}$.

Then we will let μ to be the arithmetic mean of μ_i. The following lemma explains why such a construction can work. In that lemma μ_i, $i=l+1,\dots,m$, is a family of distributions over $\{0,1\}^n\times\{0,1\}^n$, μ stands for their arithmetic mean, and f is any function from $\{0,1\}^n\times\{0,1\}^n$ into $\mathbb{N}$.

Lemma 1. *Assume that for every i the μ_i-probability of the set*

$$\{(x,y) \mid i-a < f(x,y) < i+b\}$$

is at least $1-\gamma$. Assume further that for every i the μ_i-probability of every rectangle $A \times B \subset \{0,1\}^n \times \{0,1\}^n$ is at most

$$\varepsilon|A \times B| + \delta.$$

Then every deterministic protocol of depth d computes f correctly on a fraction at most

$$\gamma + (\varepsilon 2^{2n} + \delta 2^d)(a+b)/(m-l)$$

of input pairs with respect to μ.

Proof. We will view μ as the marginal of the distribution ν over triples (x,y,i) where $\nu(x,y,i) = \mu_i(x,y)/(m-l)$.

Fix a deterministic protocol P of depth d and call $P(x,y)$ its output on input pair (x,y). Notice that the upper bound we have to show has the term γ that corresponds to the upper bound for the μ_i-probability of the event $f(x,y) \notin (i-a; i+b)$, which holds for all i. Thus the ν-probability of this event is also at most γ. It suffices to show that ν-probability of the event "$P(x,y) = f(x,y) \in (i-a; i+b)$" is at most $(a+b)(\varepsilon + \delta 2^d)/(m-l)$. Obviously, this event is included in the event "$i \in (P(x,y)-b; P(x,y)+a)$" hence it suffices to upper bound the ν-probability of the latter event.

The protocol P defines a partition of the set $\{0,1\}^n \times \{0,1\}^n$ into at most 2^d rectangles such that $P(x,y)$ is constant on every rectangle from the partition [4]. The contribution of every rectangle $A \times B$ from the partition to the probability of the event is the sum of $\mu_i(A \times B)/(m-l)$ over all $i \in (k-b; k+a)$ where k stands for the value of $P(x,y)$ on the rectangle. By union bound and by the assumption this contribution is at most $(a+b)(\varepsilon|A \times B| + \delta)/(m-l)$. Summing up the contributions of all rectangles we obtain the upper bound $(a+b)(\varepsilon 2^{2n} + \delta 2^d)/(m-l)$.

We will apply this lemma for the function $f(x,y) = C(x,y)$. To construct μ_i we will need the following combinatorial lemma.

Lemma 2. *For every $n \geqslant 1$ and every $3 < i \leqslant 2n$ there is a bipartite graph $G_{n,i}$ whose left and right nodes are all binary strings of length n, that has at least 2^{i-1} and at most 2^{i+1} edges and for every left set A and right set B with*

$$\log|A|, \log|B| > 2n - i + \log n + 4$$

the rectangle $A \times B$ has at most $|A| \cdot |B| \cdot 2^{i-2n+1}$ edges.

Let us finish the proof of the theorem assuming this lemma. Let $l < 2n$ be an integer number to be chosen later. Apply Lemma 2 to all $i = l+1, \ldots, 2n$. The number of edges $E_{n,i}$ in the resulting graph is between 2^{i-1} and 2^{i+1}. Let μ_i be the uniform distribution over the edges of $G_{n,i}$. We will show that the assumptions of Lemma 1 are met for some small $\gamma, \varepsilon, \delta, a+b$ and $m = 2n$.

Let us start with the first assumption. We may assume that the graph $G_{n,i}$ is computable given n, i (using the brute force search we can find the first graph satisfying the lemma). Thus Kolmogorov complexity of each edge in $G_{n,i}$ is at most $i+2\log n+O(1)$ (every edge can be identified by a $2\log n+O(1)$ bit prefix code of n followed by $i+1$ bit index of the edge). On the other hand, all but a fraction of $1/n$ of its edges have complexity at least

$$\log E_{n,i} - \log n \geqslant i - 1 - \log n.$$

Thus the first condition of the lemma holds for $a = \log n+O(1)$, $b = 2\log n+O(1)$ and $\gamma = 1/n$.

The second condition. Assume first that both $\log|A|$ and $\log|B|$ are larger than $2n - i + \log n + 4$ (this bound comes from of Lemma 2). The probability that a random edge from $G_{n,i}$ falls into $A \times B$ is at most the number of edges in $A \times B$ divided by the total number of edges in $G_{n,i}$. By Lemma 2 the number of edges in $A \times B$ is at most $|A \times B| \cdot 2^{i-2n+1}$ and $E_{n,i}$ is at least 2^{i-1}. Hence

$$\mu_i(A \times B) = O(|A \times B|/2^{2n}).$$

Otherwise either $|A|$, or $|B|$ is less than $2^{2n-i+\log n+4}$ and we use the trivial upper bound $|A \times B| \leqslant 2^n \times 2^{2n-i+\log n+4}$ for the number of edges of $G_{n,i}$ in $A \times B$ and the inequality $i > l$. We have

$$\begin{aligned}\mu_i(A \times B) \leqslant &|A \times B|/2^{i-1} = O(2^{3n-2i+\log n})\\ =&O(2^{3n-2l+\log n})).\end{aligned}$$

Thus the second condition holds for

$$\varepsilon = O(2^{-2n}) \text{ and } \delta = O(2^{3n-2l+\log n}).$$

By Lemma 1 if a deterministic depth d protocol computes $C(x,y)$ on a fraction p of input pairs with respect to μ then

$$p \leqslant \frac{1}{n} + \frac{(1+2^{d+3n-2l+\log n})(c_1\log n + c_2)}{2n-l}. \tag{1}$$

The constant c_2 depends on the choice of the universal machine in the definition of $C(x,y)$. The constant c_1 is absolute.

By Yao's principle Equation (1) also holds for success probability of every depth d randomized protocol to compute $C(x,y)$. We may assume that $p \geqslant 2/n$ (otherwise the statement of the theorem is trivial). That is we can drop the term $\frac{1}{n}$ in the right hand side of (1) at the expense of halving the left hand side:

$$\frac{p}{2} \leqslant \frac{(1+2^{d+3n-2l+\log n})(c_1\log n + c_2)}{2n-l}. \tag{2}$$

Now we have to choose l so that this inequality yields the best lower bound for d. A simple analysis reveals that an almost optimal choice of l is such that the exponent in the power of 2 in the right hand side of (2) is 0, that is $l = (d + 3n + \log n)/2$. Plugging such l in (2), we obtain

$$\frac{p}{2} \leqslant \frac{4(c_1 \log n + c_2)}{n - \log n - d}.$$

The statement of the theorem easily follows.

It remains to prove Lemma 2. The lemma is proved by a probabilistic method. We will show that a randomly chosen graph has the desired properties with positive probability. The probability distribution over graphs is defined as follows. Every pair (left node, right node) is an edge of the graph with probability 2^{i-2n} and decisions for different pairs are independent.

We have to show that both requirements hold with probability more than one half. To this end we will use the Chernoff bound in the exponential form [1, Cor A.1.14]: for any independent random variables $T_1, \dots, T_k$ with values 0,1 the probability that their sum T exceeds twice the expectation $\mathrm{E}T$ of T is less than $2^{-\mathrm{E}T/4}$ and the probability that T is less than $\mathrm{E}T/2$ is less than $2^{-\mathrm{E}T/6}$.

The first requirement states that the number of edges in the graph is between 2^{-1} and 2^{i+1}. The expected number of edges is 2^i. Hence by Chernoff bound the probability that the requirement is not met is at most $2^{-2^i/4} + 2^{-2^i/6} < 1/2$, as $i \geqslant 4$.

The second requirement states that for all A, B of cardinality at least $2^{2n-i+\log n+4}$ the number of edges in $A \times B$ does not exceed its expectation twice. Fix a and b greater than $2^{2n-i+\log n+4} \geqslant 32$. Fix A and B of sizes a, b respectively. The expected number of edges that connect A and B is $ab2^{i-2n}$. Thus the probability that the number of edges between A and B exceeds its average two times is at most $2^{-ab2^{i-2n-2}}$. The number of possible A's of size a is at most 2^{na}. Similarly, the number of possible B's of size b is at most 2^{nb}. By union bound, the probability that there are A and B of sizes a, b respectively, that violate the statement of the theorem is at most $2^{nb+na-ba2^{i-2n-2}}$. The exponent in this formula can be written as the sum of $b(n - a2^{i-2n-3})$ and $a(n - b2^{i-2n-3})$. The lower bound for $|A|, |B|$ was chosen so that both terms $n - a2^{i-2n-3}$ and $n - b2^{i-2n-3}$ be less than $-n$. By union bound the probability that there are A and B, that violate the statement of the theorem is at most

$$\sum_{b,a=32}^{2^n} 2^{-bn-an} = \sum_{b=32}^{2^n} 2^{-bn} \sum_{a=32}^{2^n} 2^{-an} < 1/2.$$

References

1. Noga Alon, Joel Spencer, *The probabilistic method.* John Wiley & sons, 2nd edition, 2000.
2. Harry Buhrman, Hartmut Klauck, Nikolai K. Vereshchagin, Paul M. B. Vitányi. "Individual Communication Complexity". Symposium on Theoretical Aspects of Computer Science, 2004. LNCS v. 2996, P. 19–30.

3. Harry Buhrman, Michal Koucký, Nikolay Vereshchagin. Randomized Individual Communication Complexity. *IEEE Conference on Computational Complexity* 2008: 321-331
4. E. Kushilevitz, N. Nisan, *Communication Complexity*, Cambridge University Press, 1997.
5. M. Li, P. Vitányi. An Introduction to Kolmogorov Complexity and its Applications. Springer Verlag, 1997.
6. A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual IEEE Symposium on Foundation of Computer Science*, pages 222–227, 1977