Topological arguments for Kolmogorov complexity[*]

Alexander Shen alexander.shen@lirmm.fr

Andrei Romashchenko andrei.romashchenko@lirmm.fr

LIRMM, CNRS & Montpellier II, on leave from IITP, Moscow

We present several application of simple topological arguments in problems of Kolmogorov complexity. Basically we use the standard fact from topology that the disk is simply connected. It proves to be enough to construct strings with some nontrivial algorithmic properties.

1 Introduction

In this paper we show that a very simple and intuitive topological technique can be surprisingly effective in algorithmic information theory. We present several examples of "topological" proofs of existence of strings with some nontrivial algorithmic properties. We focus on technical aspects of the arguments, assuming that the reader is familiar with basic notions of algorithmic information theory such as Kolmogorov complexity and the mutual information (see the classic textbook [1]).

Let us start with an example. Consider a string x that has complexity n (we consider plain complexity $C(x)$, but this does not matter for now). We want to find a string y such that $C(x|y) \approx n/2$. This can be done as follows: consider the shortest description p of x; it has length n; let y be the first half of this description. Then it is easy to check that $C(x|y) = n/2 + O(\log n)$.

However, if we want $C(x|y)$ to be close to $n/2$ with (maximal possible) precision $O(1)$, one needs a different argument. Let us start with $y = x$ (when $C(x|y) \approx 0$) and delete bits (say, at the end) one by one until we get $y = \Lambda$ (the empty string) when $C(x|y) \approx n$). When the last bit of y is deleted, the conditional complexity $C(x|y)$ changes by at most $O(1)$, so it cannot cross the threshold $n/2$ without visiting $O(1)$-neighborhood of $n/2$.

Topological arguments of this type can be used in two (and more) dimensions, though they become a bit less trivial. We provide several examples of this type in the following sections.

2 Vyugin's result and its extensions

M. Vyugin [2] has shown that for every n and for every string x with $C(x) \geq 2n + O(1)$ one can find a string y such that both conditional complexities $C(x|y)$ and $C(y|x)$ are equal to $n + O(1)$. This is proved with a (rather ingenious) game argument. As we shall see in this paper, the condition $C(x) \geq 2n$ is stronger than necessary; it is enough to assume that $C(x) \geq n + c\log n$ for some c. This can be shown (unless $C(x)$ is not very large) by a simple topological argument. (The game argument still seems to be necessary if $C(x)$ is really big compared to n.)

Similar reasoning allows us to construct y such that both complexities $C(x|y)$ and $C(y|x)$ has prescibed values with $O(1)$-precision, even if those values are different. (This question was discussed in Vyugin's paper [2], but no positive result of this type is given there except for the already mentioned case $m = n + O(1)$.) Again we need some restrictions that guarantee that prescribed values are not unreasonable large or small. Here is the exact statement.

[*] This work was supported in part by NAFIT ANR-08-EMER-008-01 grant.

E. Formenti (Ed.): AUTOMATA and JAC 2012 conferences
EPTCS 90, 2012, pp. 127–132, doi:10.4204/EPTCS.90.10

Theorem 1. *Let P be some polynomial. There exists a constant c such that for every string x and for all integers m,n such that*

- $n + c\log n \leq C(x) \leq P(n)$;
- $c\log n \leq m \leq P(n)$,

there exists a string y such that $|\,C(x|y) - n| \leq c$ and $|\,C(y|x) - m| \leq c$.

(Since $C(x|y)$ does not exceed $C(x)$, we need $C(x)$ to be greater than n; the "safety margin" of size $O(\log n)$ is assumed in the statement. We also require m and $C(x)$ to be polynomially bounded.)

Proof. Let p be the shortest description of x; it is a string of length $C(x)$. Consider also an incompressible string z of length sligthly greater than m, e.g., let $|z|$ and $C(z)$ be equal to $2m + O(1)$. Moreover, we take z independent from p, so $C(z|p)$ is also $2m + O(1)$. (Random z has these properties with positive probability; in fact, some other properties of z will be needed, see below.)

The string y is then constructed as (the encoding of) a pair (p', z') where p' and z' are prefixes of p and z (respectively); it remains to decide how long p' and z' should be. In other words, we have two parameters, $|p'|$ and $|z'|$, and the space of the parameters is a rectangle. Each point in this rectangle determines $y = (p', z')$ and is mapped to the point $(C(x|y), C(y|x))$ at the right. We need to show that some point is mapped into (n,m) with $O(1)$-precision.

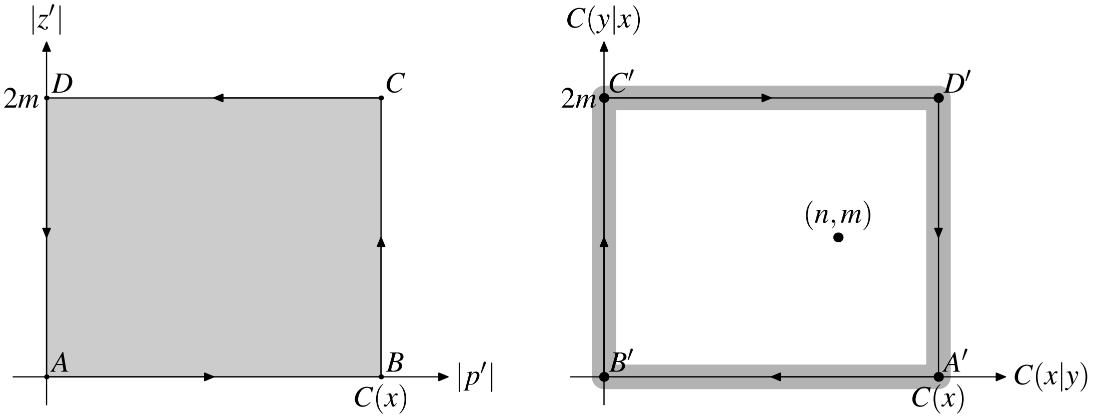

Figure 1: Each pair in the left rectangle determines $y = (p', z')$ and is mapped into a pair $C(x|y), C(y|x)$ in the right one.

To show this, we note that the mapping is "continuous" in the sense that neighbor points on the left are mapped into points at distance $O(1)$ on the right[1]. Indeed, $C(u|v)$ changes only by $O(1)$ if u or v is changed by adding or deleting the last bit. Consider a path A–B–C–D–A that goes counterclockwise around the rectangle on the left; as we shall see, the image path on the right will go clockwise (with logarithmic precision) around the rectangle and makes one turn around the point (m,n). Then we can continuously transform the path on the left into one point (since rectangle is simply connected); if its image on the right never comes close to (m,n), we get a contradiction.

Now we look closely at the path around the rectangle and its image. Note that our assumptions guarantee that $\log C(x) = \Theta(\log n)$, so we write just $O(\log)$ having in mind $O(\log n)$ or $O(\log C(x))$; note also that $O(\log m) \leq O(\log)$.

[1] Technically, *Lipschitz property* is a better name here. However, we will use discrete versions of topological results about continuous mappings, so we keep the name.

- Point A: here $y = (\Lambda, \Lambda)$, so $C(y|x) = O(1)$ and $C(x|y) = C(x) + O(1)$, so the image is $A' = (C(x), 0)$ with $O(1)$-precision.
- Edge A–B: here $y = (p', \Lambda)$. Then $C(y|x) = O(\log)$ since $C(p|x) = O(\log C(x))$ [the conditional complexity of a shortest description of x given x is $O(\log C(x))$], and the length of p' can be described by $O(\log C(x))$ bits, too. And $C(x|y)$ is somewhere between 0 and $C(x) + O(1)$.
- Point B: here $y = (p, \Lambda)$, so $C(y|x) = O(\log C(x))$ and $C(x|y) = O(1)$; the image is $B' = (0, 0)$ with $O(\log)$-precision. So the edge A–B is mapped into a path along $A'B'$ going from A' to B' with $O(\log)$-precision.
- Edge B–C: here $y = (p, z')$, so $C(y|x)$ is somewhere between 0 and $2m + O(\log)$ (recall that the length of z' is between 0 and $2m$ and requires $O(\log m) = O(\log)$ bits; $C(p|x)$ is also $O(\log)$). On the other hand, $C(x|y) = O(1)$, since p determines x.
- Point C: here $y = (p, z)$, so $C(x|y) = O(1)$ and $C(y|x) = O(\log) + C(z|x) = 2m + O(\log)$. So the image is $C' = (0, 2m)$ with $O(\log)$-precision.
- Edge C–D: here $y = (p', z)$, so $C(y|x) = 2m + O(\log)$ and $C(x|y)$ is between 0 and $C(x) + O(\log)$.
- Point D: here $y = (\Lambda, z)$, so $C(y|x) = C(z|x) + O(1) = 2m + O(1)$ and $C(x|y) = C(x|z) = C(x) + O(\log)$ since x and z have only logarithmic mutual information. So the image of D is $D' = (C(x), 2m)$ with $O(\log)$-precision.
- Edge D–A: here $y = (\Lambda, z')$, so $C(y|x)$ is between 0 and $2m + O(\log)$, and $C(x|y)$ is $O(\log)$ (note that z' can have only $O(\log)$ bit of additional information compared to z).

This analysis shows that the path on the right follows the trajectory A'–B'–C'–D'–A' with $O(\log)$-precision and therefore turns around the point (m, n) if this point is $O(\log)$-far from the boundary of the rectangle, and this is exactly what our assumption guarantees. □

3 Decreasing complexity by using an oracle

Let a and b be two strings. They have some complexities $C(a)$ and $C(b)$. If a third string t is given, we can consider the conditional complexities $C(a|y)$ and $C(b|y)$ which are (in general) smaller that $C(a)$ and $C(b)$. Now the question: can we describe the pairs $(C(a|y), C(b|y))$ that can be obtained by choosing an appropriate value of y? We answer this question for the case when a and b have small mutual information, and the answer is simple: we can get an arbitrary pair (α, β) such that $0 \le \alpha \le C(a)$ and $0 \le \beta \le C(b)$ and α, β are not too close to the endpoints of the corresponding intervals (the distance is big compared to the logarithms of complexities and to the mutual information).

Theorem 2. *For some constant c the following statement holds: for every two strings a, b and for every integers α, β such that*

- $\alpha, \beta \ge c(\log C(a) + \log C(b) + I(a:b))$;
- $\alpha \le C(a) - c(\log C(a) + \log C(b) + I(a:b))$;
- $\beta \le C(b) - c(\log C(a) + \log C(b) + I(a:b))$,

there exists a string y such that $|C(a|y) - \alpha| \le c$ and $|C(b|y) - \beta| \le c$.

Note that this statement is evidently true if instead of $O(1)$-precision we are satisfied with a precision of $O(\log C(a) + \log C(b) + I(a:b))$. Indeed, we can consider the shortest descriptions p and q for a and b and then let $y = (p', q')$ where p' is p without α last bits, q' is q without β last bits; the information distance between a, p and between b, q is logarithmic, p' and q' are independent with our precision, etc.

Proof. To get $O(1)$-precision, we need to combine the construction above with topological arguments similar to the proof of Theorem 1. Consider the shortest descriptions p and q for a and b. Then $|p| = C(a)$ and $|q| = C(b)$. For every pair (u,v) of integers such that $0 \le u \le |p|$ and $0 \le v \le |q|$ consider

$$y(u,v) = (p \text{ without } u \text{ last bits, } q \text{ without } v \text{ last bits}).$$

As we have discussed, $C(a|y(u,v))$ and $C(b|y(u,v))$ are close to u and v respectively; the distance is $O(\log C(a) + \log C(b) + I(a:b))$.

In other terms, let us consider the mapping

$$(u,v) \mapsto (C(a|y(u,v)), C(b|y(u,v));$$

it is defined on the rectangle $[0,C(x)] \times [0,C(y)]$ and is close to the identity mapping, the distance is $O(\log C(a) + \log C(b) + I(a:b))$. This mapping is also continuous in the sense explained above. Now the topological argument can be used to show that the image $O(1)$-covers the $O(\log C(a) + \log C(b) + I(a:b))$-interior of the rectangle. □

Remark. This argument can be generalized easily to three (or more) dimensions. For example, let us consider three strings a,b,c that are almost independent. In this case we get a mapping of a three-dimensional box to itself which is "continuous" and is close to identity. Then a topological argument (based on the fact that identity mapping of the two-dimensional sphere S^2 is not homotopic to the constant mapping) shows that the image of this mapping cover (with $O(1)$-precision) the interior of the box.

4 Combination with Muchnik's technique

For the case when a and b are dependent, the result of Theorem 2 looks rather weak. We can extent the area of pairs (α,β) that can be covered, if we combine the topological technique with an argument based on Muchnik's theorem on conditional description.

Theorem 3. *For some constant c the following statement holds: for every two strings a,b and for every integers α,β such that*

- $\alpha \le C(a) - c\log n$,
- $\beta \le C(b) - c\log n$,
- $-C(a|b) + c\log n \le \beta - \alpha \le C(b|a) - c\log n$,

(*where $n = C(a) + C(b)$*) *there exists a string y such that $|C(a|y) - \alpha| \le c$ and $|C(b|y) - \beta| \le c$.*

Proof. The proof is based on the following lemma.

Lemma 1. *For all strings a and b there exist strings a' and b' such that*

- $C(a') = |a'| = C(a)$ *and* $C(b') = |b'| = C(b)$,
- $C(a'|a) = O(\log n)$ *and* $C(b'|b) = O(\log n)$,
- $I(a'_l : b'_m) = \max\{0, l+m-C(a,b)\} + O(\log n)$ *for each* $l = 1,\dots,|a'|$ *and* $m = 1,\dots,|b'|$,

where $n = C(a) + C(b)$, and a'_l, b'_m denote prefixes of string a' and b' of length l and m respectively.

Proof of lemma: We use the method of conditional descriptions proposed by Andrei Muchnik. First, we apply Theorem 2 from [3] and get a conditional description of a given b and given the empty string. Muchnik's theorem guarantees that there exists a string a' such that

- $|a'| = C(a)$,
- $C(a'|a) = O(\log n)$,
- $C(a|a') = O(\log n)$,
- $C(a|a'', b) = O(\log n)$, where a'' is the prefix of a' of length $C(a|b)$.

Then, we apply the same theorem of Muchnik again: we construct a description of b conditional on $|a'|+1$ strings $a'_0, a'_1, \dots, a'_{|a'|}$. We get b' such that

- $|b'| = C(b)$,
- $C(b'|b) = O(\log n)$,
- $C(b|a'_l, b'_m) = O(\log n)$ for all $l \le |a'|$ and $m \le |b'|$ such that $l + m = C(a,b)$.

The constructed a' and b' satisfy the theorem. □

To prove the theorem we need to combine the argument from Theorem 2 and Lemma 1. Let a' and b' be the strings from Lemma. For every pair (l,m) of integers such that $0 \le l \le |a'|$ and $0 \le m \le |b'|$ we define

$$y(l,m) = (\text{first } l \text{ bits of } a',\ \text{first } m \text{ bits of } b').$$

Similarly to the proof of Theorem 2, we consider the mapping

$$(l,m) \mapsto (C(a|y(l,m)), C(b|y(l,m))),$$

which is defined on the rectangle $[0, C(a)] \times [0, C(b)]$.

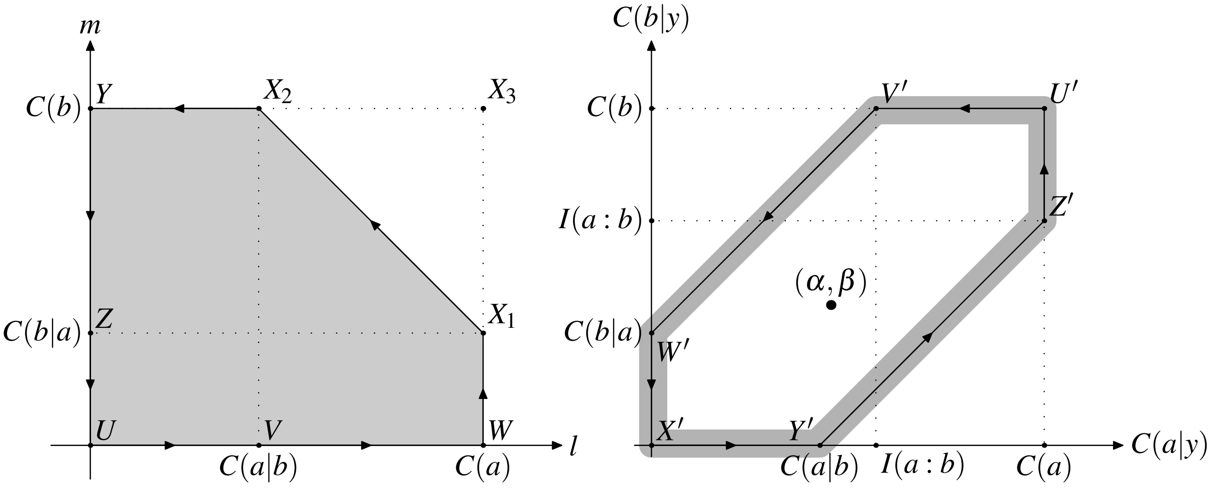

Figure 2: Each pair of integers on the left determines $y = (a'_l, b'_m)$, which is mapped to a pair $(C(a|y), C(b|y))$ on the right.

This mapping is "continuous" (neighbor points mapped into points at distance $O(1)$). Consider the path U–V–W–X_1–X_2–Y–Z–U that goes counterclockwise around the pentagon on Fig. 2. We will show that the image path will go clockwise (with logarithmic precision) around the hexagon U'–V'–W'–X'–Y'–Z'–U' on Fig. 2. This path makes a turn around the point (α,β). Hence, point (α,β) (more precisely, some point in $O(1)$-neighborhood of (α,β)) has a pre-image (l,m) in the rectangle $[0,C(x)] \times [0,C(y)]$.

Let us look closely at the path around the pentagon and its image.

- $m = 0$ and $l = 0 \ldots C(a|b)$: the image goes along U'–V' with $O(\log n)$-precision;
- $m = 0$ and $l = C(a|b) \ldots C(a)$: the image goes along V'–W' with $O(\log n)$-precision;
- $l = C(a)$ and $m = 0 \ldots C(b|a)$: the image goes along W'–X' with $O(\log n)$-precision;
- $l = C(a) - \lambda$ and $m = C(b|a) + \lambda$ for $\lambda = 0 \ldots I(a:b)$: the image remains in $O(\log n)$-neighborhood of X';
- $m = C(b)$ and $l = C(a|b) \ldots 0$: the image goes along X'–Y' with $O(\log n)$-precision;
- $l = 0$ and $m = C(b) \ldots C(b|a)$: the image goes along Y'–Z' with $O(\log n)$-precision;
- $l = 0$ and $m = C(b|a) \ldots 0$: the image goes along Z'–U' with $O(\log n)$-precision;

Thus, the path on in the image follows the trajectory U'–V'–W'–X'–Y'–Z'–U' with $O(\log)$-precision. Therefore turns around the point (α, β) if this point is $O(\log n)$-far from the boundary of the hexagon, and this is exactly what our assumption guarantees. □

Remark. Instead of the path U–W–X_1–X_2–Y–U we could take another path U–W–X_3–Y–U. The shortcut X_1–X_2 is equivalent to the longer path X_1–X_3–X_2 since all points of the triangle $X_1X_2X_3$ are mapped into $O(\log n)$-neighborhood of X' (if $l + m \geq C(a,b)$, then $y(l,m)$ contains enough information to reconstruct both a and b).

From Theorem 3 it follows in particular that if $C(a|b)$ and $C(b|a)$ are not logarithmically negligible, one can cut by factor 2 the complexities of a and b while adding an oracle.

Acknowledgements. The authors are grateful to Laurent Bienvenu who suggested to write down this simple argument, Tarik Kaced, and all the colleagues in Escape/NAFIT/Kolmogorov seminar team.

References

[1] M. Li & P. Vitányi (2008): *An introduction to Kolmogorov complexity and its applications*. 3rd ed., Springer-Verlag, New York.

[2] M. Vyugin (2002): *Information distances and conditional complexities*. Theoretical Computer Science 271(1–2), pp. 145–150 doi:10.1016/S0304-3975(01)00037-8.

[3] An. A. Muchnik (2002): *Conditional complexity and codes*. Theoretical Computer Science 271(1–2), pp. 97–109 doi:10.1016/S0304-3975(01)00033-0.