On strong definition of information distance

Mikhail V. Vyugin*

Abstract

C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, and W.H. Zurek defined information distance between two strings x, y as

$$d(x,y) = max\{K(x \mid y), K(y \mid x)\}$$

where $K(x \mid y)$ is the conditional Kolmogorov complexity. It is natural to consider information distance in the stronger sense: we say that information distance between x, y equals to n if

$$K(x \mid y) \approx n, K(y \mid x) \approx n.$$

The main result of this paper says that a constant c exists such that for any n the following holds: for any string x such that $K(x) \geqslant 3n + c$ there exists a string y with given information distance between x and y up to an additive constant, i.e.

$$n \leqslant K(x \mid y), K(y \mid x) \leqslant n + c.$$

1 Introduction

We write *string* to denote a finite binary string. Other finite objects, such as pairs of strings, may be encoded into strings in natural ways. The conditional Kolmogorov complexity, $K(x|y)$ of x relative to y is the length of the shortest program that computes x given y as an input. For precise definition and main properties of conditional complexity see [2].

In [1] information distance between two strings x, y was defined as $d(x,y) = max\{K(x \mid y), K(y \mid x)\}$. It is natural to consider information distance in the stronger sense: we say that information distance between x, y equals to n if

$$K(x \mid y) \approx n, K(y \mid x) \approx n.$$

The last two equalities can be considered in three ways: up to an additive $O(\log(K(x)+K(y)))$ term, up to an additive $O(\log n)$ term or up to an additive constant.

*Dept. of Mathematical Logic and Theory of Algorithms, Moscow State University, Vorobjevy Gory, Moscow 119899, Russia. E-mail: misha@vyugin.mccme.ru.

Consider some questions relative to this strong notion of information distance.

The first question is following. Fix any string x of complexity bigger than n. Does there exist a string y with information distance between x and y equal to n? We can consider this question in the three different ways described above. If we consider it up to an additive $O(\log K(x))$ term then it has a simple positive answer. Indeed, given x and $K(x)$ we can get the shortest program p to compute x. This program is a *random* string, i.e. its complexity is close to its length. Let us change the last part of p of length n by a string r of length n which is independent of p i.e. $K(p \mid r) = K(r \mid p) = n + const$. The resulting string y is needed.

Consider the above question up to an additive $O(\log n)$ term. It follows from Theorem 1 and Remark 2 that it also has a positive answer. It also follows from Theorem 1 and Remark 1 that up to an additive constant the answer is positive for x of complexity at least $2n$. We do not know is the latter result still holds for x of complexity between n and $2n$.

The second question is following. Is it right that for any x of complexity at least n there exists y such that

$$K(x \mid y) = n, K(y \mid x) = m$$

up to an additive $O(\log(n+m))$ term? For what m and n it is right?

There are two cases. The first case is $m \geqslant n - \alpha$ where α is not too big. In this case such y always exists. It is a simple corollary of the case $m = n$. The second case is $m < n - \alpha$ where α is big enough. An.Muchnik proved that the answer in this case is negative.

Note that up to $O(K(x))$ the answer is positive in all cases.

Acknowledgements. N.Vereshchagin and A.Shen explained me the main results and methods of Kolmogorov complexity. The problem solved in this paper was posed by An.Muchnik. He helped me to simplify the proof. Author deeply grateful them for very useful discussions.

2 Existence of a string with given information distance from x

Theorem 1. *There exists a constant c such that for any n the following holds: for any string x such that $K(x) \geqslant 3n + c$ there exists a string y such that*

$$n \leqslant K(x \mid y), K(y \mid x) \leqslant n + c.$$

Proof. Fix n. We will prove the theorem using the following game. There are two players - Man and Nature. Let $X = Y = (0,1)^*$. Consider X as a set of left side vertices of a bipartyte graph and Y as a set of right side vertices. The process of the game is a process of constructing the graph.

At the start of the game all pairs of vertices of the form (z, z) are connected by Man's edges. At Man's step he can connect some left side vertex x to some

right side vertex y by an undirected edge. He can also declare some left side vertices as "simple" (mark them). At its step Nature can connect some left side vertex x to some right side vertex y by a directed edge from x to y or vice versa. Rule of the game puts following restrictions:

- (1) Maximum number of Man's edges outgoing from one vertex is $2^{n+1}-1$. This restriction guarantees the upper bound on both conditional complexities.
- (2) Maximum number of directed edges outgoing from one (left or right side) vertex is $2^n - 1$. It is necessary to get the lower bound on the conditional complexities.
- (3) Maximum number of marked vertices is 2^{3n+1}. This guarantees that all "simple" strings x have complexity less than $3n + c$.

We assume that the game always spents infinitely long time. Note that the game is determined by parameter n.

We say that undirected edge connecting vertices x and y is "free" if x and y are not connected by Nature. If Nature connects such vertices, we say that undirected edge is "covered". Man's goal is to guarantee that after infinitely long time for any left side vertex x which is not marked there is a "free" undirected edge outgoing from it.

Note that for any left side vertex after sufficiently long time no new edges outgoing from it will be covered.

Let Nature follows the next *natural* strategy. It enumerates pairs (x, y) such that $K(x \mid y) < n$ or $K(y \mid x) < n$ and connects x and y by a directed edge from right to left or from left to right respectively. This strategy is computable given n and it follows to restrictions.

By Lemma 1 which is formulated below, there exists a computable winning strategy for Man. This strategy wins against any strategy of Nature. And in particular against the natural strategy. Therefore for any left side vertex x which is not marked there is a free edge outgoing from it (after infinitely many time).

When Nature follows the natural strategy and Man follows the computable winning strategy we have a computable process of the game for any value of n. Let us run all these processes in parallel. All edges will be marked by the corresponding value of n. It will be computable process also. For each n, vertex z and undirected edge e outgoing from z we define *number* of e corresponding to z as the number of e in the set of all edges outgoing from z in order of edges' enumeration in nth process. We shall write this number as bynary string of length $n+1$ (when a string is shorter than $n+1$ we fill it by zeroes). Let x and y was connected by an undirected edge in the nth process. We use the number of the edge connecting them corresponding to x in nth process as the program to get y given x. So given such number and x we know n which is the length of this number minus one and y as the end of the edge which has given number corresponding to x in nth process.

So, we estimated the complexity $K(y \mid x)$ from above by the number $n + c$ for some constant c. The estimation of $K(x \mid y)$ we can get by the same way. The lower estimations follow from the fact that Nature did not connect x and y by a directed edge.

It remains to note that if x was marked in nth game then its complexity $K(x)$ less than $3n + c$ for some constand c. We can get this estimation by the same way as explained above.

□

Lemma 1. *In the game from the proof of Theorem 1 there exists a winning strategy for Man. This strategy is computable given parameter n.*

Proof. Consider any (left or right side) vertex z. By the *weight* of z we call a pair (k, l) where k is the number of Man's edges outgoing from z which were covered by Nature's edge going from left to right, l is the number of Man's edges covered by Nature's edges going from right to left.

Man waits when Nature covers his edge. If it never happens, he wins. The situation before Man's move is as following.

All left side vertices are divided into three classes. First class consists of all marked vertices. There are no free edges outgoing from them. The second class consists of exactly one vertex x which is not marked and there are no free edges outgoing from it. At its last move Nature covered an edge outgoing from this vertex x. The third class is the set of all other left side vertices. There is exactly one free edge outgoing from each of them. Any free edge connects vertices with equal weights. At most one free edge outgoes from each right side vertex. We shall verify that described situation holds by induction on moves of Nature and Man.

Let Nature covered Man's edge connecting vertices x and y from left and right side respectively. By induction hypothesis weights of x and y were equal and consequently they stay equal. Man has to mark x or connect it to some vertex $\tilde{y}$. Let Man connects x to such $\tilde{y}$ that $\tilde{y}$ has the same weight as x and x was not connected to $\tilde{y}$ earlier. If there is no such $\tilde{y}$ then Man marks x. After that Man again waits when Nature covers his edge.

The strategy was entirely described. It remains to prove that the number of marked vertices is bounded by 2^{3n+1}.

Let $(k, l) \neq (0, 0)$ be any fixed weight and x be a vertex of this weight. First, note that $k, l < 2^n$. ¿From this follows that the number of covered edges outgoing from one vertex is less than $2^{n+1} - 1$. Therefore the total number of Man's edges outgoing from one vertex is less than 2^{n+1}.

Free right side vertex is such vertex that there are no free edges outgoing from it. The number of left side vertices of weight (k, l) is equal to the number of right side vertices of the same weight. Therefore after Man's move the number of marked left side vertices of that weight equals to the number of free right side vertices of the same weight.

Suppose there are 2^{n+1} marked vertices of weight (k, l). Let Man has to connect x of weight (k, l) to $\tilde{y}$ of the same weight. Then such $\tilde{y}$ exists among

2^{n+1} free right side vertices of weight (k, l) (x was not connected with all of them earlier). So, we have that the number of marked vertices of any fixed weight is no more than 2^{n+1}. The number of all different weights is 2^{2n}. Therefore the total number of marked vertices is no more than 2^{3n+1}.

□

Remark 1. In the statement of Theorem 1 we can change the inequality $K(x) \geqslant 3n + c$ by inequation $K(x) \geqslant 2n + c$.

Proof. It is enough to change the definition of weight. Define weight as a difference between elements of a pair (see definition of weight in the proof of Lemma 1). Then the number of weights is no more than 2^{n+1} instead of 2^{2n} in the above proof. Reader can rewrite the above proof using new definition of weight. □

Remark 2. If $n \leqslant K(x) \leqslant 2n$ then we use the construction which was done in introduction up to $O(K(x))$, because i this case $O(K(x)) = O(n)$.

References

[1] C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, and W.H. Zurek. "Information Distance", IEEE Trans. on Information Theory **44** (1998), No 4, 1407–1423.

[2] M. Li, P. Vitányi. An Introduction to Kolmogorov Complexity and its Applications. Springer Verlag, 1997.