Multisource Algorithmic Information Theory

Alexander Shen*

March 10, 2006

Abstract

Multisource information theory in Shannon setting is well known. In this article we try to develop its algorithmic information theory counterpart and use it as the general framework for many interesting questions about Kolmogorov complexity.

1 Introduction

Multisource information theory deals with information transmission in a network. Such a network includes information sources (one or many), the destinations (one or many) where information should be delivered, and channels that are used for transmission; some (or all) channels may have limited capacity. Classical Shannon approach considers sources as random variables and is well developed, see, e.g., [4, 14]. It tries to find conditions that make some information transmission request feasible.

Similar questions could (and should) be asked for algorithmic information theory.

Consider a directed graph whose edges are "channels" and nodes are "processors". Some nodes get outside information; this information should be processed (in nodes) and transmitted (via edges) into some other nodes.

More formally, an information transmission *request* consists of the following parts:

• A finite acyclic directed graph.

• A set of *input* nodes.

• An *input string* for each input node.

• A set of *output* nodes.

• A (desired) *output string* for each output node.

• An integer *capacity* for each edge (the value $+\infty$ that means unlimited capacity is also allowed).

We say that information request is c-feasible if one can find for each edge e a string M_e in such a way that the length of M_e does not exceed the capacity of edge e and

$$K(X|Y_1,\ldots,Y_k) \le c$$

*LIF CNRS, Marseille; Laboratoire Poncelet, CNRS, Institute for Information Transmission Problems, Moscow; alexander.shen@lif.univ-mrs.fr. Supported by RFBR grants 03-01-00475 and Scientific Schools grant 358.2003.1

for any node z, any outgoing string X (for z) and incoming strings $Y_1, \ldots, Y_k$ (for z), where

• K stands for (conditional) Kolmogorov complexity, i.e., the length of the shortest program that gets $Y_1, \ldots, Y_k$ as input and produces X as output, see the textbook [7] or tutorial [13].

• *outgoing* strings for node z are strings M_e for all outgoing edges e and the output string for z (if z is an output node);

• *incoming* strings for node z are strings M_e for all incoming edges e and the input string for z (if z is an input node).

The integer c measures the amount of new information that is allowed to appear "from nowhere"; we cannot let $c = 0$ since Kolmogorov complexity is defined up to an additive constant. The most natural choice is $c = O(\log n)$ for input and output strings of length at most n. With this choice, it does not matter which version of Kolmogorov complexity (plain, prefix=self-delimiting, etc.) we are using.

So in fact we should consider not an isolated request but a family of requests (usually for the same graph and input/output nodes) depending on parameter N; the size of the strings used in the request should be at most N (or polynomial in N), and the feasibility means that N-th request is $c \log N$-feasible for some c and for all N. (This is a standard setting for algorithmic information theory.)

Our goal is to show how many different results in classical information theory and Kolmogorov complexity could be naturally expressed in this language (in terms of feasibility of informational requests for some networks).

2 A trivial example

Consider a network that has two nodes and one edge (Fig. 1). (Let us agree that all edges are directed top-down, so the direction arrows are omitted). The top node is an input node and has input stgring A; the bottom node is an output node and has output string B. The channel has capacity k. This request is feasible (for small c) if and only if $K(B|A)$ is close to 0 and $K(B)$ does not exceed k significantly: information transmission is possible if and only if B does not have significant information that is not present in A (conditional complexity of B given A is small) and total amount of information in B does not exceed (significantly) the capacity of communication channel.

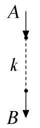

Figure 1: The simplest information transmission request

To express this evident idea formally, we (unfortunately) need a rather obscure statement:

Let A_n and B_n are sequences of strings and k_n be a sequence of integers. Assume that $|A_n|$, $|B_n|$ and k_n are bounded by a polynomial in n. Then the following two properties are equivalent:

(1) there exists a sequence of strings X_n such that $|X_n| \le k_n + O(\log n)$ and $K(X_n|A_n) = O(\log n)$, $K(B_n|X_n) = O(\log n)$;

(2) $K(B_n|A_n) = O(\log n)$ and $K(B_n) \le k_n + O(\log n)$.

This equivalence follows from two (rather trivial) remarks: first says that

$$K(B|A) \le K(B|X) + K(X|A) + O(\log K(A, B, X))$$

for all strings A, B, X (so (1) implies (2)); the second remark says that for any A, B and k there exists a string X such that

$$|X| \le K(B),\ K(X|A) \le K(B|A) + O(\log K(B)),\ K(B|X) = O(1)$$

(hint: let X be the shortest program for B) and implies that (1) follows from (2).

For the case $A = B$ the statement has clear intuitive meaning: a string A can be transmitted through a communication channel if and only if its complexity does not exceed the capacity of the channel.

3 A less trivial example

Consider the following information transmission request (which can be called "transmission of A when B is publicly known"), see Fig. 2. We need to encode A in the top node (using B if needed), transmit the encoding to the bottom node where decoding is performed (using B, too).

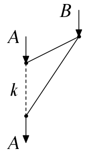

Figure 2: Transmission of A when B is known

It is easy to see that this request is feasible if and only if $K(B|A) \le k$. (This statement should be understood also in precise asymptotic way with sequence of requests and $O(\log n)$; we omit the exact formulation.) Indeed, the decoding algorithm knows B and k additional bits, so its output (A) has conditional complexity (with condition B) at most k. On the other hand, if $K(A|B) \le k$, let message X (for the limited capacity channel) be the shortest program that produces A with input B; both unlimited channels transmit B. Note that the conditional complexity of the shortest program that transforms B to A (with pair A, B as the condition) is logarithmic: knowing the length of such a program, we may try all programs of that length in parallel until some of them does the job.

4 A nontrivial example: Muchnik's theorem

Our next example is Muchnik's theorem that corresponds to Wolf – Slepyan theorem in Shannon imformation theory. It says that in the previous example one does not really need B for encoding (for decoding it is still needed, of course). The transmission request graph has the corresponding edge deleted (Fig. 3): Muchnik noted [9] that the condition $K(A|B) \leq k$ is still sufficient for the feasibility of this request. (It remains necessary for evident reasons.)

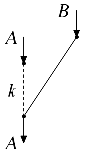

Figure 3: Wolf – Slepyan / Muchnik request

Here is the exact statement of Muchnik's theorem: *Let A and B be arbitrary strings of complexity at most n. Then there exists a string X of length $K(A|B) + O(\log n)$ such that $K(X|A) \leq O(\log n)$ and $K(A|X, B) \leq O(\log n)$.*

The proof of this theorem used expander-like graphs (similar methods were used also in [5] to get interesting results about resource-bounded Kolmogorov complexity). Roughly speaking, the message X sent through the restricted channel is a "fingerprint" (hash-value) of A that is a simple function of A; it happens that this hash value (plus small amount of additional information) could be sufficient to select A among all strings that have conditional complexity (with respect to B) at most k if a suitable (and small) family of hash functions is used.

5 Bidirectional encoding

Our next example is another well known result about Kolmogorov complexity [2] that says that *the length of the shortest program that transforms A into B and at the same time transforms B to A equals* $\max(K(A|B), K(B|A)) + O(\log n)$ *for any strings A, B of size at most n.*

It corresponds to the following transmission request (Fig. 4) and says that inequalities

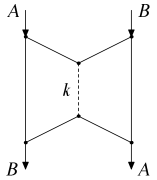

Figure 4: Bidirectional encoding

$K(A|B) \leq k$ and $K(B|A) \leq k$ are sufficient to make this request feasible (again with $O(\log n)$ terms that we omit). It is also clear that these inequalities are necessary since both strings that are sent along the lines in the bottom have complexity at most k and allow to get A from B and vice versa.

6 Conditional coding for two conditions

This example generalizes two previous ones. Consider the information transmission request shown in Fig. 5: Again the necessary condition for the feasibility of this request is simple:

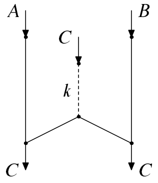

Figure 5: Coding C with respect to A and B

$K(C|A) \leq k$ and $K(C|B) \leq k$. As Muchnik has shown [9], this condition is also sufficient (with standard precautions about logarithmic terms). His result says that *for any three strings A, B, C of length at most n and for any k such that $K(C|A) \leq k$ and $K(C|B) \leq k$ there exists a string X of length k such that $K(X|C) = O(\log n)$, $K(C|A, X) = O(\log n)$ and $K(C|B, X) = O(\log n)$.*

Note that this result remains nontrivial even if we omit the condition $K(X|C) = O(\log n)$; no other (simpler) proof is known for this (potentially) weaker statement that corresponds to a (potentially) easier information transmission request (Fig. 6).

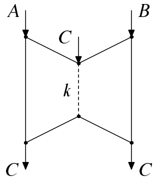

Figure 6: Coding C with either A or B known

7 Necessary condition for feasibility

All the conditions used in the previous examples could be obtained in an uniform way, by looking at the information flow through cuts in the network.

A *cut* is an arbitrary set I of nodes. We are interested in the information flow through I, i.e., the amount of information that comes to I from outside.

More formally, consider the total capacity of all edges whose starting point does not belong to I and endpoint belongs to I. If there is an unlimited capacity edge among them, the cut I gives no necessary condition for feasibility. Assume now that all capacities $u_1, \ldots, u_s$ of these edges are finite. Then the following necessary condition appears:

$$K(W_1, \ldots, W_m | V_1, \ldots, V_l) \leq u_1 + \ldots + u_s,$$

where $V_1, \ldots, V_l$ are input strings for all input vertices in I, and $W_1, \ldots, W_l$ are output strings for all output vertices in I. As usually, this inequality should be true with logarithmic precision, up to $O(\log n)$ terms (if all strings are of length at most n).

Indeed, the amount of information that enters I from outside (aside from input string) is limited by the total capacity of edges that enter I. (This is a standard Ford – Fulkerson type inequality.) Knowing s strings for edges that come into I and input strings, we can reconstruct all the strings inside I (there is no loops in the graph, so topological sorting is possible).

It is easy to see that all necessary conditions appearing in previous sections could be obtained in this way. For example, the conditions given in Section 5 are obtained through the following cut and the symmetric one (Fig. 7):

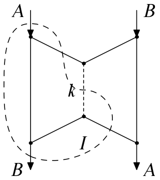

Figure 7: A cut for bidirectional encodings

A natural question arises whether the necessary conditions obtained in this way (for all possible cuts) are also sufficient for the feasibility of an information transmission request. In the situations considered they were; another case where they are sufficient is given in the next section. However, there are many cases where these conditions are not sufficient, as we shall see later.

8 Single-source networks

Consider the network with only one input string and several output strings identical to the input one. In other words, we have a broadcast request with a single source and several destinations. This problem is considered (for Shannon setting) in [1, 8]; the same ideas can be used for algorithmic version.

The main difficulty and the way to overcome it could be illustrated by the following example. Assume that we want to send a message A of size $2k$ to three destinations (Fig. 8).

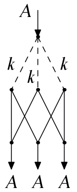

Figure 8: Broadcast to three destinations

Three first channels have limited capacity k; the remaining channels are unlimited. For each of three destinations (separately) the task is easy: we cut A into two parts of size k and send these two parts along two channels. But doing the same for all three destinations at the same time would require dividing A into "three halves" in such a way that any two are sufficient to reconstruct A.

It can be done using standard "linear secret sharing": three messages of length k are A_1, A_2 and $A_1 \oplus A_2$, where A_1 and A_2 are two halves of A and $\oplus$ stands for bitwise addition. Knowing any two of these three k-bit strings, we reconstruct the third one as bitwise sum of the known two and therefore know A.

A similar trick works for an arbitrary broadcast request. *Consider an information request that has only one input string of length n and several output strings identical to the input one, and some integers as edge capacities. Assume that necessary conditions are fulfilled for any cut I. More precisely, assume that for any set J of nodes that does not contain input vertex and contains at least one output vertex, the sum of capacities of edges that go into J is at least n. Then the request is $c\log n$-feasible for some c that does not depend on n and input string but may depend on the graph.*

The idea of the proof can be explained as follows. If there is only one output string, we can treat the bits as commodity in Ford – Fulkerson theorem. For each edge we know the indices of bits that should be sent through it; nodes do just the repacking of bits.

In a general case (of several destinations) we use linear coding. This means that all messages are considered as elements of vector space over a finite field. A message sent through some edge is a linear function of A. So each edge carries some vector space of possible messages (and its dimension is proportional to the capacity of the edge). A node performs a linear operation (the vector made of incoming strings is linearly transformed into the vector made of outcoming strings).

If transformation matrices are fixed for each node, we get and input – output linear mapping for each output node. Its matrix is a product of some parts of node transformation matrices. If input – output matrix is invertible for all output nodes, we are done. So we need to prove that it is possible to make all these matrices invertible. It is already known that we can do this for each matrix. Therefore the determinant as a polynomial function of matrix elements is not identically zero. Since the number of zeros of a polynomial is bounded, we can conclude that for some transformation matrices (and even for most of them) all the determinants are nonzero.

This argument requires technical clarification; in particular, the size of the field should be chosen carefully. (If it is too small, the zeros of the polynomials could cover the whole space; if it is too large, the overhead that appears because capacities are not multiples of the logarithm of the field size, becomes large.) But this clarification is not difficult.

Now we switch to the examples where the necessary conditions provided by information flow considerations are not sufficient. Several examples of this type were already considered in algorithmic information theory.

9 Common information

Consider the following information request (Fig. 9): two strings x and y are given. We should prepare three messages u, v and w of lengths (at most) α, β and γ such that x can be recontructed from u and v, and at the same time y can be reconstructed from u and w.

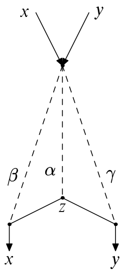

Figure 9: Common information

The motivation: u contains some "common information" that is present both in x and y, while v and w are remaining parts of x and y.

The requirements can be reformulated as conditions on u:

$$|u| \le \alpha,\ K(x|u) \le \beta,\ K(y|u) \le \gamma$$

(after u is chosen, v and w could be conditional descriptions of x and y with respect to u). We can also replace $|u|$ by $K(u)$ (by taking the shortest program for u instead of u itself).

The necessary conditions are

$$K(x) \le \alpha + \beta,\ K(y) \le \alpha + \gamma,\ K(x, y) \le \alpha + \beta + \gamma.$$

For example, let us consider the case when $K(x) = K(y) = 2n$ and $K(x, y) = 3n$. Informally, both x and y contain $2n$ information bits each, but are dependent, so the total amount of information is only $3n$ instead of $4n$. Let $\alpha = \beta = \gamma = n$, then all the flow conditions are satisfied. And the question can be reformulated as follows: is it possible to extract n bits of common information so that n additional bits are enough to specify x (or y)?

The answer is: it depends on x and y. It is possible, for example, if x and y are overlapping substrings of an incompressible string (of length $2n$ with n common bits). In this case u can be the intersection of x and y. On the other hands, there exists a triple of strings with the same complexities for which the request is not feasible.

There are several examples of this type (starting from [6], where this question is considered both from Shannon and algorithmic information theory).

More geometric example can be obtained as follows: consider a field of size 2^n and a two-dimensional affine plane over it. Let (x, y) be a random pair of concurrent line and point. With high probability they have complexities as stated above, but there is no common information. Moreover, one can prove that

$$K(u) = O(K(u|x) + K(u|y)),$$

so that if u has small conditional complexity with respect to x and y (and this follows from our requirements), then u is (unconditionally) simple.

This (together with other constructions of pairs of strings without common information) is explained in [3].

10 Program simplification

One can look for a simple information transmission request whether the necessary conditions (based on information flow) are not sufficient. It turns out that Muchnik theorem is quite close to the boundary: a bit more general request provides an example required.

Consider the following request (suggested by M. Vyugin, Fig. 10). The difference with Muchnik's theorem is that here the output string differs from the both input strings. The necessary condition here are $K(B|A) \leq k$ and $K(B|A, P) = 0$.

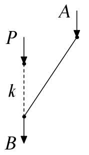

Figure 10: Program simplification

Informally the problem can be explained as follows. There exist a string B which can be obtained from string A if we know some additional information P (which can be considered as a program that transforms A to B). This program could be rather long. On the other hand, we know that conditional complexity of B when A is known does not exceed k; this means that there exists another (shorter) program that transforms B to A. Our goal is to find a "simplification" P' of program of B that has three properties: (1) it has no new information compared to B (i.e., $K(B'|B) = 0$); (2) it is still enough to transform A to B; (3) it has minimal possible length among programs that satisty (2).

Muchnik theorem says that is is possible (though not at all trivial) to find such a simplification if P equals B. But in general it is not true, as shown in [10].

11 Minimal sufficient statistic

Another request where our necessary conditions are not sufficient is motivated by the notion of minimal sufficient statistic.

Consider two following example. Imagine that we toss a biased coin (probability θ of heads is unknown, but all coin tosses are independent) and get a string $b_1 \ldots b_n$ of zeros and ones. Looking at this string, we try to guess θ. Intuitively it is clear that the only important information in $b_1 \ldots b_n$ is the number of 1's; no other information is relevant to θ. So the number of 1's is called a "minimal sufficient statistic" for θ. It contains all the information relevant to θ but nothing else.

Now we consider the information transmission request (Fig. 11) that formalizes this situation. Consider two strings A and B. The string B contains some infomation about A, as well as some other information. We try do delete irrelevant information from B and get a string B' that is simpler but still contains all information about A that was present in B.

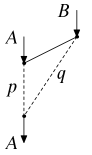

Figure 11: Minimal sufficient stgatistic

The last requirement can be formalized as follows: $K(A|B') \leq K(A|B)$ (if some information were lost when going from B to B', then conditional complexity would increase).

In terms of the graph: the simplified version B' is sent along the right edge and the information needed to restore A from B' is sent along the left edge. So our goal is to have $p = K(A|B)$ and $q = K(A) - K(A|B)$.

These values satisfy the information flow conditions, which are

$$K(A) \leq p + q, \quad K(A|B) \leq p$$

for this graph.

It is easy to see that this goal is achieved (and conditions are sufficient for the feasibility of the request) if A and B are overlapping parts of a random string; in this case B' is the common part of A and B and the remaining part of A is sent via the left channel.

However, in general the necessary conditions are not sufficient. For simplicity let us consider strings A and B that both have complexity $2n$ and the pair (A, B) has complexity $3n$. Then for a given p and q three cases are possible:

- the request is feasible for all strings A and B (from the class considered);
- the request is unfeasible for all strings A and B (from the class considered);
- none of the above (i.e., the answer depends on the choice of A and B),

and the (p, q)-plane is divided to three regions corresponding to this three cases. What are these regions? The analysis (which we omit now) gives the following answer (Fig. 12):

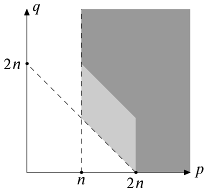

Figure 12: Three possibilities

the black part corresponds to the first case (always feasible), the white part corresponds to the second case (always infeasible), and the gray parallelogram inbetween corresponds to the third case (may be feasible or not depending on A and B). Note the white region is the region where information flow conditions are not fulfilled.

(Remark. For simplicity we omitted all the technical precautions about $O(\log n)$ precision that are needed to made the statement precise.)

12 Concluding remarks

One of the goals of multisource algorithmic information theory is to perform a similar analysis for an arbitrary graph and (for given complexities of input and output strings, as well as there combinations) divide the space of capacity parameters in three regions.

Our examples shows that even for simple graphs this task could be hard in both directions. Muchnik theorem shows that an elaborated combinatorial technique is needed to prove the feasibility of the request for all strings. The constructions of negative examples (in the last three sections) also involve combinatorial arguments that seem to be rather specific for the graph considered. (See [10] where three different methods to obtain negative results are explained.)

It would be interesting to find some more general criteria or, establish some formal connections between Shannon multisource information theory and algorithmic one (in the spirit of [11, 12]).

Acknowledments

This article is based on the discussions and results reported at the Kolmogorov seminar (Moscow). The author is grateful to all participants of the seminar for many useful comments. I am thankful to Uppsala University and Laboratoire d'Informatique Fondamentale de Marseille for hospitality and support.

References

[1] R. Ahlswede, N. Cai, S.-Y.R. Li, R.W. Yeung, Network information flow, *IEEE Trans. Inform. Theory*, v. 46, p. 1004–1016, July 2000.

[2] C. Bennett, P. Gács, M. Li, P. Vitanyi, W. Zurek, Information distance. *Proc. 25th ACM Symp. Theory of Comput.*, 1993, 21–30. Final version: *IEEE Trans. Inform. Theory*, IT-44:4 (1998), 1407–1423.

[3] An. Muchnik, A. Romashchenko, N. Vereshagin, A. Shen, Upper semi-lattice of binary strings with relation "x is simple conditional to y". DIMACS Tech. Report, 97-74 (December 1997). Revised version: Proceedings of 1999 Computational Complexity conference, Atlanta. Final version (with A. Chernov): *Theoretical Computer Science*, v. 271, no. 1–2, p. 69–95 (2002).

[4] I. Cziszar and J. Korner. *Information theory: Coding Theorems for Discrete Memoryless Systems.* Academic Press, New York. 1997. 2nd edition.

[5] H. Buhrman, L. Fortnow, S. Laplante, Resource-bounded Kolmogorov complexity revisited, *SIAM Journalin Computing*, v. 31, no. 3, p. 887–905 (2002)

[6] P. Gács, J. Körner, Common information is far less than mutual information, *Problems of Control and Information Theory*, v. 2 (2), p. 149–162

[7] M. Li, P. Vitanyi, *An Introduction to Kolmogorov Complexity and Its Application.* Springer, 1997. 2nd edition.

[8] S.-Y.R. Li, R.W. Yeung, N. Cai, Linear network coding, *IEEE Transactions on Information Theory*, v. 49, p. 371–381, Feb. 2003.

[9] An. Muchnik, Conditional complexity and codes, *Theoretical Computer Science*, v. 271, no. 1–2, p. 91–109 (2002).

[10] An. Muchnik, A. Shen, N. Vereshchagin, M. Vyugin, *Non-reducible descriptions for conditional Kolmogorov complexity.* Report TR04-054, Electronic Colloqium on Computational Complexity, ISSN 1433-8092.

[11] D. Hammer, A. Romashenko, A. Shen, N. Vereshchagin, Inequalities for Shannon entropies and Kolmogorov complexities. *Proceedings of CCC'97 Conference, Ulm.* Final version: Inequalities for Shannon entropy and Kolmogorov Complexity, *Journal of Computer and System Sciences*, v. 60, p. 442–464 (2000)

[12] A. Romashchenko, A. Shen, N. Vereshchagin, Combinatorial interpretation of Kolmogorov complexity, ECCC Report 7(26):2000; IEEE conference on Computational Complexity, published in *Theoretical Computer Science*, v. 271, no. 1–2, p. 111–123 (2002).

[13] A. Shen, *Algorithmic Information Theory and Kolmogorov Complexity.* Lecture notes of an introductory course. Uppsala University Technical Report 2000-034. Available online at `http://www.it.uu.se/research/publications/reports/2000-034`

[14] R. Yeung, *A First Course in Information Theory.* Springer, 2002.