

# Increasing Kolmogorov Complexity

Harry Buhrman\*    Lance Fortnow†    Ilan Newman‡    Nikolai Vereshchagin§

September 7, 2004

**classification:** Kolmogorov complexity, computational complexity

## 1 Introduction

How much do we have to change a string to increase its Kolmogorov complexity. We show that we can increase the complexity of any non-random string of length  $n$  by flipping  $O(\sqrt{n})$  bits and some strings require  $\Omega(\sqrt{n})$  bit flips. For a given  $m$ , we also give bounds for increasing the complexity of a string by flipping  $m$  bits.

By using constructible expanding graphs we give an efficient algorithm that given any non-random string of length  $n$  will give a small list of strings of the same length, at least one of which will have higher Kolmogorov complexity. As an application, we show that BPP is contained in P relative to the set of Kolmogorov random strings. Allender, Buhrman, Koucký, van Melkbeek and Ronneberger [2] building on our techniques later improved this result to show that all of PSPACE reduces to P with an oracle for the random strings.

## 2 Increasing Complexity by Flipping Bits

Using the notation of Li and Vitányi, we use  $C_U(x)$  to represent the size of the smallest program  $p$  such that  $U(p) = x$ . We fix a universal reference computer  $U$  and let  $C(x) = C_U(x)$ .

Assume we are given a binary string  $x$ . By how much we can increase its complexity by flipping at most  $m$  bits of  $x$ ? Let  $N^m(x)$  denote the set of all strings with Hamming distance at most  $m$  from  $x$ . Let  $N^m(A)$  stand for the union of  $N^m(x)$  over  $x \in A$ .

We use the notation  $O(1), c, c_1, \dots$  for constants depending on the reference machine  $U$  and  $d, d_1, \dots$  for absolute constants. The following, rather general theorem, asserting that the complexity of any ‘typical’ string in a set can be increased by flipping  $m$  bits to the expected  $\log |N^m(A)|$  is an immediate implication of the ‘cardinality’ lower bound for Kolmogorov complexity.

**Theorem 1.** *Let  $k, m, a \leq n$  be such that the following condition hold*

*(\*) for every set  $A \subseteq \{0, 1\}^n$  with  $|A| > 2^a$ ,  $N^m(A) \geq 2^k$  for  $k < n$ , or  $N^m(A) \geq 2^n(1 - 1/c_2)$  for  $k = n$ .*

*Then, there are constants  $c_1, c_2$  depending on the reference computer such that for every string  $x$  of complexity at least  $C(x|n) \geq a + 2C(k, m|n, a) + c_1$  there is a string  $y$  obtained from  $x$  by flipping at most  $m$  bits such that  $C(y|n) \geq k$ .*

*Proof.* Consider the following set

$$B = \{x \in \{0, 1\}^n \mid C(y|n) < k \text{ for all } y \in N^m(x)\}.$$

---

\*CWI, Amsterdam

†University of Chicago.

‡Haifa University

§Moscow State University, Email: ver@mccme.ru. The work was done while visiting CWI; also supported in part by the RFBR grant 02-01-22001.

As the Kolmogorov complexity of all strings in  $N^m(B)$  is less than  $k$  we have  $|N^m(B)| < 2^k$ . In the case  $n = k$  we may upper bound  $|N^m(B)|$  better. Recall the following lower bound for the number of random strings (for the proof see [5]): for appropriate choice of  $c_2$  for every  $n$  the number of strings  $y$  of length  $n$  with  $C(y|n) \geq n$  is more than  $2^n/c_2$ . Therefore in the case  $k = n$  we have  $|N^m(B)| < 2^n(1 - 1/c_2)$ .

In both cases we thus obtain  $|B| \leq 2^a$ . The set  $B$  may be enumerated given  $k, m, n$ . Therefore every string  $x \in B$  can be described by  $m, n, k$  and its index in  $B$  of bit length  $a$ . Thus  $C(x|n) < a + 2C(k, m|n, a) + c_1$  for all  $x \in B$ , where  $c_1$  is a constant depending on the reference computer. In other words, for every  $x$  such that the last inequality is false there is  $y \in N^m(x)$  with  $C(y|n) \geq k$ .  $\square$

Theorem 1 is rather general and applies to any graph rather just the Boolean cube, when we replace ‘flipping bits’ with going to neighbors. This will be discussed in Section 3.

We now want to apply Theorem 1. For this we need to analyze the expanding properties of the Boolean cube. The complete analysis is given by the following theorem. We first introduce a notation. Let  $b(n, l)$  denote the binomial sum:  $b(n, l) = \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{l}$ .

**Theorem 2 (Harper).** *Let  $J \leq 2^n$ . Take all the strings with less than  $l$  ones and take  $J - l$  first strings with  $l$  ones in the lexicographical order, where  $l$  is chosen so that  $b(n, l - 1) < J \leq b(n, l)$ . Then the resulting set has the least  $|N^1(A)|$  among all sets  $A$  with  $|A| = J$ .*

We will use the following corollary of Harper’s theorem.

**Corollary 3.** *If  $|N^m(A)| \leq b(n, l)$  and  $l < n$  then  $|A| \leq b(n, l - m)$  and  $\frac{|N^m(A)|}{|A|} > (\frac{n-l}{l})^m$ .*

We note that the second bound is very weak and becomes trivial for  $l > n/2$ . It will be sufficient though for our applications.

*Proof.* It is enough to prove the theorem in the case  $m = 1$ . For  $m > 1$  we can use induction where inductive step is due to the case  $m = 1$ .

The first statement immediately follows from Harper’s theorem. Let us prove the second one assuming that  $l \leq n/2$ . Let  $J = |A|$ . It suffices to establish the inequality assuming that  $A$  is the worst case set defined in the Harper’s theorem. We have

$$\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{l'} < |A| = J \leq \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{l}$$

for some  $l'$ . We claim that  $l' < l$ . Indeed, otherwise  $|A| > b(n, l - 1)$  and therefore  $A$  has a string with  $l$  ones, thus  $N(A)$  has a string with  $l + 1$  ones hence  $|N(A)| > b(n, l)$ , a contradiction. For the worst case set  $A$  we will prove that  $|\Delta(A)|/|A| \geq (n - l')/(l' + 1) \geq (n - l + 1)/l$  where  $\Delta(A)$  stands for the set of strings obtained from strings in  $A$  by changing a 0 to 1 (but not vice verse). (Actually  $\Delta(A)$  and  $N(A)$  differ by only one string,  $00\dots 0$ .)

Let  $B$  consist of all strings with less than  $l'$  ones thus  $B \subset A$ . Obviously,  $\Delta(A)$  and  $\Delta(B - A)$  do not intersect, as every string in the first set has at most  $l'$  ones and every string in the second set has  $l' + 1$  ones. Therefore it suffices to prove that  $|\Delta(B)|/|B| \geq (n - l')/(l' + 1)$  and  $|\Delta(B - A)|/|B - A| \geq (n - l')/(l' + 1)$ .

The first inequality is proved as follows:  $\Delta(B)$  is the set of all strings with at most  $l'$  ones except  $00\dots 0$ , so  $|\Delta(B)| = \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{l'}$ . And  $|B| = \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{l'-1}$ . The ratio of  $i$ th term in the first sum and  $i$ th term in the second sum is  $\binom{n}{i}/\binom{n}{i-1} = (n - i + 1)/i \geq (n - l' + 1)/l' \geq (n - l')/(l' + 1)$ .

Let us prove the second inequality. Let  $x$  be a string with  $l'$  ones and let  $C_x$  denote the set of all strings with  $l'$  ones that are less than or equal to  $x$ . We claim  $|\Delta(C_x)|/|C_x|$  is a non-increasing function in  $x$ . To prove this claim it suffices to show that  $|\Delta(C_x \cup \{x'\}) - \Delta(C_x)|$  is a non-increasing function in  $x$  where  $x'$  denotes the successor of  $x$ . The set  $\Delta(C_x \cup \{x'\}) - \Delta(C_x)$  consists of all strings obtained by flipping all zeros in  $x'$  preceding the leading 1 (all other flips result in strings that are already in  $\Delta(C_x)$ ). Hence  $\Delta(C_x \cup \{x'\}) - \Delta(C_x)$  is equal to the number of zeros preceding the leading 1 in  $x'$ . And the latter number does not increase as  $x'$  increases.

For  $x$  equal to the last string with  $l'$  ones we have  $|\Delta(C_x)|/|C_x| = \binom{n}{l'+1}/\binom{n}{l'} = (n - l')/(l' + 1)$  so we are done.  $\square$

As a result we obtain the following triplets of  $k, m, a$  for which condition (\*) and hence Theorem 1 hold.

**Theorem 4.** *There is a constant  $c_3$ , such that for every  $k \leq n$ ,  $m$  and a string  $x$  of complexity at least  $C(x|n) \geq a + 2C(m|n, a) + c_3$ , there is a string  $y$  obtained from  $x$  by flipping at most  $m$  bits such that  $C(y|n) \geq k$ . Here  $a = k - \lfloor m \log((n-l)/l) \rfloor$  where  $l$  is the least number such that  $2^k \leq b(n, l)$ .*

*Proof.* Let  $l$  be as above and let  $c_1$  be the constant from Theorem 1. We first note that the conditions of Theorem 1 hold for  $a, k, m$ . Indeed, assume that  $|N^m(A)| < 2^k$ , then by the definition of  $l$ ,  $|N^m(A)| < \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{l}$  and by Corollary 3 we have  $|A| < |N^m(A)|((n-l)/l)^{-m} < 2^k((n-l)/l)^{-m} \leq 2^a$ . Hence, Theorem 1 asserts that for every string  $x$  with  $C(x|n) \geq a + 2C(m|n, a) + c_1$  there is a string  $y$  obtained from  $x$  by flipping at most  $m$  bits such that  $C(y|n) \geq k$ .

It suffices to prove that  $C(k, m|n, a) \leq C(m|n, a) + O(1) \leq \log m + O(1)$ . To this end we will prove that  $k$  can be retrieved from  $m, n, a$ . By definition  $l$  is a function of  $n, k$  and  $a$  is a function of  $n, k, m$ . The function  $l(n, k)$  is non-decreasing in  $k$  hence the function  $a(n, k, m) = k - \lfloor (m+1) \log((n-l)/l) \rfloor$  is also non-decreasing in  $k$ , as the sum of two non-decreasing functions. Moreover, the first term increases by 1 as  $k$  increments by 1. This implies that  $k$  can be retrieved from  $m, n, a$  hence  $C(k, m|n, a) \leq C(m|n, a) + O(1)$ .  $\square$

For  $p \in (0, 1)$  let  $H(p) = -p \log p - (1-p) \log(1-p)$  be the Shannon Entropy function. Note that for every  $\alpha \in [0; 1)$  there are two different  $\beta_1, \beta_2$  such that  $h(\beta_1) = h(\beta_2) = \alpha$ ; they are related by the equality  $\beta_1 + \beta_2 = 1$ . Let  $H^{-1}(\alpha)$  stand for the least of them. The function  $H^{-1}(\alpha)$  increases in the range  $(0, 0.5)$  as so does  $H$ .

**Theorem 5.** *For all  $\alpha < 1$  and  $i > 0$  there is  $m(\alpha, i)$  (depending also on the reference computer) such that for all large enough  $n$  the following holds: For all  $x$  of length  $n$  with  $C(x|n) \leq \alpha n$  there is  $y$  obtained from  $x$  by flipping at most  $m(\alpha, i)$  bits such that  $C(y|n) \geq C(x|n) + i$ . For any fixed  $i$  there is a positive  $\alpha$  such that  $m(\alpha, i) = 1$ .*

*Proof.* Fix  $\alpha$  and  $i$  and let  $x$  be such that  $C(x|n) \leq \alpha n$  and let  $k = C(x|n) + i$ . Let  $l$  be the least number such that  $b(n, l) \geq 2^k$ . We first prove that  $l \leq \beta n$  for some constant  $\beta < 1/2$ , for large enough  $n$ . This means that  $b(n, \beta n) \geq 2^k$  for some constant  $\beta < 1$ , for large enough  $n$ . Let  $\beta$  be any number in the interval  $(H^{-1}(\alpha); 1/2)$  As  $\alpha < 1$ , the interval is not empty. Then,  $b(n, \beta n) \geq \binom{n}{\beta n} \geq 2^{nH(\beta)(1+o(1))}$  (where the last inequality is standard, see e.g. [7]). Plugging in the definition of  $\beta$  can continue the inequality:  $b(n, \beta n) \geq 2^{nH(\beta)(1+o(1))} \geq 2^{n\alpha+i} \geq 2^k$  for large enough  $n$ .

Define now  $a = k - \lfloor m \log((n-l)/l) \rfloor$ . Applying Theorem 4, with  $a, k, l$  as above, we get that for every  $x$  there is  $y$  obtained from  $x$  by flipping at most  $m$  bits such that  $C(y|n) \geq k$ , as needed, provided that

$$C(x|n) \geq a + 2C(m|n, a) + c_3. \quad (1)$$

To show that (1) holds, note that  $C(m|n, a) \leq \log m$ . Plugging this, along with the definition of  $a, k$ , in (1) we get that it is enough to show that  $C(x|n) \geq C(x|n) + i - \lfloor m \log((n-l)/l) \rfloor + 2 \log m + c_3$ .

Using that  $l \leq \beta n$  and the appropriate bound on  $\beta$  we get that it is enough to have  $\lfloor m \log((1-\beta)/\beta) \rfloor > i + 2 \log m + c_3$ . Note that the definition of  $\beta$  implies that  $\beta < 1/2$  hence  $\frac{1-\beta}{\beta} > 1$ . Therefore for large enough  $m$  we will have  $\lfloor m \log((1-\beta)/\beta) \rfloor > i + 2 \log m + c_3$ .

Finally, let  $m = 1$ . Note that  $\log((1-\beta)/\beta)$  tends to infinity as  $\beta$  tends to 0. Therefore for any fixed  $i$  there is a positive  $\beta$  such that  $\lfloor m \log((1-\beta)/\beta) \rfloor > i + 2 \log m + c_3$ . Let  $\alpha$  be equal to any positive real such that  $H(\alpha) < \beta$ .  $\square$

*Remark 1.* We note that Theorem 5 works for fixed  $i$ , with respect to  $n$ , while  $m$  depends on  $i$  and  $\alpha$  for fixed  $\alpha$  or could be fixed when  $\alpha$  gets small enough. One could ask whether it might be true that  $i$  could be a function of  $n$ , e.g, could the following strengthening of Theorem 5 be true: For any  $\alpha$  (or even for some  $\alpha$ ) the complexity of a string  $x$  that is bounded by  $\alpha n$  could be increased to  $\alpha n + i(n)$  by changing only one bit. It obvious that we cannot expect such a strengthening for  $i(n) > \log n$ , as given  $x$  the complexity of any  $y$  that differs from it in one place is at most  $C(x|n) + \log n$ . Other lower bounds on  $m$  vs. the amount of increase in complexity, and the relation to  $\alpha$  are developed in Theorem 7 and Theorem 9.

Let us estimate how many bits we need to flip to increase complexity from  $k - 1$  to  $k$  when  $k$  is close to  $n$ , say for  $k = n$ .

**Theorem 6.** *For every  $x$  with  $C(x|n) < n$  by flipping at most  $c_3\sqrt{n}$  bits of  $x$  we can increase its complexity (by at least 1).*

*Proof.* Assume first that  $C(x|n) \leq n - 3$ . Let  $k = C(x|n) + 1 \leq n - 2$  and  $m = c_4\sqrt{n}$  for a constant  $c_4$  to be defined later. Apply Theorem 4. As  $2^k \leq 2^n/4$  we have  $l \leq n/2 - d_2\sqrt{n}$  and  $(n - l)/l \geq (n/2 + d_2\sqrt{n})/(n/2 - d_2\sqrt{n}) \geq 1 + 2d_3/\sqrt{n} \geq 2^{d_4/\sqrt{n}}$  for large enough  $n$ . This implies that  $a \leq k - c_4d_4$ . By Theorem 4 for every  $x$  with  $C(x|n) \geq k - c_4d_4 + 2C(m|a, n) + c_3$  there is  $y$  obtained from  $x$  by flipping at most  $m$  bits with  $C(y|n) \geq k$ . Obviously  $C(m|a, n) \leq \log c_4 + c_5$ . Therefore if  $c_4$  is large enough we have  $k - c_4d_4 + 2C(m|a, n) + c_1 \leq k - 1$  and we are done.

Assume now that  $C(x|n) \geq n - 2$ . Let us prove that by flipping  $O(\sqrt{n})$  bits we can increase the complexity of  $x$  up to  $n$ . This time we will apply Theorem 1 and Corollary 3 directly. For some  $c_3$  for  $l = n/2 + c_3\sqrt{n}$  we have  $b(n, l) \geq 2^n(1 - 1/c_2)$ , where  $c_2$  is the constant from Theorem 1. Let  $m = c_3\sqrt{n} + c_4\sqrt{n}$ , where  $c_4$  is chosen so that  $b(n, l - m) \leq 2^{n - c_5}$ , and  $c_5$  will be chosen later. Let  $a = n - c_5$  and  $k = n$ . By Corollary 3 the conditions of Theorem 1 are fulfilled. As  $a + 2C(k, m|n) + c_1 \leq n - c_5 + 2\log c_5 + c_6 \leq n - 2$  if  $c_5$  was chosen appropriately, we are done.  $\square$

Now we proceed to the lower bounds of the number of flipped bits. We will show that for every  $m$  there is  $\alpha$  such that the complexity of some strings of complexity  $\alpha n$  cannot be increased by flipping at most  $m$  bits. And there are strings for which we need to flip  $\Omega(\sqrt{n})$  bits.

**Theorem 7.** *For every  $m, k \geq 1$  there is a  $\theta(k, m) < 1$  such that for every  $\alpha > \theta(k, m)$ , for almost all  $n$  there is a string  $x$  of length  $n$  such that  $C(x|n) \leq \alpha n$  and  $C(y|n) < C(x|n) + k$  for every string  $y$  obtained from  $x$  by flipping at most  $m$  bits.*

*Proof.* Let  $\theta(k, m) = H(1/(1 + 2^{k/m}))$ , and let  $\theta(k, m) < \alpha$ . As  $k > 0$  we note that  $1/(1 + 2^{k/m}) < 1/2$ . Hence  $\theta(k, m) < 1$ . Without loss of generality assume that  $\alpha < 1$ .

Pick any  $\beta$  in the interval  $(1/(1 + 2^{k/m}); H^{-1}(\alpha))$ . Again by the bound above, and using the fact that  $H$  is monotone in the interval  $(0; 0.5)$ , the interval for  $\beta$  is non empty. Let  $l = \beta n + c_2m$  for a constant  $c_2$  to be defined later.

We first prove that every string  $x$  having at most  $l$  ones satisfies the inequality  $C(x|n) < \alpha n$ , for large enough  $n$ . Indeed, the number of such strings is equal to  $b(n, l)$  and hence is at most  $2^{nH(l/n)(1+o(1))}$  [7] (as  $l < n/2$ ). Therefore  $C(x|n) < nH(\beta)(1 + o(1)) + O(1) < n\alpha$  for large enough  $n$ , where the constant  $O(1)$  depends on  $\beta, c_2, m$  and the reference computer.

So we need to show that there is a string  $x$  having at most  $l$  ones and satisfying the second statement of the theorem. Assume that this is not the case. Let then  $x_0$  be a random string having at most  $\beta n$  ones, that is,  $C(x_0|n) \geq \log(b(n, \beta n))$ . If  $x_0$  satisfies the statement then we are done. Otherwise there is  $x_1$  having at most  $\beta n + m$  ones such that  $C(x_1|n) \geq C(x_0|n) + k$ . Repeating this argument  $c_2$  times we either find a string satisfying the statement or obtain a string  $x_{c_2}$  with  $C(x_{c_2}|n) \geq C(x_0|n) + c_2k$  having at most  $\beta n + c_2m = l$  ones. Hence  $C(x_{c_2}|n) \geq \log(b(n, \beta n)) + c_2k$ . On the other hand,  $C(x_{c_2}|n) \leq \log(b(n, l)) + 2C(l|n) + c_1 \leq \log(b(n, l)) + 2\log c_2 + c_3$ , where  $c_3$  depends on  $k, m, \alpha$  and the reference computer. To obtain the contradiction we have to show that the upper bound of  $C(x_{c_2}|n)$  is less than the lower bound. The ratio of  $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{l}$  and  $\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{\beta n}$  can be bounded using the following

**Lemma 8.** *If  $j \geq s \geq 0$  and  $j + s \leq n/2$  then*

$$\frac{b(n, j + s)}{b(n, j)} \leq 1 + \left(\frac{n - j + s}{j - s + 1}\right)^s.$$

*Proof.*

$$\begin{aligned} \frac{b(n, j+s)}{b(n, j)} &\leq 1 + \max_{i=1}^s \binom{n}{j+i} / \binom{n}{j+i-s} \\ &\leq 1 + \max_{i=1}^s \left( \frac{n-j-i+s}{j+i-s+1} \right)^s \leq 1 + \left( \frac{n-j+s}{j-s+1} \right)^s. \end{aligned}$$

□

By Lemma 8 we have  $\frac{b(n,l)}{b(n,\beta n)} \leq 2 \left( \frac{1-\beta}{\beta} \right)^{c_2 m}$ . Thus, to achieve contradiction it is enough to choose  $c_2$  so that

$$1 + c_2 m \log((1-\beta)/\beta) + 2 \log c_2 + c_3 < c_2 k. \quad (2)$$

Indeed, by the choice of  $\beta$  we have  $m \log((1-\beta)/\beta) < k$ . Hence the left hand side of (2) as a function of  $c_2$  grows slowly than the right hand side and for large enough  $c_2$  the inequality holds. □

We will show now that sometimes we need to flip  $\Omega(\sqrt{n})$  bits of  $x$  to increase its complexity even by 1.

**Theorem 9.** *There is a constant  $c$  such that for almost all  $n$  there is a string  $x$  of length  $n$  and complexity at most  $n-1$ , and such that the following holds: For every string  $y$  obtained from  $x$  by flipping at most  $\sqrt{n}/c$  bits,  $C(y|n) \leq C(x|n)$ .*

*Proof.* For every  $c_1$  there is  $c_2$  such the set of strings with at most  $n/2 - c_2\sqrt{n}$  ones has cardinality less than  $2^{n-c_1}$  and therefore the complexity of every such string is less than  $n - c_1 + 2 \log c_1 + c_3$ . Pick  $c_1$  so that  $n - c_1 + 2 \log c_1 + c_3 \leq n - 1$ .

Let  $x_0$  be a random string with at most  $l = n/2 - (c_2 + 1)\sqrt{n}$  ones. Assume that for some  $x_1$  we have  $C(y|n) \geq C(x|n) + 1$  and  $x_1$  differs from  $x_0$  in at most  $\sqrt{n}/c$  bits. In this case apply the same argument to  $x_1$  and so on,  $c$  times. Either we will obtain  $x_i$  differing from  $x_0$  in at most  $i\sqrt{n}/c$  bits satisfying the statement of the theorem, or  $x_c$  such that  $C(x_c|n) \geq C(x|n) + c$ . In the first case  $x_i$  has at most  $n/2 - (c_2 + 1)\sqrt{n} + \sqrt{n} = n/2 - c_2\sqrt{n}$  ones hence  $C(x_i|n) \leq n - 1$  and we are done.

Let us show that the second case is impossible. We have  $C(x_c|n) \geq \log \sum_{i=1}^l \binom{n}{i} + c$  and  $C(x_c|n) \leq \log \sum_{i=1}^{l+\sqrt{n}} \binom{n}{i} + 2 \log c + c_4$ . By Lemma 8 we can upper bound the ratio  $\sum_{i=1}^{l+\sqrt{n}} \binom{n}{i} / \sum_{i=1}^l \binom{n}{i}$  by

$$1 + \left( \frac{n-l+\sqrt{n}}{l-\sqrt{n}} \right)^{\sqrt{n}} = 1 + \left( \frac{n/2 + (c_2+2)\sqrt{n}}{n/2 - (c_2+2)\sqrt{n}} \right)^{\sqrt{n}} \leq c_5$$

for some constant  $c_5$  for large enough  $n$ . Therefore we will have a contradiction if  $\log c_5 + 2 \log c + c_4 < c$ . □

### 3 Increasing Kolmogorov Complexity via Expanders

In this section we will use, in place of Boolean cubes, graphs that have stronger expansion properties. Recall the theorem of Margulis [6] on explicit expanders.

**Theorem 10 (Margulis).** *Let  $k$  be an integer and  $G = (V, E)$  be the graph with vertices  $V = \{0, \dots, k-1\}^2$  where a vertex  $(x, y)$  is adjacent to vertexes  $(x, y)$ ,  $(x+1, y)$ ,  $(x, y+1)$ ,  $(x, x+y)$ , and  $(-y, x)$  (all operations are mod  $k$ ). There is a positive  $\varepsilon$  such that for every  $A \subset V$  the set  $N(A)$  of all neighbors of vertexes in  $A$  has at least  $(1 + \varepsilon(1 - |A|/|V|))|A|$  elements.*

Let  $k = 2^l$ . We will identify strings of length  $n = 2l$  and nodes of the Margulis' expander  $G$ . Let  $N^d(u)$  denote the set of all nodes at the distance at most  $d$  from  $u$  in the graph  $G$ . Let  $N^d(A)$  stand for the union of  $N^d(u)$  over  $u \in A$ .

**Theorem 11.** *There is a constant  $c_2$  such that for every node  $u$  in  $G$  with  $C(u|n) < n$  there is a node  $v \in N^{c_2}(u)$  with  $C(v|n) > C(u|n)$ .*

*Proof.* Let  $c$  be a constant to be specified later. Let  $c_1$  be the constant such that for every  $n$  the number of strings  $y$  of length  $n$  with  $C(y|n) \geq n$  is more than  $2^n/c_1$ . Let  $c_2$  be a constant such that  $(1 + \varepsilon c_1)^{c_2} \geq 2^c$ .

Assume that the statement of the theorem is false for some node  $u$ . Let us exhibit a small set containing  $u$ . Let

$$A_i = \{u' \in V \mid \forall v \in N^i(u') \ C(v|n) \leq C(u|n)\}$$

where  $i = 0, \dots, c_2$ . Obviously,  $A_{i-1} = N(A_i)$  and therefore we have  $A_0 \supset A_1 \supset \dots \supset A_{c_2}$ . By definition, all strings in  $A_{c_2}$  have Kolmogorov complexity at most  $C(u|n) < n$ . Therefore we can upper bound  $|A_0|$  in two ways:  $|A_0| \leq 2^{C(u|n)+1}$  and  $|A_0| \leq 2^n - 2^n/c_1$ . By expansion property we have

$$|A_0| \geq (1 + \varepsilon c_1)|A_1| \geq \dots \geq (1 + \varepsilon c_1)^{c_2}|A_{c_2}| \geq 2^c|A_{c_2}|.$$

Hence  $A_{c_2}$  is small,  $|A_{c_2}| \leq 2^{-c}|A_0| \leq 2^{C(u|n)+1-c}$ . Since  $u$  is in  $A_{c_2}$  and  $A_{c_2}$  can be enumerated given  $l$  and  $C(u|n)$ , we can describe  $u$  by its index in the enumeration of  $A_{c_2}$  of length  $C(u|n) + 1 - c$  and by  $c$  (and  $C(u|n)$  can be computed from the length of the index and  $c$ ). Hence  $C(u|n) \leq (C(u|n) + 1 - c) + 2 \log c + O(1)$ . If  $c$  is large then this is a contradiction.  $\square$

Using Theorem 11 we may design a polynomial time algorithm that having access to the oracle  $\tilde{R} = \{x \mid C(x \mid |x|) \geq |x|\}$  for every even length  $2l$  finds a string in  $\tilde{R}$  of length  $2l$ .

**Theorem 12.** *There is an algorithm that having access to the oracle  $\tilde{R} = \{x \mid C(x \mid |x|) \geq |x|\}$  for every even length  $2l$  in time  $\text{poly}(l)$  finds a string in  $\tilde{R}$  of length  $2l$ .*

*Proof.* We will find strings  $u_0, \dots, u_l$  such that  $|u_i| = 2i$  and  $u_i \in \tilde{R}$ . Let  $u_0$  be the empty string. Certainly  $u_0 \in \tilde{R}$ .

To find  $u_i$  given  $u_{i-1}$  append first  $00$  to  $u_{i-1}$  and let  $u$  be the resulting string. As  $C(u_{i-1} \mid 2(i-1)) \geq 2(i-1)$  we have  $C(u_i \mid 2i) \geq 2i - c_3$  for some constant  $c_3$ . By Theorem 11 there is a string  $v$  in  $N^{c_3 c_2}(u)$  such that  $v \in \tilde{R}$ . Making at most  $5^{c_3 c_2}$  queries to the oracle  $\tilde{R}$  we find the first such  $v$  and let  $u_i = v$ .  $\square$

*Remark 2.* The same argument applies as well to every set of the form  $\{x \mid C(x \mid |x|) \geq f(|x|)\}$  where  $f(n) \leq n$  and  $f(n+1) \leq f(n) + O(\log n)$  for all  $n$ . In this case we search for  $v$  in  $N^{(c_3 + O(\log n))c_2}(u)$  in place of  $N^{c_3 c_2}(u)$ . As  $N^{(c_3 + O(\log n))c_2}(u)$  still has polynomial size the algorithm runs in polynomial time. Note that the algorithm need no other information about  $f$  than the constant hidden in  $O(\log n)$ .

*Remark 3.* The argument applies also to find random strings of odd lengths, but that requires more technical details. Given a string  $u$  of even length  $n = 2l$  with  $C(u|n) \geq n$  we need to find a string  $v$  of odd length  $n = 2l + 1$  with  $C(v|n) \geq n$ . To this end we can use Margulis' expander for the largest  $k$  such that  $k^2 \leq 2^{2l+1}$ . Obviously  $k^2 \geq 2^{2l}$  and we may identify strings of length  $2l + 1$  ending with  $0$  with the first  $2^{2l}$  nodes of the graph, and the other nodes with the first remaining strings of length  $2l + 1$ . Again we have  $C(u0 \mid 2l + 1) \geq 2l + 1 - c_3$  for a constant  $c_3$ . For large enough  $l$  the difference between  $2^{2l+1}$  and  $k^2$  is less than  $2^{2l+1}/(2c_1)$  where  $c_1$  is a constant such that the number of random strings of length  $2l + 1$  is at least  $2^{2l+1}/c_1$ . Therefore at least  $k^2/(2c_1)$  nodes in the graph are random and we can apply the arguments from the proof of Theorem 11 with  $2c_1$  in place of  $c_1$ .

**Corollary 13.**  $BPP \subset P^{\tilde{R}}$

*Proof.* Let  $M$  be a probabilistic machine recognizing a language  $A$ . Let  $n$  be the length of input to  $M$ . We can assume that the probability that  $M$  errs on at least one string of length  $n$  is at most  $2^{-n}$ . Let  $n^d$  be the length of random strings used by  $M$  on inputs of length  $n$ .

Here is the deterministic algorithm with oracle  $\tilde{R}$  to recognize  $A$ : Find a string  $r \in \tilde{R}$  of length  $n^d$  and run  $M$  on the input  $x$  using  $r$  as the sequence of random bits for  $M$  (we use the same string  $r$  for all inputs  $x$ ). Then output the result of  $M$ .

If for some string of length  $n$  the answer is incorrect then the string  $r$  falls into a set of cardinality  $2^{n^d - n}$  that is identified by  $n$  and  $M$  and hence  $C(r \mid n^d) \leq n^d - n + O(1) < n^d$  for  $n$  large enough, which is a contradiction. Thus our polynomial time algorithm with oracle  $\tilde{R}$  is correct for almost all inputs. Hardwiring the table of answers for small inputs we obtain a polynomial time algorithm with oracle  $\tilde{R}$  that recognizes  $A$  (on all inputs).  $\square$

Let us turn to the unconditional Kolmogorov complexity  $C(x)$ . Let  $R = \{x \mid C(x) \geq |x|\}$ . We will show that Theorem 12, the next two remarks and Corollary 13 generalize to  $R$ . As to Theorem 11, we can prove only a weaker its version:

**Theorem 14.** *There is a constant  $c_2$  such that for every node  $u$  in  $G$  with  $C(u) < n$  there is a node  $v \in N^{c_2 \log n + c_2}(u)$  with  $C(u) > C(v)$ .*

*Proof.* Essentially the same proof, as for Theorem 11 but this time we need to choose  $c_2$  so that  $(1 + \varepsilon_{c_1})^{c_2 \log n + c_2} \geq 2^{c+2 \log n}$ . In place of inequality  $C(u|n) \leq C(u|n) + 1 - c + 2 \log c + O(1)$  we have the inequality  $C(u) \leq C(u) + 1 - c - 2 \log n + 2 \log c + 2 \log l + O(1)$ . The term  $2 \log n$  is needed as this time we have to identify the length of  $u$ .  $\square$

However, to prove the analog of Theorem 12 we need only to increase Kolmogorov complexity of strings  $u$  with  $C(u) \geq |u| - O(1)$ . For that special case we have

**Theorem 15.** *For every constant  $c_3$  there is a constant  $c_4$  such that for every node  $u$  in  $G$  with  $n > C(u) \geq n - c_3$  there is a node  $v \in N^{c_4}(u)$  with  $C(u) > C(v)$ .*

*Proof.* Again the same proof but in place of inequality  $C(u|n) \leq C(u|n) + 1 - c + 2 \log c + O(1)$  we have the inequality  $C(u) \leq C(u) + 1 - c + 2 \log c + O(1)$ . This time we can find the length of  $u$  from the length  $C(u) + 1 - c$  of the index of  $u$  in  $A_{c_4}$  and from  $c$ , as  $C(u)$  and  $|u|$  are close to each other.  $\square$

Therefore Theorem 12, the next two remarks and Corollary 13 generalize to the unconditional Kolmogorov complexity.

## References

- [1] Ahlswede, Gács, Körner. Bounds on conditional probabilities with applications in multi-user communication. *Z. Wahrscheinlichkeitstheorie verw. Gebiete* 34 (1976) 157–177.
- [2] Allender, Buhrman, Koucký, van Melkbeek and Ronneberger. Power from Random Strings. 43rd IEEE Symposium on the Foundations of Computer Science (2002) 669–678.
- [3] L.H. Harper. Optimal numberings and isoperimetric problems on graphs. *J. Combinatorial Theory* 1 (1966) 385–393.
- [4] G.O.H. Katona. The Hamming-sphere has minimum boundary. *Studia Scientiarum Mathematicarum Hungarica* 10 (1975) 131–140.
- [5] M. Li, P.M.B. Vitányi. *An introduction to Kolmogorov complexity and its applications*. New York, Springer-Verlag, 1997.
- [6] G.A. Margulis. Explicit constructions of concentrators. *Explicit construction of concentrators. Probab. Info. Trans.*, 9 (1975), 325–332. (Translated into English from “Problemy peredachi informatsii” 9(2) (1973) 71–80.)
- [7] Rosen (ed.), *Handbook of Discrete Combinatorial Mathematics*, CRC Press, 2000.