The Power and Perils of MDL

Pieter Adriaans
Department of Computer Science, University of Amsterdam
Plantage Muidergracht 24, 1018TV Amsterdam, The Netherlands
Pieter.Adriaans@ps.net

Paul Vitányi
CWI, Kruislaan 413
1098 SJ Amsterdam, The Netherlands
Paul.Vitanyi@cwi.nl

Abstract— **We point out a potential weakness in the application of the celebrated Minimum Description Length (MDL) principle for model selection. Specifically, it is shown that (although the index of the model class which actually minimizes a two-part code has many desirable properties) a model which has a shorter two-part code-length than another is not necessarily better (unless of course it achieves the global minimum). This is illustrated by an application to infer a grammar (DFA) from positive examples. We also analyze computability issues, and robustness under recoding of the data. Generally, the classical approach is inadequate to express the goodness-of-fit of individual models for individual data sets. In practice however, this is precisely what we are interested in: both to express the goodness of a procedure and where and how it can fail. To achieve this practical goal, we paradoxically have to use the, supposedly impractical, vehicle of Kolmogorov complexity.**

I. Introduction

In learning algorithms using the two-part minimal description length principle (MDL), based on the original work of J. Rissanen, we observe that although it may be true that the maximal compression yields the best solution, it may still not be true that every incremental compression brings us closer to the solution. Moreover, in the case of most MDL problems there is a complicating issue in the fact that the maximal compression cannot be computed. In many practical applications of MDL, it is too hard to find the global minimizer over all model classes, the problem being NP-hard or even non-computable. To obtain the shortest code, the natural way is to approximate it by a process of finding ever shorter candidate two-part codes. Since we start with a finite two-part code, and with every new candidate two-part code we decrease the code length, eventually we must achieve the shortest two-part code. Unfortunately, there are two problems: (i) the computation to find the next shorter two-part code may be very long, and we may not know how long; and (ii) we may not know when we have reached the shortest two-part code: with each candidate two-part code there is the possibility that further computation may yield a still shorter one. But because of item (i) we cannot a priori bound the length of that computation.

A. A Common Misconception

Therefore, in practice, we look for ever shorter two-part codes, and if our available time runs out we make do with the last candidate we found. The underlying assumption is that a shorter two-part code for the data yields a better model than a longer two-part code. It is the purpose of this paper to debunk this myth: While a sequence of ever shorter two-part codes for the data converges in a finite number of steps to the best model, it is not the case that this convergence is necessarily monotonic. In fact, in the sequence of candidate two-part codes converging to the shortest, it is possible that the models involved oscillate from being good to bad, only to converge at the (unknown) very end to the best model. Convergence is only monotone if the model-codes in the successive two-part codes are always the shortest (most compressed) codes for the models involved. But this property cannot be guarantied by any effective method.

It is very difficult, if not impossible, to formalize the goodness-of-fit of an individual model for individual data in the classic statistics setting, which is probabilistic. Therefore, it is impossible to express the practically important issue above in those terms. Fortunately, new developments in the theory of Kolmogorov complexity make it possible to rigorously analyse the questions involved, and exhibit the phenomena, in certain model classes, as in [3]. Here we elaborate on that treatment, make it more accessible and probe its implications for MDL. This then is illustrative for what happens in practical situations. Because of space limitations we omit definitions and details on Kolmogorov complexity here; we refer to the textbook [2].

B. What Does It Mean That A Model Fits Given Data

Denote the *complexity of the finite set* A by $K(A)$—the length (number of bits) of the shortest binary program p from which the reference universal prefix machine U computes a listing of the elements of S and then halts. That is, if $A = \{x_1, \ldots, x_d\}$, then $U(p) = \langle x_1, \langle x_2, \ldots, \langle x_{d-1}, x_d \rangle \ldots \rangle\rangle$. The shortest program p, or, if there is more than one such shortest program, then the first one that halts in a standard dovetailed running of all programs, is denoted by A^*. Consider a data sample D and a model M, such that $D \subseteq M \subseteq \{0,1\}^{\leq n}$. Denote the cardinalities by lower case letters:

$$d = |D|, m = |M|.$$

The *conditional complexity* $K(D \mid M, d))$ of D given M and d is the length (number of bits) in the shortest binary program p from which the reference universal prefix machine U from input M (given as a list of elements) and the number of elements d, outputs D as a list of elements and halts. We elaborate on the approach of [3]. If $D \subseteq M \subseteq \{0,1\}^{\leq n}$ we have

$$K(D \mid M, d)) \leq \log \binom{m}{d} + O(1). \tag{1}$$

1-4244-1429-6/07/$25.00 ©2007 IEEE

Indeed, consider the selfdelimiting code of D, given M and the number d of elements in D followed by the $\lceil \log \binom{m}{d} \rceil$ bit long index of D in the lexicographical ordering of the number of ways to choose d elements from M. This code is called the *data-to-model code*. Its length quantifies the maximal "typicality," or "randomness," any data sample of $|D|$ elements can have with respect to model M.

DEFINITION 1: The lack of typicality of D with respect to M is measured by the amount by which $K(D \mid M, d)$ falls short of the length of the data-to-model code. The *randomness deficiency* of a data sample D, of known cardinality $d = |D|$, in model M, is defined by

$$\delta(D \mid M, d) = \log \binom{m}{d} - K(D \mid M, d), \tag{2}$$

for $D \subseteq M$, and ∞ otherwise.

If the randomness deficiency is close to 0, then there are no simple special properties that single D out from the majority of data samples to be drawn from M. This is not just terminology: If $\delta(D \mid M, d)$ is small enough, then D satisfies *all* properties of low Kolmogorov complexity that hold for the majority of subsets of M. To be precise: A *property* P represented by M is a subset of M, and we say that D satisfies property P if $D \subseteq P$.

LEMMA 1: Let d, m, n be natural numbers, and let $D \subseteq M \subseteq \{0,1\}^{\leq n}$, $|D| = d, |M| = m$, and let δ be a simple function of the natural numbers to the reals, like $\log$ or $\surd$.

(i) If P is a property satisfied by all $D \subseteq M$ with $\delta(D \mid M, d) \leq \delta(n)$, then P holds for a fraction of at least $1 - 1/2^{\delta(n)}$ of the subsets of cardinality d of M.

(ii) Let n and M be fixed, and let P be any property that holds for a fraction of at least $1 - 1/2^{\delta(n)}$ of the subsets of cardinality d of M. There is a constant c, such that every such P holds simultaneously for every $D \subseteq M$ with $|D| = d$ and $\delta(D \mid M, d) \leq \delta(n) - K(P \mid M) - c$.

We omit the proof. The *minimal randomness deficiency* function (with known cardinality of the data sample) is

$$\beta_D(\alpha) = \min_M \{\delta(D \mid M, d) : M \supseteq D,\ K(M|d) \leq \alpha\}, \tag{3}$$

where we set $\min \emptyset = \infty$. The smaller $\delta(D \mid M, d)$ is, the more D can be considered as a *typical* data sample from M. This means that a set M for which D incurs minimal randomness deficiency, in the model class of contemplated sets of given maximal Kolmogorov complexity, is a "best fitting" model for D in that model class—a most likely explanation, and $\beta_D(\alpha)$ can be viewed as a *constrained best fit estimator*.

II. MINIMUM DESCRIPTION LENGTH ESTIMATOR

The length of the minimal two-part code for D, with known cardinality $d = |D|$ consisting of the model cost $K(M)$ ($|M| = m$) and the length of the index of D in the enumeration of choices of d elements out of m, in the model class of sets M of given maximal Kolmogorov complexity α, the complexity of M upper bounded by α, is given by the *MDL* function or *constrained MDL estimator*:

$$\lambda_D(\alpha) = \min_M \{\Lambda(M) : M \supseteq D,\ K(M|d) \leq \alpha\}, \tag{4}$$

where $\Lambda(M) = K(M|d) + \log \binom{m}{d} \geq K(D|d) - O(1)$ is the total length of two-part code of D with help of model M and the cardinality d. This function $\lambda_D(\alpha)$ is the celebrated two-part Minimum Description Length code length as a function of α, with the model class restricted to models of code length at most α. Indeed, consider the following *two-part code* for D when we know its cardinality d: the first part is a shortest self-delimiting program p for M and the second part is $\lceil \log \binom{m}{d} \rceil$ bit long index of D in the lexicographical ordering of all choices of d elements from M. Since M, d determines $\log \binom{m}{d}$ this code is self-delimiting and we obtain the two-part code, where the constant $O(1)$ is the length of the program to reconstruct D from its two-part code and known cardinality d. For those α's that have $\lambda_D(\alpha) = K(D|d) + O(1)$, the associated model M (witness for $\lambda_D(\alpha)$) for D, or the description of $M|d$ of $\leq \alpha$ bits, is called a *sufficient statistic*. We omit the proof of the following lemma.

LEMMA 2: If M is a sufficient statistic for D, then the randomness deficiency of D in M is $O(1)$, D is a typical data sample for M, and M is a model of best fit for D.

III. POWER AND PITFALLS

The previous analysis of MDL allows us to justify its application and to identify problems in its application that may not be apparent at first glance.

A. Computability

How difficult is it to compute the functions λ_D, β_D, and the minimal sufficient statistic? To express the properties appropriately we require the notion of functions that are not computable, but can be approximated monotonically by a computable function either from above, called *upper semi-computable*, or from below, called *lower semi-computable*. When it is both, then it is *computable*. We omit the formal definitions.

- The function $\lambda_D(\alpha)$ is upper semi-computable but not computable up to any reasonable precision.
- Moreover, there is no algorithm that given D^* and α finds $\lambda_D(\alpha)$.
- The function $\beta_D(\alpha)$ is not upper- or lower semi-computable, not even to any reasonable precision, but we can compute it given an oracle for the halting problem.
- There is no algorithm that given D and $K(D)$ finds a minimal sufficient statistic for D up to any reasonable precision.

B. Invariance under Recoding of Data

In what sense are the functions invariant under recoding of the data? If the functions β_D, λ_D give us the stochastic properties of the data D, then we would not expect those properties to change under recoding of the data into another format. Yet, if we recode the elements of $D = \{x_1, \ldots, x_d\}$ by a mapping c of $\{0,1\}^{\leq n}$ to obtain $c(D) = \{c(x_1), \ldots, c(x_d)\}$ such that $c(x_i) = x_i^*$ with $|x_i^*| = K(x_i)$ ($1 \leq i \leq d$), then we are in trouble. We can choose the x_i's such that $K(c(D) \mid$

$\{0,1\}^{\leq \mu}, d) \approx \binom{2^{\mu+1}-1}{d}$, with $\mu = \max\{K(c(x_i)) : 1 \leq i \leq d\}$. Then,

$$\delta(c(D) \mid \{0,1\}^{\leq \mu}, d) \approx 0.$$

That is, $c(D)$ is a typical d-element subset of $\{0,1\}^{\leq \mu}$, and the latter in turn is the best fitting model for $c(D)$. Therefore $\lambda_{c(D)}(\alpha)$ drops to the Kolmogorov complexity $K(c(D))$ already for some $\alpha \leq K(\mu) + O(1) = O(\log n)$, since $\lambda_{c(D)}(K(\mu) + O(1)) = K(\{0,1\}^{\leq \mu} \mid d) + \log \binom{2^{\mu+1}-1}{d} \approx K(c(D))$, so almost immediately (and it stays within logarithmic distance of that line henceforth). That is, $\lambda_{c(D)}(\alpha) = K(x_1^*, \ldots, x_d^*)$ for every α, up to logarithmic additive terms in argument and value, irrespective of the (possibly quite different) shape of λ_D. It is clear that a coding c that achieves this is not a recursive function, and neither is the inverse. However, it is not the non-recursiveness alone, but also the necessary partiality of the inverse function (not all data samples contain data of maximal Kolmogorov complexity) that causes the collapse of the structure function. Nonetheless, the structure function is invariant under "proper" recoding of the data, as follows:

THEOREM 1: Let f be a recursive permutation of the set of finite binary strings in $\{0,1\}^n$ (one-one, total, and onto), and extend f to subsets $D \subseteq \{0,1\}^n$. Then, $\lambda_{f(D)}$ is "close" to λ_D in the sense that the graph of $\lambda_{f(D)}$ is situated within a strip of width $K(f) + O(1)$ around the graph of λ_D.

Proof: Let $M \supseteq D$ be a witness of $\lambda_D(\alpha)$. Then, $M_f = \{f(y) : y \in M\}$ satisfies $K(M_f) \leq \alpha + K(f) + O(1)$ and $|M_f| = |M|$. Hence, $\lambda_{f(D)}(\alpha + K(f) + O(1)) \leq \lambda_D(\alpha)$. Let $M' \supseteq f(D)$ be a witness of $\lambda_{f(D)}(\alpha)$. Then, $M'_{f^{-1}} = \{f^{-1}(y) : y \in M'\}$ satisfies $K(M'_{f^{-1}}) \leq \alpha + K(f) + O(1)$ and $|M'_{f^{-1}}| = |M'|$. Hence, $\lambda_D(\alpha + K(f) + O(1)) \leq \lambda_{f(D)}(\alpha)$ (since $K(f^{-1}) = K(f) + O(1)$). ■

C. Finding the MDL Code

Given $D \subseteq \{0,1\}^n$, the data to explain, and the model class consting of all models (sets) $M \subseteq \{0,1\}^n$ that have complexity at most $K(M) \leq \alpha$. Here, α is the maximum complexity of an explanation we allow. As usual, we denote $m = |M|$ and $d = |D|$. We search for programs p of length at most α that print a finite set $M \supseteq D$. Such pairs (p, M) are possible explanations. The *best explanation* is defined to be the (p, M) for which $\delta(D \mid M, d)$ is minimal. Since the function $\delta(D \mid M, d)$ is not computable, there is no algorithm that halts with the best explanation. The programs use unknown computation time and thus we can never be certain that we have found all possible explanations. Following [3], we can overcome this problem: Initially, we are given data sample D.

We minimize the randomness deficiency by minimizing the MDL code length, justified by

$$\beta_D(\alpha) = \lambda_D(\alpha) - K(D|d) \tag{5}$$

as in [3], and thus maximizing the fitness of the model for this data sample. To this end, run all programs dovetailed fashion. If a program, say p, halts, then check its output to see whether it is a subset, say M, of $\{0,1\}^n$ in agreed-upon standard notation. If so, then check whether that subset contains all elements in the data sample D. At every computation step t consider all pairs (p, M) such that program p has printed the set M containing D by time t. Let (p_t, L_t) stand for the pair (p, M) such that $|p| + \log \binom{m}{d}$ is minimal among all these pairs (p, M). The best hypothesis L_t changes from time to time due to the appearance of a better hypothesis. Since no hypothesis is selected as the best one twice, from some moment onwards the explanation (p_t, L_t) which is best does not change anymore.

IV. DOES SHORTER MDL CODE MEAN BETTER MODEL?

Thus, if we continue to approximate the MDL code then we will eventually reach the optimal code which is approximately the best explanation at the given model complexity. During this process we have some guarantee of goodness (details omitted). That is the good news. The bad news is, that we do not know when we have reached this optimal solution, and the noncomputability of computing λ_D to a given precision assures us that there simply does not exist a convergence criterion we could use to terminate the approximation somewhere close to the optimum. Thus, in practice we must terminate the search prematurely. A natural assumption is that the longer we approximate the optimal MDL code the better the resulting model explains the data. Thus, many practitioners simply assume that if one approximates the MDL code, than every next shorter MDL code also yields a better model. Alas, this is not true. To give an example that shows where things go wrong it is easiest to first give the conditions under which premature search termination is all right, slightly correcting an idea first given in [3].

Assume that in the indirect MDL algorithm, as described in Section III-C, we change the currently best explanation (p_1, M_1) for data D ($|D| = d$) to the explanation (p_2, M_2) only if $|p_2| + \log \binom{|M_2|}{d}$ is much less than $|p_1| + \log \binom{|M_1|}{d}$, say $|p_2| + \log \binom{|M_2|}{d} \leq |p_1| + \log \binom{|M_1|}{d} - c' \log\log \binom{2^n}{d}$ for a constant c'. We show: if c' is large enough and p_1 is a shortest program of M_1, d, then $\delta(D \mid M_2, d)$ is less than $\delta(D \mid M_1, d)$. That is, every time we change the explanation we improve its goodness unless the change is just caused by the fact that we have not yet found the minimum length program for the current model.

THEOREM 2: Let (p_1, M_1) and (p_2, M_2) be two consecutive candidate best explanations in the search process for the best model for data sample D ($|D| = d$, $0 < d < 2^n$) above. There is a constant c such that if $|p_2| + \log \binom{|M_2|}{d} \leq |p_1| + \log \binom{|M_1|}{d} - (|p_1| - K(M_1)) - 2c \log\log \binom{2^n}{d}$ then $\delta(D \mid M_2, d) \leq \delta(D \mid M_1, d) - c \log\log \binom{2^n}{d} + O(1)$.

Proof: For every pair of sets $M_1, M_2 \supseteq D$ we have

$$\delta(D \mid M_2, d) - \delta(D \mid M_1, d) = \Lambda(M_2) - \Lambda(M_1) + \Delta,$$

with

$$\begin{aligned}\Delta &= K(M_2 \mid d) + K(D \mid M_2, d) - K(M_1 \mid d) + K(D \mid M_1, d) \\ &\leq K(M_1, D \mid d) - K(M_2, D \mid d) + O(1) \\ &\leq K(M_1 \mid M_2, D) + O(1).\end{aligned}$$

Since

$$\begin{aligned}\Lambda(M_2)-\Lambda(M_1) &\leq |p_2| + \log\binom{|M_2|}{d} - \Lambda(M_1)\\ &= |p_2| + \log\binom{|M_2|}{d} - (|p_1| + \log\binom{|M_1|}{d})\\ &\qquad + (|p_1| - K(M_1))\\ &\leq -2c\log\log\binom{2^n}{d},\end{aligned}$$

we need to prove that $K(M_2 \mid M_1, D) \leq c\log\log\binom{2^n}{d} + O(1)$. Note that (p_1, M_1), (p_2, M_2) are consecutive explanations in the algorithm and every explanation may appear only once. Hence to identify M_1 we only need to know p_2, M_2, α and D. Since p_2 may be found from M_2 and length $|p_2|$ as the first program computing M_2 of length $|p_2|$, obtained by running all programs of length at most α dovetailed style, we have $K(M_2 \mid M_1, D) \leq 2\log|p_2| + 2\log|\alpha| + O(1) \leq 4\log\log\binom{2^n}{d} + O(1)$. Hence we can choose $c = 4$. ■

Thus, to be sure that in the sequence $(p_1, M_1), (p_2, M_2), \ldots$ of candidate explanations of ever shorter MDL codes the explanation (p_{i+1}, M_{i+1}) is actually a better explanation for the data than the preceding (p_i, M_i), it suffices that $|p_{i+1}| + \log\binom{|M_{i+1}|}{d} \leq |p_i| + \log\binom{|M_i|}{d} - (|p_i| - K(M_i)) - 2c\log\log\binom{2^n}{d}$. The unknown, and in general noncomputable, quantification of the required improvement in MDL code length is $|p_i| - K(M_i)$. If we have an hypothesis M_i encoded by a program p_i that is far from optimal, then the slack in model code length given by $|p_i| - K(M_i)$ is large, and it is possible that we improve the MDL code length by giving a worse hypothesis M_{i+1} using, however, an encoding p_{i+1} that is shorter than the encoding p_i of the previous candidate M_i. Thus,

COROLLARY 1: (i) On the one hand, if $|p_{i+1}| + \log\binom{|M_{i+1}|}{d} \leq |p_i| + \log\binom{|M_i|}{d} - 2c\log\log\binom{2^n}{d}$ and $|p_i| = K(M_i) + O(1)$, then M_{i+1} is a better explanation for data D than is M_i, in the sense that $\delta(D \mid M_{i+1}, d) \leq \delta(D \mid M_i, d) - 4\log\log\binom{2^n}{d} + O(1)$.

(ii) On the other hand, if $|p_i| - K(M_i)$ is large, then M_{i+1} may be a much worse explanation than M_i as we show with some examples below.

V. Inferring a Grammar (DFA) From Positive Examples

The field of grammar induction studies a whole class of algorithms that aims at constructing a grammar by means of incremental compression of the data set represented as a digraph representation of a DFA accepting the data set. This digraph can be seen as a model for the data set. Every word in the data set is represented as a path in the digraph with the symbols either on the edges or on the nodes. The learning process takes the form of a guided incremental compression of the data set by means of merging or clustering of the nodes in the graph. None of these algorithms explicitly makes an estimate of the data-to-model code. Instead they use heuristics to guide the model reduction. After a certain time a proposal for a grammar can be constructed from the current state of the compressed graph. Examples of such algorithms are SP [8], [7], EMILE [4], ADIOS [6], and a number of DFA induction algorithms, specifically evidence driven state merging (EDSM), [5]. To analyse the MDL estimation for DFAs, given a data sample, we first fix details of the code. For the model code, the coding of the DFA, we encode as follows. Let $A = (Q, S, t, q_0, F)$ with $q = |Q|$, $s = |S|$. Then there q possibilities for F, by renaming of the states we can always take care that $F \subseteq Q$ are the last f states of Q. There are q^{sq} different possibilities for t, and q possibilities for q_0. Altogether, for every choice of q, s there are $\leq q^{qs+2}$ distinct DFAs, some of which may accept the same languages. We encode a DFA A with q states and s symbols in self-delimiting format in $(qs+3)\lceil\log q\rceil + 2\lceil\log\log q\rceil + \lceil\log s\rceil + 2\lceil\log\log s\rceil \approx (qs+4)\log q + 2\log s$ bits. Thus, we reckon the model cost of a (q, s)-DFA as $m(q, s) = (qs+4)\log q + 2\log s$ bits. Let $L^n(A) = L(A) \bigcap \{0,1\}^n$. Given a DFA model A, the word length n, in $\log n + 2\log\log n$ bits which we simplify to $2\log n$ bits, and the size d of the data sample $D \subseteq \{0,1\}^n$, we can describe D by its index j in the set of d choices out of $l = L^n(A)$ items, that is, up to rounding upwards, $\log\binom{l}{d}$ bits. For $d = 1$ or $d = l$ we set the data-to-model cost to $1 + 2\log n$, for $1 < d \leq l/2$ we set it to $2\log n + lH(d/l)$ with H the Shannon entropy function (ignoring the possible savings of $\log l/2$ term), and for $l/2 < d < l$ we set it to the cost of $l - d$. This reasoning brings us to the following MDL cost of a data sample D for DFA model A: The *MDL code length* of a data sample D of d strings of length n, for a DFA model A such that $D \subseteq L^n(A)$, denoting $l = |L^n(A)|$, is given by

$$MDL(D, A) = (qs+4)\log q + 2\log s + 2\log n + lH(d/l).$$

Given data sample D and DFA A with $D \subseteq L^n(A) \subseteq \{0,1\}^n$, we can estimate the randomness deficiency. By (2), the randomness deficiency is

$$\delta(D \mid A, d, n) = \log\binom{l}{d} - K(D \mid A, d, n).$$

Then, substituting the estimate of $lH(d/l)$ for $\log\binom{l}{d}$, up to logarithmic additive terms,

$$\delta(D \mid A, d, n) = lH(d/l) - K(D \mid A, d, n).$$

Thus, by finding a computable upper bound for $K(D \mid A, d, n)$, we can obtain a computable lower bound on the randomness deficiency $\delta(D \mid A, d, n)$ that expresses the fittness of DFA model A with respect to data sample D.

A. Less MDL Code Length Doesn't Mean Better Model

We show by example that the randomness deficiency behaves independently of the MDL code: the randomness deficiency can either grow or shrink with a reduction of the length of the MDL code. Let the set D be a sample set consisting of 50% of all binary strings of length n with an even number of 1's. Note, that the number of strings with an even number of 1's equals the number of strings with an odd number of ones, so $d = |D| = 2^n/4$. Initialize with a DFA A such that $L^n(A) = D$. We can obtain D directly from A, n, so we have $K(D \mid A, n) = O(1)$, and since $d = l$ we have $\log\binom{l}{d} = 0$, so that altogether $\delta(D \mid A, d, n) = -O(1)$, while $MDL(D, A) = (qs + 4)\log q + 2\log s + 2\log n + O(1) = (2q + 4)\log q + 2\log n + O(1)$, since $s = 2$. Without loss of generality we can assume that the MDL algorithm involved works by splitting or merging nodes of the digraphs of the produced sequence of candidate DFA's. But the argument works for every MDL algorithm, whatever technique it uses.

Initialize: Assume that we start our MDL estimation with the trivial DFA A_0 that literally encodes all d elements of D as a binary directed tree with q nodes. Then, $2^n/2 - 1 \leq q \leq 2^{n+1} - 1$, which yields

$$MDL(D, A_0) \geq 2^n n/2$$
$$\delta(D \mid A_0, d, n) \approx 0,$$

the latter equation since $d = l$, so $\log\binom{l}{d} = 0$, and $K(D \mid A_0, d, n) = O(1)$. Since the randomness deficiency $\delta(D \mid A_0, d, n) \approx 0$, we have that A_0 is a best fitting model for D. Indeed, it represents all conceivable properties of D since it literally models D. However, A_0 doesn't achieve the optimal MDL code.

Better MDL estimation: In a later MDL estimation we improve the MDL code by inferring the parity DFA A_1 with two states ($q = 2$) that checks the parity of 1's in a sequence. Then,

$$MDL(D, A_1) \leq 8 + 2\log n + \log\binom{2^n/2}{2^n/4} \approx 2^{n-1} - n/4$$
$$\delta(D \mid A_1, d, n) = \log\binom{2^n/2}{2^n/4} - K(D \mid A_1, d, n)$$
$$\approx 2^{n-1} - n/4 - K(D \mid A_1, d, n)$$

We now consider two different instantiations of D, denoted as D_0 and D_1. The first one is regular data, and the second one is random data.

Case 1, regular data: Suppose $D = D_0$ consisting of the lexicographical first 50% of all n-bit strings with an even number of occurrences of 1's. Then $K(D_0 \mid A_1, d, n) = O(1)$ and

$$\delta(D_0 \mid A_1, d, n) = 2^{n-1} - O(n).$$

In this case, even though DFA A_1 has a much better MDL code than DFA A_0, it has nonetheless a much worse fit since its randomness deficiency is far greater.

Case 2, random data: Suppose, $D = D_1$ where D_1 is a random subset consisting of 50% of the n-bit strings with even number of occurrences of 1's. Then, $K(D_1 \mid A_1, d, n) = \log\binom{2^n/2}{2^n/4} + O(1) \approx 2^{n-1} - n/4$, and

$$\delta(D_1 \mid A_1, d, n) \approx 0.$$

In this case, DFA A_1 has a much better MDL code than DFA A_0, and it has equally good fit since both randomness deficiencies are about 0.

REMARK 1: We conclude that improved MDL estimation of DFA's for multiple data samples doesn't necessarily result in better models, but can do so nonetheless.

REMARK 2 (SHORTEST MODEL COST): By Theorem 2 we know that if, in the process of MDL estimation by a sequence of decreasing MDL codes, a candidate DFA is represented by its shortest program, then the following candidate DFA which improves the MDL estimation is actually a model of at least as good fit as the preceding one. Let us look at an Example: Suppose we start with DFA A_2 that accepts all strings in $\{0, 1\}^*$. In this case we have $q = 1$ and

$$MDL(D_0, A_2) = \log\binom{2^n}{2^n/4} + O(1)$$
$$\delta(D_0 \mid A_2, d, n) = \log\binom{2^n}{2^n/4} - O(1).$$

Here $\log\binom{2^n}{2^n/4} = 2^n H(\frac{1}{4}) - O(n) \approx 3 \cdot 2^{n-2} - O(n)$. ($H(\frac{1}{4}) \approx \frac{2}{3}$) Suppose the subsequent candidate DFA is the parity machine A_1. Then,

$$MDL(D_0, A_1) = \log\binom{2^n/2}{2^n/4} + O(1)$$
$$\delta(D_0 \mid A_1, d, n) \approx \log\binom{2^n/2}{2^n/4} - O(1),$$

since $K(D_0 \mid A_1, d, n) = O(1)$. Since $\log\binom{2^n/2}{2^n/4} = 2^{n-1} - O(n)$, we have $MDL(D_0, A_2) \approx \frac{2}{3}MDL(D_0, A_2)$, and $\delta(D_0 \mid A_2, d, n) \approx \frac{2}{3}\delta(D_0 \mid A_1, d, n)$. So the improved MDL cost is accompanied by improved fitness by decreasing randomness deficiency. This indeed is forced by Theorem 2, since both DFA A_1 and DFA A_2 have $K(A_1), K(A_2) = O(1)$. That is, the DFA's are represented and costed according to their shortest programs (a forteriori of length $O(1)$) and therefore improved MDL estimation increases the fitness of the successive DFA models significantly.

REFERENCES

[1] Mitchell T. M., , Machine Learning, McGraw-Hill, New York, (1997)

[2] Li M., Vitányi P.M.B. An Introduction to Kolmogorov Complexity and Its Applications, 2nd ed., Springer-Verlag, New York, (1997)

[3] Vereshchagin N.K., Vitányi P.M.B., Kolmogorov's structure functions and model selection, IEEE Trans. Information Theory, vol. 50, nr. 12, 3265–3290, (2004)

[4] Adriaans P., Vervoort M., The EMILE 4.1 grammar induction toolbox, In: *Grammatical Inference: Algorithms and Applications; 6th International Colloquium, ICGI 2002*, P. Adriaans and H. Fernau and M. van Zaanen eds., LNCS/LNAI 2484, 293–295, 2002.

[5] Lang K. J., Pearlmutter B. A., Price R. A. , Results of the Abbadingo One DFA learning competition and a new evidence-driven state merging algorithm. In: *Grammatical Inference: Algorithms and Applications; 6th International Colloquium, ICGI 2002*, P. Adriaans and H. Fernau and M. van Zaanen eds., LNCS/LNAI 2484, 1–12, 2002.

[6] Solan Z., Horn D., Ruppin E., Edelman S., Unsupervised learning of natural languages, *Proc. Natn'l Academy Sci.*, 102: 33(2005), 11629-11634.

[7] Wolff J.G., Computing As Compression: An Overview of the SP Theory and System, *New Generation Comput.*, 13:2(1995), 187–214.

[8] Wolff, J. G., Information Compression by Multiple Alignment, Unification and Search as a Unifying Principle in Computing and Cognition, *J. Artificial Intelligence Research*, 19:3(2003), 193–230.