On Algorithmic Strong Sufficient Statistics

Nikolay Vereshchagin[⋆]

Department of Mathematical Logic and the Theory of Algorithms,
Faculty of Mechanics and Mathematics, Lomonosov Moscow State University,
Leninskie gory 1, Moscow 119991, Russia
ver@mccme.ru

Abstract. The notion of a strong sufficient statistic was introduced in [8]. In this paper, we give a survey of nice properties of strong sufficient statistics and show that there are strings for which complexity of every strong sufficient statistic is much larger than complexity of its minimal sufficient statistic.

1 Introduction

Sufficient Statistics. Let x be a binary string. A finite set $A \subset \{0,1\}^*$ is called an *(algorithmic) sufficient statistic for* x if $x \in A$ and the sum of the Kolmogorov complexity[1] of A and the binary logarithm of the cardinality of A is close to the Kolmogorov complexity of x:

$$C(A) + \log|A| \approx C(x).$$

More specifically, we call A an ε-*sufficient* statistic for x if the left hand side exceeds the right hand side by at most ε. We do not require the inverse inequality, as it holds with precision $O(\log C(x))$ anyway.

For every x the singleton $\{x\}$ is an $O(1)$-sufficient statistic for x. The complexity of this statistic is about $C(x)$. If x is a random string of length n (that is, $C(x) \approx n$) then there is a $O(\log n)$-sufficient statistic for x of much lower complexity: the set of all strings of length n, whose complexity is about $\log n$, is a $O(\log n)$-statistic for x. We shall think further of ε as having the order $O(\log n)$ and call such values *negligible.*

Sufficient Statistics and Useful Information. Sufficient statistics for x are usually thought to capture all the "useful" information from x. The explanation is the following. Let A be a sufficient statistic for x. One can show that in this case both the *randomness deficiency* $\log|A| - C(x|A)$ of x in A and $C(A|x)$ are negligible.[2] Let z be the binary notation of the ordinal number of x in A

[⋆] The work was in part supported by the RFBR grant 12-01-00864 and the ANR grant ProjetANR-08-EMER-008.

[1] Kolmogorov complexity of finite subsets of $\{0,1\}^*$ is defined as follows. We fix any computable bijection $B \mapsto [B]$ from the family of all finite subsets of $\{0,1\}^*$ to the set of binary strings, called an *encoding.* Then we define $C(A)$ as the complexity $C([A])$ of the code $[A]$ of A.

[2] $C(x|A)$ and $C(A|x)$ are defined as $C(x|[A])$ and $C([A]|x)$, respectively, where $A \mapsto [A]$ is a fixed computable encoding of sets by strings (see the previous footnote).

P. Bonizzoni, V. Brattka, and B. Löwe (Eds.): CiE 2013, LNCS 7921, pp. 424–433, 2013.

(with respect to the lexicographical order on A). As $C(A|x)$ is negligible, both conditional complexities $C(x|A,z)$ and $C(A,z|x)$ are also negligible.[3] Speaking informally, the two part code (A,z) of x has the same information as x itself, and its second part z is a string of length $\log|A|$ that is random conditional to its first part A. (Indeed, $C(z|A)$ is up to an additive constant equal to $C(x|A)$, which is close to $\log|A|$.) This encourages us to qualify z as an accidental information (noise) in the pair (A,z), and hence in x. In other words, all useful information from x is captured by the set A.

Minimal Sufficient Statistics. If x has a sufficient statistic A of complexity i and log-cardinality j (so that $i+j \approx C(x)$) then for every $k \leqslant j$ it has a sufficient statistic B of complexity $i+k$ and log-cardinality $j-k$ (this statement is true with logarithmic precision; the complexity of B is actually $i+k+O(\log j)$). This was observed in [3,2,5]: the set B is obtained by partitioning A into subsets of size at most 2^{j-k} and considering the part containing x. Thus the most valuable sufficient statistic is the one that has smallest complexity and largest cardinality. Such statistics are informally called *minimal sufficient statistics, MSS,* for x. MSS for x are often considered as the models extracting all useful information from x and having no noise.

When trying to define the notion of an MSS formally, we face the following problem: for certain strings x a negligible increase in ε may cause a large decrease of the minimal complexity of ε-sufficient statistics for x. For such x it is not clear which value of ε to choose in the definition of ε-sufficient statistic and the notion of MSS cannot be defined in a meaningful way. In this paper we shall focus on strings for which this is not the case. To define more carefully what it means, consider for a given string x its *structure set* P_x. It consists of all pairs (i,j) of natural numbers for which x has an (i,j)-description, where an (i,j)-*description* is any set $A \ni x$ with $C(A) \leqslant i$ and $\log|A| \leqslant j$. The boundary of P_x is the graph of the function $h_x(i) = \min\{j \mid (i,j) \in P_x\}$, called the *structure function* of x. For every x the boundary of P_x lies above the *sufficiency line* (with logarithmic precision), which by definition consists of all pairs (i,j) with $i+j = C(x)$ (the dash line on Fig. 1). Sufficient statistics correspond to those pairs (i,j) from P_x that are close to the sufficiency line. We shall say (quite informally) that *a string x has an MSS*, if there is a natural i with $h_x(i) \approx C(x)-i$ and $h_x(i') \gg C(x)-i'$ for all i' which are "significantly less" than i. Notice that by observation from [3,2,5] mentioned above, in this case we also have $h_x(i') \approx C(x)-i'$ for all $i \leqslant i' \leqslant C(x)$ (with logarithmic precision).

Example 1. Let y be a string whose structure function h_y leaves the sufficiency line at the point $(C(y),0)$ (so that $\{y\}$ is essentially the only sufficient statistic

[3] $C(x|A,z)$ is defined as $C(x|[[A],z])$, where $(x,y) \mapsto [x,y]$ is a computable bijection between pairs of strings and strings; the notation $C(A,z|x)$ is understood in a similar way.

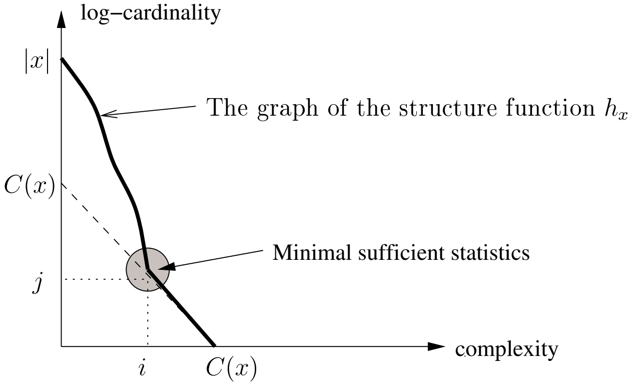

Fig. 1. The structure function h_x. The complexity and log-cardinality of minimal sufficient statistics for x are i and j, respectively.

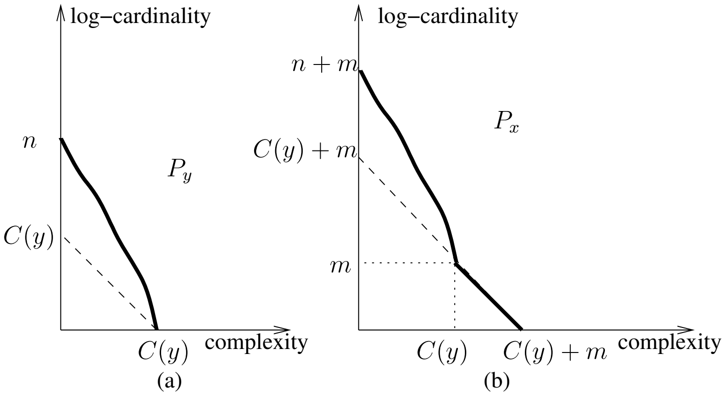

Fig. 2. The sets P_y and P_x

for y), see Fig. 2(a).[4] Let $x = [y, z]$, where z is a string of length m that is random conditional to y (that is, $C(z|y) \approx m$). Intuitively, x is obtained from y by adding m bits of noise and y captures all useful information from x. One can show ([8]) that the set P_x looks as drawn on Fig. 2(b).[5] Consider the set $A = \{[y, z'] \mid |z'| = m\}$ as a model for x. This model is a $(C(y) + O(\log m), m)$-description of x and hence an MSS for x. The information in A is almost the

[4] One can show ([7]) that for every decreasing function $h : \{0, 1, \ldots, k\} \to \mathbb{N}$ with $h(0) \leqslant n$ and $h(k) = 0$ there is a string y of length n for which the boundary of the set P_y is at the distance at most $O(\log n)$ from the graph of h.

[5] More specifically, the set P_x is $O(\varepsilon + \log(C(x) + m + j))$-close to the set

$$\{(i, j) \mid (j \leqslant m \Rightarrow i + j \geqslant C(x)) \ \wedge \ (j \geqslant m \Rightarrow (i, j - m) \in P_x)\}.$$

same as in y, which supports the viewpoint that an MSS for x captures all useful information in x.

Universal Sufficient Statistics. However, as discovered in [2,7], for every string x that has an MSS there is an MSS that can hardly be considered as a denoised version of x. To find such an MSS, fix an algorithm $\mathcal{A}$ that for input k enumerates (in some order) all strings of complexity at most k. Let N_k stand for the number of such strings and let $N_k = 2^{j_1} + 2^{j_2} + \cdots + 2^{j_s}$ be its binary expansion, where $j_1 > j_2 > \cdots > j_s$. Partition the list of strings enumerated by $\mathcal{A}(k)$ into 2^{j_1} first enumerated strings, 2^{j_2} strings enumerated after them and so on. Let S_{k,j_1}, S_{k,j_2}, …, S_{k,j_s} denote the obtained parts.

By definition $|S_{k,j}| = 2^j$ and it is not hard to show that $C(S_{k,j}) \leqslant k - j + O(\log k)$. Let $k = C(x)$. Consider the part where x goes, i.e., find j such that x belongs to $S_{k,j}$. In this case $S_{k,j}$ is a sufficient statistic for x, as $C(S_{k,j}) + \log |S_{k,j}| \leqslant (k - j + O \log k)) + j \approx C(x)$. One can show ([7]) that for every x which has an MSS, for some k close to $C(x)$ and for some j the set $S_{k,j}$ is an MSS for x. This fact is discouraging, because the family $S_{k,j}$ has only two parameters k, j. It implies that for all strings x from Example 1 there are k, j such that the set $S_{k,j}$ is also an MSS for x (where $k \approx C(x) \approx C(y) + m$ and $j \approx m$). Intuitively $S_{k,j}$ has no information about x, and on the other hand one can show that both conditional complexities $C(S_{k,j}|y)$ and $C(y|S_{k,j})$ are negligible. (See [7,8] for more details.)

Total Conditional Complexity. Thus we have to explain why it happens that the good model A from Example 1 has the same information as the bad model $S_{k,j}$. Also we would like to identify a property of MSS allowing to distinguish between good and bad models, such as the model A from Example 1 and the model $S_{k,j}$.

The first question is easy to answer: our definition of "having the same information" is too broad, we implicitly assumed that u and v have the same information, if both $C(u|v)$ and $C(v|u)$ are negligible. Under this assumption every string x has the same information as its shortest description x^*. In the context of separating the information into a useful one and an accidental one, such an assumption is certainly misleading. Indeed, the entire information in x^* (which is a random string) is noise, while x may have useful information. In algorithmic statistics, it is more helpful to think that u and v have the same information only if *total* conditional complexities $\mathrm{CT}(u|v)$ and $\mathrm{CT}(v|u)$ are negligible. The total conditional complexity $\mathrm{CT}(u|v)$ is defined as the minimal length of a total program p for u conditional to v: $\mathrm{CT}(u|v) = \min\{|p| \mid U(p, v) = u$ and $U(p, z)$ halts for all $z\}$ (here U is the universal Turing machine). The total conditional complexity can be much greater than the ordinary one [6].[6] If both $\mathrm{CT}(u|v)$ are

[6] In particular, in the full version of the paper we shall show that for all n there is string x of length n with $\mathrm{CT}(x|p) \geqslant n/3 - O(1)$ for every shortest description p of x. Moreover, this inequality holds for every description p of x of length at most $C(x) + n/3$. On the other hand, by a result of [1], for every x of length n there is a description p of x with $\mathrm{CT}(p|x) = O(\log n)$ and $|p| \leqslant C(x) + O(1)$.

$\mathrm{CT}(v|u)$ are negligible, then their structure sets P_u and P_v are close to each other and u, v have similar algorithmic-statistical properties. We shall call such strings *equivalent* in the sequel.

Strong Sufficient Statistics and Their Nice Properties. To distinguish between good and bad models, the paper [8] introduced a notion of a *strong* sufficient statistic. We call $A \ni x$ a *strong* statistic (or model) for x if $\mathrm{CT}(A|x)$ is negligible. (We do not assume that A is a sufficient statistic.) As we mentioned, the sufficiency requirement implies only that ordinary (not total) conditional complexity $C(A|x)$ is negligible. That is, not all sufficient statistics are strong (later we shall prove that). More specifically, we call A an *ε-strong* model for x if $\mathrm{CT}(A|x) \leqslant \varepsilon$ and we call A an *ε-good* model for x if A is ε-strong and ε-sufficient for x. We call (quite informally) a set A a *strong MSS for x* if A is an MSS for x and A is a strong model for x.

It easy to see that A is a strong model for x iff both total complexities $\mathrm{CT}(x|A, z)$, $\mathrm{CT}(A, z|x)$ are negligible, where z is the ordinal number of x in A. Indeed, given the pair (A, z) we can find x by means of a short total program (even if A is not strong). Conversely, if A is a strong statistic for x, then from x we can compute A by means of a short total program and then compute the ordinal number of x in A.

Strong sufficient statistics have the following nice properties.

(a) The model A from Example 1 is a strong MSS for x Indeed, given x we can find A by a constant length total program that maps $[y, z]$ to the set $\{[y, z'] \mid |z'| = |z|\}$. That is, x has a strong MSS if and only if x is equivalent to a string of the form specified in Example 1.

(b) Strong MSS are unique in the following sense: if both A, B are strong MSS for x, then $\mathrm{CT}(A|B) \approx \mathrm{CT}(B|A) \approx 0$ [8, Theorem 6]. We state here the result in a highly informal way, for the precise statement see [8].

(c) Good statistics satisfy the observation from [3,2,5]: If x has a good statistic A of complexity i and log-cardinality j, then for every $k \leqslant j$ it has a good statistic B of complexity $i + k$ and log-cardinality $j - k$ (with logarithmic precision): again, the set B is obtained by partitioning A into subsets of size at most 2^{j-k} and considering the part containing x.

Our Result. Recall that one of the goals of introducing the notion of a strong MSS is to separate MSS from Example 1 from MSS of the form $S_{k,j}$. We conjecture that this is true: there are strings x that have ε-strong MSS but have no ε-strong MSS of the form $S_{k,j}$ for some $\varepsilon = \Omega(|x|)$. In this paper we answer another question left open in [8]: is it true that every string that has an MSS has also a strong MSS? We show that this is not the case: there are strings that have MSS but all their strong sufficient statistics have much larger complexity than that of their MSS.

2 Results

Our results establish the existence of strings x that have an MSS but all their strong sufficient statistics have much larger complexity than that of their MSS.

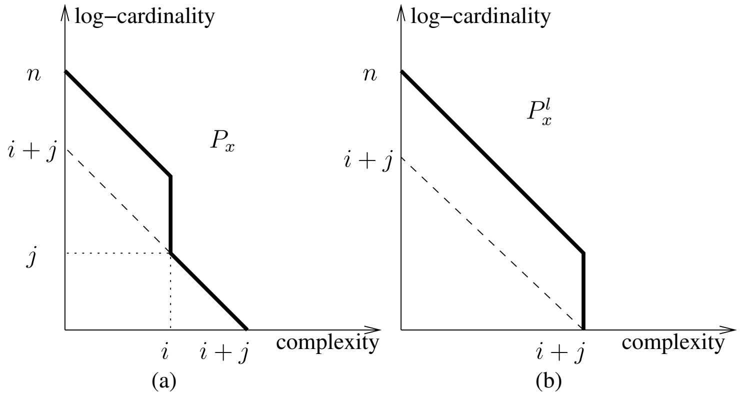

Fig. 3. The sets P_x and P_x^l

Recall that P_x denotes the set of all pairs (i, j) such that x has an (i, j)-description, that is, x belongs to a set A with $C(A) \leqslant i$ and $\log |A| \leqslant j$. Define a similar set P_x^l for l-strong models. By definition P_x^l consists of all pairs (i, j) such that x has an l-strong i, j-description:

$$P_x^l = \{(i, j) \mid (\exists A \ni x)\ C(A) \leqslant i,\ \log |A| \leqslant j,\ \mathrm{CT}(A|x) \leqslant l\}.$$

Our main result, Theorem 1, gives an example of a string x when the set of all explanations and good explanations differ in a maximal possible way: the sets P_x and P_x^l are shown on Fig. 3. The complexity of every l-strong sufficient statistic for x (for some non-negligible l) is at least (about) j bits more than that of its MSS, which is about i. This is the best separation possible, as the singleton $\{x\}$ is an $O(1)$-strong sufficient statistic of complexity about $C(x)$ for every x, and in our case $C(x)$ is about j bits more than i.

Theorem 1. *Assume that integer numbers i, j satisfy the inequality $i+j \leqslant n-4$. Then there is a string x of length n and complexity $i+j+O(\log n)$ such that (a) $(i+O(\log n), j) \in P_x$, (b) $(i, n-i-4) \notin P_x$ and (c) $(i+j, n-i-j-4) \notin P_x^i$.*

Item (a) of Theorem 1 is responsible for the right slanted segment of the boundary of P_x and item (b) is responsible for the left slanted segment of the boundary of P_x. Item (c) is responsible for the graph of P_x^l for any $O(\log n) \leqslant l \leqslant i$.

Let (say) in Theorem 1 $i = j = n/3$. Then the string x existing by the theorem has an MSS of complexity $n/3$ while all $n/3$-strong $n/3$-sufficient statistics for x have complexity at least $2n/3$.

Theorem 1 does not say anything about how rare are such strings x. Such strings are rare, as for majority of strings x of length n the set $\{0, 1\}^n$ is a strong MSS for x. A more meaningful question is whether such strings might appear with high probability in a statistical experiment. More specifically, assume that we sample a string x in a given set $A \subset \{0, 1\}^n$, where all elements are equiprobable. Might it happen that with high probability (say with probability 99%) x has an MSS but has no strong MSS? An affirmative answer to this question is given in the following

Theorem 2. *Assume that integer i, j, k satisfy the inequalities*

$$i + j \leqslant n - 4, \quad k \leqslant j.$$

Then there is set $A \subset \{0,1\}^n$ of cardinality 2^j and complexity at most $i + O(k + \log n)$ such that all but 2^{j-k} its elements x have complexity $i + j + O(k + \log n)$ and have neither $i, (n - i - 4)$-descriptions nor i-strong $(i + j), (n - i - j - 4)$-descriptions.

If $k = \log n$, say, then the set A is an MSS for a majority of its elements. Indeed, the structure set of all but $|A|/n$ elements from A has the shape shown on Fig. 3(a). On the other hand, for those elements the set P^l_x (for any $O(\log n) \leqslant l \leqslant i$) has the shape shown on Fig. 3(b).

Remark 1. Theorem 2 brings up the following questions. Imagine that somebody suggests a set A as a “statistical explanation” for the data x (that belongs to A). What properties of A are required to make this explanation reasonable? For example, do we want that A is a sufficient statistics for most of its elements? Do we want A to be simple (in the sense of normal conditional complexity or the total one) conditional to *every* its element? We think that defining the notion of a “reasonable explanation” one should be most restrictive, as far as every model as in Example 1 satisfies the restrictions. More specifically, we would call an MSS A a “reasonable explanation” for x if there is a short total program p that maps *every* $x' \in A$ to $[A]$. That is, the total conditional complexity of $[A]$ given any element of A is low in a uniform way. (This implies that A is a sufficient statistics for most of its elements.) This requirement is not that strong as one could think. Indeed, assume that A is a strong MSS for x and p a short total program with $U(p, x) = [A]$. Then the model $A' = \{x' \in A \mid U(p, x') = [A]\}$ is an MSS for x that is a “reasonable explanation” in this sense. Indeed, here is a total program of length about $|p|$ that transforms any $x' \in A'$ to $[A']$: given x' apply p to x' to find A and return the code of the set consisting of all $x'' \in A$ with $U(p, x'') = [A]$.

Proofs of Theorems 1 and 2. We start with the following observation.

Lemma 1. *Assume that A is an i-strong statistic for a string x of length n. Let $y = [A]$ be the code of A. Then y has an $(i + O(\log n), n)$-description.*

Proof. Let p be a string of length at most i such that $U(p, x)$ is defined for all strings x of length n. Consider the set $\{U(p, x) \mid x \in \{0,1\}^n\}$. Its cardinality is at most 2^n and complexity at most $i + O(\log n)$. If $\mathrm{CT}(y|x) \leqslant i$ for some $x \in \{0,1\}^n$ then there is p such that y belongs to such a set and hence y has a $(i + O(\log n), n)$-description.

By Lemma 1, to prove Theorem 1 it suffices to find a set $A \subset \{0,1\}^n$ with
(a) $C(A) \leqslant i + O(\log n)$, $\log |A| \leqslant j$
which is not covered by sets from the following three families:
(b) the family $\mathcal{B}$ consisting of all sets $B \subset \{0,1\}^*$ with with $C(B) \leqslant i$, $\log |B| \leqslant n - i - 4$,
(c) the family $\mathcal{C}$ consisting of all sets M with $C(M) \leqslant i+j$, $\log |M| \leqslant n-i-j-4$ whose code $[M]$ has a $(i + O(\log n), n)$-description, and
(d) the family $\mathcal{D}$ consisting of all singletons sets $\{x\}$ where $C(x) < i + j$.
As x we can take any non-covered string in A. Notice that item (a) implies that the complexity of x is at most $i + j + O(\log n)$, and item (d) implies that it is at least $i + j$.

A direct counting reveals that the family $\mathcal{B} \cup \mathcal{C} \cup \mathcal{D}$ covers at most

$$2^{i+1}2^{n-i-4} + 2^{i+j+1}2^{n-i-j-4} + 2^{i+j} \leqslant 2^{n-3} + 2^{n-3} + 2^{n-4} < 2^{n-1}$$

strings and hence at least half of all n-bitstrings are non-covered. However we cannot let A be any 2^j-element non-covered set of n-bitstrings, as in that case $C(A)$ could be large.

We first show how to find A, as in (a), that is not covered by $\mathcal{B} \cup \mathcal{D}$ (but may be covered by $\mathcal{C}$). This is done using the method of [7]. To construct A notice that both the families $\mathcal{B}$ and $\mathcal{D}$ can be enumerated given i, j, n by running the universal machine U in parallel on all inputs. We start such an enumeration and construct A "in several attempts". During the construction we maintain the list of all strings covered by sets from $\mathcal{B} \cup \mathcal{D}$ enumerated so far. Such strings are called *marked*. Initially, no strings are marked and A contains the lexicographic first 2^j strings of length n. Each time a new set $B \in \mathcal{B}$ appears, all its elements receive a b-mark and we replace A by any set consisting of 2^j yet non-marked n-bitstrings. Each time a new set $\{x\}$ in D appears, the string x receives a d-mark, but we do not immediately replace A. We do that only when all strings in A receive a d-mark, replacing it by any set consisting of 2^j yet non-marked n-bitstrings. The above counting shows that such replacements are always possible.

The last version of A (i.e., the version obtained after the last set in $\mathcal{B} \cup \mathcal{D}$ have appeared) is the sought set. Indeed, by construction $|A| = 2^j$ and A is not covered by sets in $\mathcal{B} \cup \mathcal{D}$. It remains to verify that $C(A) \leqslant i + O(\log n)$. This follows from the fact that A is replaced at most $O(2^i)$ times, and hence can be identified by the number of its replacements and i, j, n (we run the above construction of A and wait until the given number of replacements are made).

Why is A replaced at most $O(2^i)$ times? The number of replacements caused by appearance of a new set $B \in \mathcal{B}$ is at most 2^{i+1}. The number of strings with a d-mark is at most 2^{i+j} and hence A can be replaced at most $2^{i+j}/2^j = 2^i$ times due to receiving d-marks.

Now we have to take into account strings covered by sets from the family $\mathcal{C}$. We cannot modify the above arguments just by putting a c-mark on all strings from each set C enumerated into $\mathcal{C}$. Indeed, up to 2^{n-4} strings may receive a c-mark, and hence A might be replaced up to 2^{n-j-4} times due to c-marks.

We change the construction of A as follows. First we represent $\mathcal{C}$ as an intersection of two families, $\mathcal{C}'$ and $\mathcal{C}''$. The first family $\mathcal{C}'$ consists of all sets M with

$C(M) \leqslant i + j$ and the second family $\mathcal{C}''$ of all sets C with $\log |C| \leqslant n - i - j - 4$ whose code $[C]$ has a $(i + O(\log n), n)$-description. The first family is small (less than 2^{i+j+1} sets) and the second family has only small sets (at most $2^{n-i-j-4}$-element sets) and is not very large ($|\mathcal{C}''| = 2^{O(n)}$). Both families can be enumerated given i, j, n and, moreover, the sets from $\mathcal{C}''$ appear in the enumeration in at most $2^{i+O(\log n)}$ portions. Due to this property of $\mathcal{C}''$ we can update A each time a new portion of sets in $\mathcal{C}''$ appears—this will increase the number of replacements of A by $2^{i+O(\log n)}$, which is OK.

The crucial change in construction is the following: each time A is replaced, its new version is not just any set of 2^j non-marked n-bitstrings but a carefully chosen such set: we choose any such set that has at most $O(n)$ common strings with *every* set from the part of $\mathcal{C}''$ enumerated so far. (We shall show later that such a set always exists.)

Why does this solve the problem? There are two types of replacements of A: those after enumerating a new set in $\mathcal{B}$ or a new bunch of sets in $\mathcal{C}''$ and those after all elements in A have received c- or d-marks. The number of replacement of the first type is at most $2^{i+O(\log n)}$. Replacements of the second type are caused by enumerating new singleton sets in $\mathcal{D}$ and enumerating new sets C in $\mathcal{C}'$ which were enumerated into $\mathcal{C}''$ on earlier steps. Due to the careful choice of A, when each such set C appears in the enumeration of $\mathcal{C}'$ it can mark only $O(n)$ strings in the current version of A. The total number of sets in $\mathcal{C}'$ is at most 2^{i+j+1}. Therefore the total number of events "a string in the current version of A receives a c-mark" is at most $O(n2^{i+j})$. The total number of d-marks is at most 2^{i+j}. Hence the number of replacements of the second type is at most

$$(O(n2^{i+j}) + 2^{i+j})/2^j = O(n2^i).$$

Thus it remains to show that we indeed can always choose A, as described above. This will follow from a lemma that says that in a large universe one can always choose a large set that has a small intersection with every set from a given small family of small sets.

Lemma 2. *Assume that a finite family $\mathcal{C}$ of subsets of a finite universe U is given and each set in $\mathcal{C}$ has at most s elements. If*

$$|\mathcal{C}| \binom{N}{t+1} \left(\frac{s}{|U| - t}\right)^{t+1} < 1$$

then there is an N-element set $A \subset U$ that has at most t common elements with each set in $\mathcal{C}$.

Proof. To prove the lemma we use probabilistic method. The first element a_1 of A is chosen at random among all elements in U with uniform distribution, the second element a_2 is chosen with uniform distribution among the remaining elements and so forth.

We have to show that the statement of the theorem holds with positive probability. To this end note that for every fixed C in $\mathcal{C}$ and for every fixed set of indexes $\{i_1, \dots, i_{t+1}\} \subset \{1, 2, \dots, N\}$ the probability that *all* $a_{i_1}, \dots, a_{i_{t+1}}$ fall in C is at most $\left(\frac{s}{|U|-t}\right)^{t+1}$. The number of sets of indexes as above is $\binom{N}{t+1}$. By union bound the probability that a random set A does not satisfy the lemma is upper bounded by the left hand side of the displayed inequality.

We apply the lemma for U consisting of all non-marked n-bitstrings, $N = 2^j$ and $\mathcal{C}$ consisting of all sets in $\mathcal{C}''$ appeared so far. Thus we need to show that for some $t = O(n)$ it holds

$$2^{O(n)}\binom{2^j}{t+1}\left(\frac{2^{n-i-j-4}}{2^{n-1}-t}\right)^{t+1} < 1,$$

which easily follows from the inequality $\binom{2^j}{t+1} \leqslant 2^{j(t+1)}$. Theorem 1 is proved.

Theorem 2 is proved similarly to Theorem 1. The only difference that we change A each time when at least 2^{j-k} strings in A receive c- or d-marks. As the result, the number of changes of A will increase 2^k times and the complexity of A will increase by k.

Acknowledgments. The author is sincerely grateful to anonymous referees for helpful comments, especially for the questions discussed in Remark 1.

References

1. Bauwens, B., Makhlin, A., Vereshchagin, N., Zimand, M.: Short lists with short programs in short time. ECCC report TR13-007, http://eccc.hpi-web.de/report/2013/007/
2. Gács, P., Tromp, J., Vitányi, P.M.B.: Algorithmic statistics. IEEE Trans. Inform. Th. 47(6), 2443–2463 (2001)
3. Kolmogorov, A.N.: Talk at the Information Theory Symposium in Tallinn, Estonia (1974)
4. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and its Applications, 2nd edn. Springer, New York (1997)
5. Kh, A.: Shen, Discussion on Kolmogorov complexity and statistical analysis. The Computer Journal 42(4), 340–342 (1999)
6. Shen, A.: Game Arguments in Computability Theory and Algorithmic Information Theory. In: Cooper, S.B., Dawar, A., Löwe, B. (eds.) CiE 2012. LNCS, vol. 7318, pp. 655–666. Springer, Heidelberg (2012)
7. Vereshchagin, N.K., Vitányi, P.M.B.: Kolmogorov's structure functions and model selection. IEEE Trans. Information Theory 50(12), 3265–3290 (2004)
8. Vereshchagin, N.: Algorithmic Minimal Sufficient Statistic Revisited. In: Ambos-Spies, K., Löwe, B., Merkle, W. (eds.) CiE 2009. LNCS, vol. 5635, pp. 478–487. Springer, Heidelberg (2009)