The Complexity of Strict Minimum Message Length Inference

G. E. FARR AND C. S. WALLACE

School of Computer Science and Software Engineering, Monash University (Clayton Campus), Clayton, Victoria 3800, Australia
Email: {gfarr,csw}@csse.monash.edu.au

Strict Minimum Message Length (SMML) inference is an information-theoretic criterion for inductive inference introduced by Wallace and Boulton and is known to possess several desirable statistical properties. In this paper we examine its computational complexity. We give an efficient algorithm for the binomial case and indeed for any SMML problem that is essentially one-dimensional in character. The problem in general is shown to be NP-hard. A heuristic is discussed which gives good results for binomial and trinomial SMML inference. The complexity of the trinomial case remains open and is worth further investigation.

Received 21 May 2001; revised 12 November 2001

1. INTRODUCTION

This paper concerns some algorithmic questions that arise in an information-theoretic approach to inductive inference.

If a model, or theory or explanation, is sought for some data, then in order to be able to decide which model to adopt, it is necessary to have, firstly, a *criterion* by which to measure how well a given model explains the data and, secondly, an *algorithm* for finding which model is best according to that criterion. We can think of the criterion as an objective function, so that we have an optimization problem for which we desire an efficient, accurate algorithm.

Criteria in common use in statistical inference include maximum likelihood, posterior mean and posterior mode. For discussion of the advantages and disadvantages of these and other criteria, see, e.g. [1, Ch. 12].

In recent decades, an arguably more general criterion based on information theory has been developed. The fundamental principle is that the best model is the one which allows the shortest combined encoding of the model and the data. A form of this principle may be discerned in the pioneering work of Solomonoff [2]. It is embodied in the Minimum Message Length (MML) criterion, developed and investigated by Wallace, Boulton and others [3, 4, 5, 6]. Rissanen [7, 8, 9, 10] has developed a related approach known as Minimum Description Length (MDL) inference. In MML inference, we envisage a *message* which encodes firstly the model (using a code which is optimal under whatever prior assumptions we make about the distribution of possible models) and secondly the data given the model (i.e. using a code which would be optimal if the data did indeed arise from that model). Such messages need not actually be constructed; we simply want to be able to attach a message *length* to any model. Minimizing this message length requires us to optimize a tradeoff between the simplicity of the model and how well the model fits the data (measured by the lengths of the first and second parts of our message respectively).

MML inference has been applied, for example, to numerical taxonomy [3], bioinformatics [11, 12, 13, 14], bushfire prediction using decision trees [15], medical diagnosis [16, 17] and the study of stone circle geometries [18].

To illustrate how the MML principle is applied, consider for example numerical taxonomy (which was also the first application of MML [3]). Here, we have a collection of items (perhaps biological specimens) and certain measurements for each item. We wish to group the items into classes, perhaps hoping to discern species or the like. The inference to be done is unsupervised: we do not know, in advance, how any particular item is to be classified. Each model consists of several multidimensional probability distributions (being our 'classes') over the measurement space, and to each distribution in the model is associated a proportion specifying its contribution to the whole 'mixture'. The data is the set of measurements, essentially a point in our multi-dimensional measurement space. Given some data, we envisage for each possible model a message, as follows. The first part consists of a statement of the model. This will state the number of distributions in the mixture, the proportions in which the distributions are to be mixed and some encoding of the distributions themselves (perhaps by stating their means and variances, if they are assumed to be Normal). The second part of the message encodes the data in a way that makes efficient use of the model stated in the first part. For each item, we state which distribution (or class) it is assumed to come from and then encode the values of its measurements using an efficient code based on that distribution. (We have glossed over some technical issues here; see [3].) We can thus assign a message length to any model and search over the model space for one with minimum message length (or

close to it). Notice that models of different numbers of classes can be dealt with seamlessly, as members of the same model space. In fact, unlike most other methods of clustering or taxonomy, the MML approach yields the appropriate number of classes as part of its output; it does not have to be told this information up front.

This paper concerns Strict Minimum Message Length (SMML) inference, which was introduced by Wallace and Boulton in 1975 [5]. SMML inference follows the MML principle strictly, and tries to be perfectly accurate in the message lengths it constructs. We introduce it briefly now and give details in the next section.

Consider the space of possible models. Although it may contain a continuum (e.g. if models are specified by real-valued parameters), any code for models will only allow us to encode (in the first part of our message) members of some countable subset of the model space. Inference here amounts to mapping each data item to some model from this subset. The message for that data item consists of an encoding of its corresponding model followed by an encoding of the data using a code based on the model. The problem is to find the mapping, from data space to our subset of the model space, so as to minimize the expectation, over all data, of the message length. We expand on this in the next section; for further treatment of SMML and MML in general, see [6].

SMML estimation has several desirable statistical properties [5, 6]: invariance under one-to-one transformations of either the parameter space or the data; dependence on sufficient statistics (as it can be expressed as a function of any sufficient statistic); a degree of precision which is appropriate considering the expected estimation error; and the capability to treat parameter estimation, model selection and hypothesis testing in the same framework. Furthermore, SMML estimation is virtually indistinguishable, by any fair criterion (and using only the available data), from a 'true oracle' which always returns the 'correct' model [19]. Connections between SMML and Kolmogorov complexity are discussed in [20].

Whichever criterion is used, carrying out practical inference requires algorithms to find the best models according to that criterion. This paper concerns the particular algorithmic issues raised by SMML inference.

One immediate difficulty with SMML inference is that it depends on all possible data, not just some particular observation at hand which we seek to explain. In many situations, the set of all possible values for the data is astronomical in size, so that SMML inference is already infeasible. For example, if the data is some sequence of N bits, then for SMML we must find out how to do inference for all the 2^N possible sequences of N bits (unless the set of models considered allows projection of the sequences into a smaller set of sufficient statistic values).

However, SMML's advantages (outlined above) are such that it remains important to consider how easy or difficult it is in cases where the set of all possible data is of more manageable size. We therefore consider its complexity as a function of the size of the set of all possible data, rather than (for example) the logarithm of that size.

We find that, even when making such a generous allowance for the possible data, SMML inference remains NP-hard in general. However, in the case where the likelihood function is binomial and the prior distribution is arbitrary, we give a polynomial time algorithm. Indeed, the same approach gives a polynomial time algorithm whenever the SMML inference problem in question is one-dimensional, in a sense to be explained. We raise the question of the complexity of SMML inference for the trinomial (and more general multinomial) distribution. This question remains open and worthy of further investigation, as it appears to be rich in structure and related to geometric packing problems.

Given the difficulty of *Strict* MML inference even in simple cases, what is to be done? In practice, it is necessary to be less strictly accurate in working out message lengths. 'Ordinary' MML inference is related to SMML by a series of approximations (see, e.g. [6]) and the algorithmic problems it gives are much easier than those raised by SMML, although still often non-trivial. All the practical applications cited above use ordinary MML rather than SMML.

After a more precise description of SMML inference in the next section, the paper moves from the positive (binomial SMML, Section 3), to the open (trinomial SMML, Section 4), to the negative (NP-hardness in general, Section 5).

2. SMML INFERENCE

We now describe SMML inference, using similar notation and terminology to the papers that introduced it [5, 6].

Let $\mathcal{X}$ be the set of all the possible values of the *data*. We *observe* some $x \in \mathcal{X}$ and wish to explain it. The possible explanations of the data in $\mathcal{X}$ make up the set Θ of possible *models*. We assume there is a *prior* distribution or density h on Θ which captures our beliefs about the probabilities of the models before we have observed any data. The *likelihood function* $f(x|\theta)$ gives the probability attached to the data x under the model θ.

From the prior h and the likelihood f it is easy to calculate the joint distribution $p(x, \theta) = f(x|\theta)h(\theta)$ and the marginal prior distribution over the data, $r(x) = \sum_{\theta\in\Theta} p(x, \theta)$ (where the sum becomes an integral if h is a density).

We seek a function $m : \mathcal{X} \to \Theta$ which attaches to any observation $x \in \mathcal{X}$ a model $\theta = m(x)$ which is in some sense a 'good' explanation for x.

In general not every model in Θ will ever be used to explain some $x \in \mathcal{X}$. The range of m, which we call Θ^*, will typically be a proper subset of Θ. In fact, when $\mathcal{X}$ is discrete and countable, Θ^* will be a discrete subset of Θ, while Θ might well be a continuum.

Each $\theta^* \in \Theta^*$ then has a probability $q(\theta^*) = \sum_{x:m(x)=\theta^*} r(x)$.

Consider a *message* which states, for some given $x \in \mathcal{X}$, firstly which model is used to explain x (i.e. $m(x)$) and secondly x itself *given* the model. The first part requires a string of $-\log_2 q(m(x))$ bits, since $m(x)$ has probability $q(m(x))$. The second part encodes x using a code which

would be optimal if the model $m(x)$ were true, so x is assumed to be distributed according to $f(x|m(x))$ and is encoded by a string of $-\log_2 f(x|m(x))$ bits. The total message length for the data x is thus

$$-\log_2 q(m(x)) - \log_2 f(x|m(x)) \tag{1}$$

bits. (In assigning $-\log_2 p$ bits to a message part of probability p, we are of course making an approximation, since real codes use whole numbers of bits, but we ignore this.)

Any such code for data, to be decodeable, must be constructed without use of any knowledge of the actual observed data x to be encoded, since this is not available to a decoder until the message is decoded. The construction must therefore aim to minimize the message length which is expected prior to the observation.

In Strict MML inference, we do not seek to minimize the message length (1) in isolation, since it depends (via its first summand) not just on $m(x)$ but also on other values of the map m. Rather, we seek an m which minimizes the expectation, over all $x \in \mathcal{X}$, of that message length. This amounts to minimizing the message length expected prior to observation, as just discussed.

We therefore minimize

$$\text{expected message length} = -\sum_{x\in\mathcal{X}} r(x)\log_2 q(m(x)) - \sum_{x\in\mathcal{X}} r(x)\log_2 f(x|m(x)) \tag{2}$$

$$= -\sum_{\theta^*\in\Theta^*} q(\theta^*)\log_2 q(\theta^*) - \sum_{x\in\mathcal{X}} r(x)\log_2 f(x|m(x)). \tag{3}$$

A map m which minimizes this expected message length should on average produce better (i.e. shorter message length) models than if we chose m to minimize any one of the message lengths (1), even though the message lengths for particular data x may be somewhat longer. This is Strict MML inference [5, 6].

The algorithmic problem we want to solve is thus:

SMML
Given: discrete or continuous set of models, Θ.
Input: set $\mathcal{X}$ of possible data, prior distribution h on Θ, likelihood f.
Output: a function $m : \mathcal{X} \to \Theta$ which minimizes the expected message length given by (2), (3).

Note that the prior distribution or density h enters the message length only via the marginal prior for the data, r. For this reason we will sometimes use r and not h (alongside f) in describing an SMML inference problem.

It is useful to regard the map m as partitioning the set $\mathcal{X}$, with each part of the partition being the set of preimages of some $\theta^* \in \Theta^*$. Let the parts of the partition be $X_1, \ldots, X_\nu$, with $\mathcal{X} = \bigcup_{j=1}^{\nu} X_j$ and the X_j pairwise disjoint. Write θ_j^* for the common image under m of all the $x \in X_j$. It will be convenient to put $q(X_j) = q(\theta_j^*)$, since this probability depends on X_j and not on the common value of $m(x)$ for all $x \in X_j$. Specifying m amounts to specifying $(X_j, \theta_j^*)_{j=1}^{\nu}$. If m minimizes the above expected message length, then the partition of $\mathcal{X}$ it induces will be called an *SMML partition*.

The total expected message length is easily expressed as a sum of contributions due to the several parts of the partition, by rewriting (2):

$$\text{expected message length} = \sum_{j=1}^{\nu}\left(-q(X_j)\log_2 q(X_j) - \sum_{x\in X_j} r(x)\log_2 f(x|\theta_j^*)\right). \tag{4}$$

Some crucial observations should now be made [5]. The contribution to the sum (4) made by (X_j, θ_j^*) (which is just the expression inside the sum for index j) depends only on X_j and θ_j^* and not on any X_k or θ_k^* for $k \neq j$. The choices of X_j are not of course independent, because they are constrained to form a partition of $\mathcal{X}$. However, the choices of θ_j^* are indeed independent. We can choose θ_j^* to minimize this jth contribution and then any SMML partition which has X_j as one of its parts will associate the same θ_j^* to X_j. Furthermore, θ_j^* only influences the expected length of that portion of the message which states the data given the model; expression (4) makes it clear that the expected length required to state the model depends only on the partition. In finding the optimal m, the difficult part will often be finding the SMML partition; computation of the values of m on parts of this partition may well be easy, as in the examples considered in this paper.

The messages considered here state the model and then the data given the model; this is of course equivalent to just stating the data together with the model. Such a message must necessarily be longer than a message which just states the data. The expected length of such a message is just the entropy of r and this gives a convenient lower bound for SMML.

We close this section with some notation. We will be working with intervals of integers and reals. Integer intervals will be written $a..b$, while real intervals will be written $[a, b]$ as usual. (We often find it clearer to represent all integer intervals in this notation, even when some happen to be singletons.)

3. THE BINOMIAL CASE

Suppose N Bernoulli trials are conducted and the number x of successes is observed, where the probability of success in a single trial is p. In our notation, $\mathcal{X} = 0..N$, $\Theta = [0, 1]$ and $f(x|p) = \binom{N}{x} p^x (1-p)^{N-x}$. Some prior distribution h on $[0, 1]$ is assumed to have been specified, and from it and f is calculated the marginal prior r for the data.

To do SMML inference we must determine a partition of the data space $\mathcal{X}$ and a probability estimate for each part of the partition, so as to minimize the expected message length described in the previous section. The remainder of this section is concerned with how to do this in the binomial case.

Suppose that some part Y of a partition of $\mathcal{X}$ is given. We want to know what $p^* \in [0, 1]$ the members of Y are to be mapped to. As discussed in Section 2, we only need to find that p^* which minimizes the contribution which (Y, p^*) make to the expected length of the portion of the message which states x given the model. We want p^* which minimizes $-\sum_{x\in Y} r(x) \log_2 \binom{N}{x} (p^*)^x (1-p^*)^{N-x}$. This is easily found to be

$$p^* = \frac{\sum_{x\in Y} x.r(x)}{N \sum_{x\in Y} r(x)}, \tag{5}$$

the weighted average of the members of Y, weighted by the distribution r and expressed as a proportion of N. It is routine to show that, although in principle the parts of the partition may be any subsets of $0..N$, the parts of any SMML partition will be intervals (of integers). Since p^* is so easily found from Y and r, the main problem is to find the partition.

Our algorithm for SMML inference in this binomial case works by starting with a trivial partition of $0..0$ and building up certain partitions of $0..n$, for $n = 1, \ldots, N$ in turn, finishing up with an SMML partition of $0..N$.

The algorithm is based on the following observations. Recall the way the expected message length can be expressed in terms of contributions by the parts of the partition, as in (4). Any partition of $0..n$ can thus be given a partial expected message length, being just the sum of the contributions each part would make to a total expected message length. Any SMML partition of $0..N$ which has a 'boundary' at n, in that one of its parts finishes at n, will induce a partition of $0..n$ which has minimum partial expected message length among all partitions of $0..n$. (Succinctly, optimum partitions induce optimum subpartitions.)

We use the following notation in describing the algorithm.

- $\pi(N, n, k)$ is the best (in SMML sense, i.e. in terms of partial expected message length as explained above) partition of $0..n$, when $\mathcal{X} = 0..N$, whose last part (i.e. the part including n) is $k..n$.
- $\Pi(N, n)$ is the best (in the same SMML sense) partition of $0..n$, when $\mathcal{X} = 0..N$.

It is clear that the SMML partition $\Pi(N, N)$ will be that $\pi(N, N, k)$ which gives the least expected message length. We take whatever k gives this minimum, where $0 \le k \le N$. We then need to be able to compute the $\pi(N, N, k)$. Since optimum partitions induce optimum subpartitions, the partition $\pi(N, N, k)$ induces an optimum subpartition $\Pi(N, k-1)$ of $0..(k-1)$. We are back to computing a $\Pi(N, n)$, which we have seen how to do. This recursive description of the $\pi(N, n, k)$ and $\Pi(N, n)$ in terms of each other forms the basis of, and justifies, the following dynamic programming algorithm.

ALGORITHM 1. Finding an SMML partition of $0..N$.

1. Input: number of trials N,
   marginal prior for data, r (as list of values).
2. Initialization:
   $$\pi(N, 0, 0) := \{0..0\},$$
   $$\Pi(N, 0) := \{0..0\},$$
   so each starts with the partition of $0..0$ consisting of the single interval $0..0$.
3. For $n := 1, \ldots, N$:
   3.1. for $k := 0, \ldots, n$,
   $$\pi(N, n, k) := \Pi(N, k-1) \cup k..n,$$
   i.e. add $k..n$ to $\Pi(N, k-1)$;
   3.2. $\Pi(N, n) :=$ best of all the $\pi(N, n, k), k = 0, \ldots, n$, i.e. the $\pi(N, n, k)$ which, over k, gives the minimum expected message length. This requires calculation, when needed, and then storage of the contributions which various intervals make to the total expected message length. (We do not need to be continually recalculating the whole expected message length from scratch.)
4. Output: $\Pi(N, N)$.

Table 1 gives the SMML partitions in the case of a uniform prior distribution (so that r is also uniform) for all $N \le 30$ and the expected message lengths which result. A number of points are worth comment. Firstly, the mirror image of any SMML partition is also an SMML partition (which holds for any prior symmetric in $[0, 1]$) and usually this means there are two asymmetric SMML partitions, although sometimes there is a single, symmetric SMML partition (e.g. $N = 1, 2, 4, 5, 6, 7, 11, 13, 14, 18, 21, 22$). Secondly, for the N shown, the sizes of the parts of the partition (when listed in their natural order) form a unimodal sequence, which is what we intuitively expect with uniform prior. Indeed these sequences of part sizes seem very well behaved, very likely possessing many regularity properties which we have not discovered. Finding such properties should be a good combinatorial exercise. However, the reader should be warned that the partitions are not as regular as may be initially hoped. For example, the number of parts of the SMML partition is 'mostly' non-decreasing as N increases, but not always so. This number of parts decreases by one when N reaches 7, 14 and 23, as seen in the table. The next two values at which it so decreases are 109 and 135, and at present we see no pattern in this phenomenon. Although $\Pi(N, N)$ is not quite monotonic in N, we have examined all values of $\Pi(N, n)$ for all $N \le 20$ and observed that the number of parts of these partitions is a non-decreasing function of n (for fixed N) and a non-increasing function of N (for fixed n).

This algorithm requires $O(N^2)$ real number operations.

A heuristic argument can be used to approximate the SMML partition in linear time, and this gives some insight into the number and sizes of the parts of that partition. The expected message length to be minimized is given by (4).

TABLE 1. SMML partitions for a binomial distribution with uniform prior.

N	SMML partition	E(msg length) (bits)
1	{0. .1}	1.000000
2	{0. .2}	1.666667
3	{0. .0, 1. .3}	2.084963
4	{0. .0, 1. .3, 4. .4}	2.453958
5	{0. .0, 1. .4, 5. .5}	2.703677
6	{0. .0, 1. .5, 6. .6}	2.962316
7	{0. .3, 4. .7}	3.164928
8	{0. .0, 1. .5, 6. .8}	3.336556
9	{0. .0, 1. .5, 6. .9}	3.491487
10	{0. .0, 1. .4, 5. .9, 10. .10}	3.646606
11	{0. .0, 1. .5, 6. .10, 11. .11}	3.761542
12	{0. .0, 1. .5, 6. .11, 12. .12}	3.886982
13	{0. .0, 1. .6, 7. .12, 13. .13}	3.998220
14	{0. .3, 4. .10, 11. .14}	4.107097
15	{0. .0, 1. .5, 6. .12, 13. .15}	4.203897
16	{0. .0, 1. .5, 6. .12, 13. .16}	4.289253
17	{0. .0, 1. .6, 7. .13, 14. .17}	4.371787
18	{0. .0, 1. .5, 6. .12, 13. .17, 18. .18}	4.456654
19	{0. .0, 1. .5, 6. .12, 13. .18, 19. .19}	4.531342
20	{0. .0, 1. .5, 6. .13, 14. .19, 20. .20}	4.600605
21	{0. .0, 1. .6, 7. .14, 15. .20, 21. .21}	4.665492
22	{0. .0, 1. .6, 7. .15, 16. .21, 22. .22}	4.736709
23	{0. .3, 4. .11, 12. .19, 20. .23}	4.800845
24	{0. .0, 1. .6, 7. .14, 15. .21, 22. .24}	4.863049
25	{0. .0, 1. .6, 7. .15, 16. .22, 23. .25}	4.920240
26	{0. .0, 1. .6, 7. .14, 15. .22, 23. .26}	4.974435
27	{0. .0, 1. .6, 7. .15, 16. .23, 24. .27}	5.024999
28	{0. .0, 1. .6, 7. .15, 16. .24, 25. .28}	5.079569
29	{0. .0, 1. .6, 7. .15, 16. .24, 25. .29}	5.129015
30	{0. .0, 1. .5, 6. .14, 15. .23, 24. .29, 30. .30}	5.176416

Clearly, the SMML partition will also minimize

$$\sum_{j=1}^{v} \sum_{x \in X_j} r(x)(-\log_2(q(X_j) f(x|\theta_j^*)) + \log_2 r(x)).$$

That is, the partition minimizes the average excess of $-\log_2(q(X_j)f(x|\theta_j^*))$ over $-\log_2 r(x)$. (Recall our discussion in Section 2 of the entropy of r as a lower bound for the expected message length of an SMML partition.)

The heuristic 'grows' the parts (of the partition) one at a time so as to minimize this average within the part being grown, without considering the consequences for parts to be grown later. The data point x bordering the growing part X_j is added to the part if the addition (after adjusting θ_j^*) reduces the average excess of the part. If not, a new part is started with x. The heuristic is easily shown to require $O(N)$ real number operations.

The heuristic is applicable to any SMML problem and does not depend on the particular binomial form considered here. It may be regarded as generating the size (and, more generally, shape) of an 'ideal' part X_j with estimate θ_j^*. Of course, in general no collection of 'ideal' parts will partition the data set. When applied to the binomial problem with uniform prior considered here, we start with a part X_1 comprising the single value $x = 0$. This part will not grow, so a new part X_2 is started comprising the value 1, which is then grown by successive addition of the values 2, 3, etc. until the growth test fails, when a third part X_3 is started and so on until the partition is complete.

This heuristic often yields an SMML partition and empirical study suggests that it always gives a partition with an expected message length very close to the optimum. A slightly elaborated version of this heuristic (for binomial likelihood with uniform prior) finds an optimum solution in most cases and, in all cases examined, finds one with expected message length within 0.015 bits of the optimum. The heuristic approach suggests that, for uniform prior, the excess of minimum expected message length over the lower bound $\log_2(N+1)$ (being the entropy of r; see Section 2) should be about $0.5\log_2(\pi e/6) = 0.2546\ldots$ bits for $N \gg 1$; see [6].

An analysis for large N of the size of an 'ideal' part with estimate θ^* suggests that its size should approximate $(12N\theta^*(1-\theta^*))^{1/2}$, whence it follows that the number of parts should approximate $\sqrt{N}$. The number and sizes of parts in the SMML partition conform closely to these approximations.

Although this section concerns the binomial case, it is apparent that our description of Algorithm 1 does not mention any aspect of the binomial distribution explicitly. In fact, Algorithm 1 serves as a polynomial time algorithm for virtually any SMML problem which is essentially one-dimensional. By this we mean that the data space $\mathcal{X}$ can be enumerated in such a way that the parts of any SMML partition are always intervals (in that enumeration). For polynomial running time, we must also require that the contribution to expected message length due to any part of a partition can be calculated in polynomial time.

It is instructive to compare, for the binomial case, the algorithms for SMML inference (discussed in this section) and ordinary MML inference. Regarding the latter, it can be calculated [6] that (for uniform prior density) the MML estimate of p is

$$\hat{p} = \frac{x + 1/2}{N + 1}.$$

(This may be compared with the maximum likelihood estimate x/N and the posterior mean $(x+1)/(N+2)$.) Once the theoretical work of deriving this formula is done, it is algorithmically trivial to calculate $\hat{p}$ from x.

4. THE TRINOMIAL CASE

We have looked briefly at the trinomial problem, where the result of a single trial can have three rather than two values, say A, B and C, with probabilities p, q, r $(p+q+r=1)$. For N trials, the data can be represented by the pair (x, y) $(0 \le x, y \le N)$ or, more symmetrically, by the triple (x, y, z) $(x, y, z \ge 0;\ x+y+z = N)$, with probability $(N!/(x!y!z!))p^x q^y r^z$. We have not found any polynomial-time algorithm for the SMML partition of the data set,

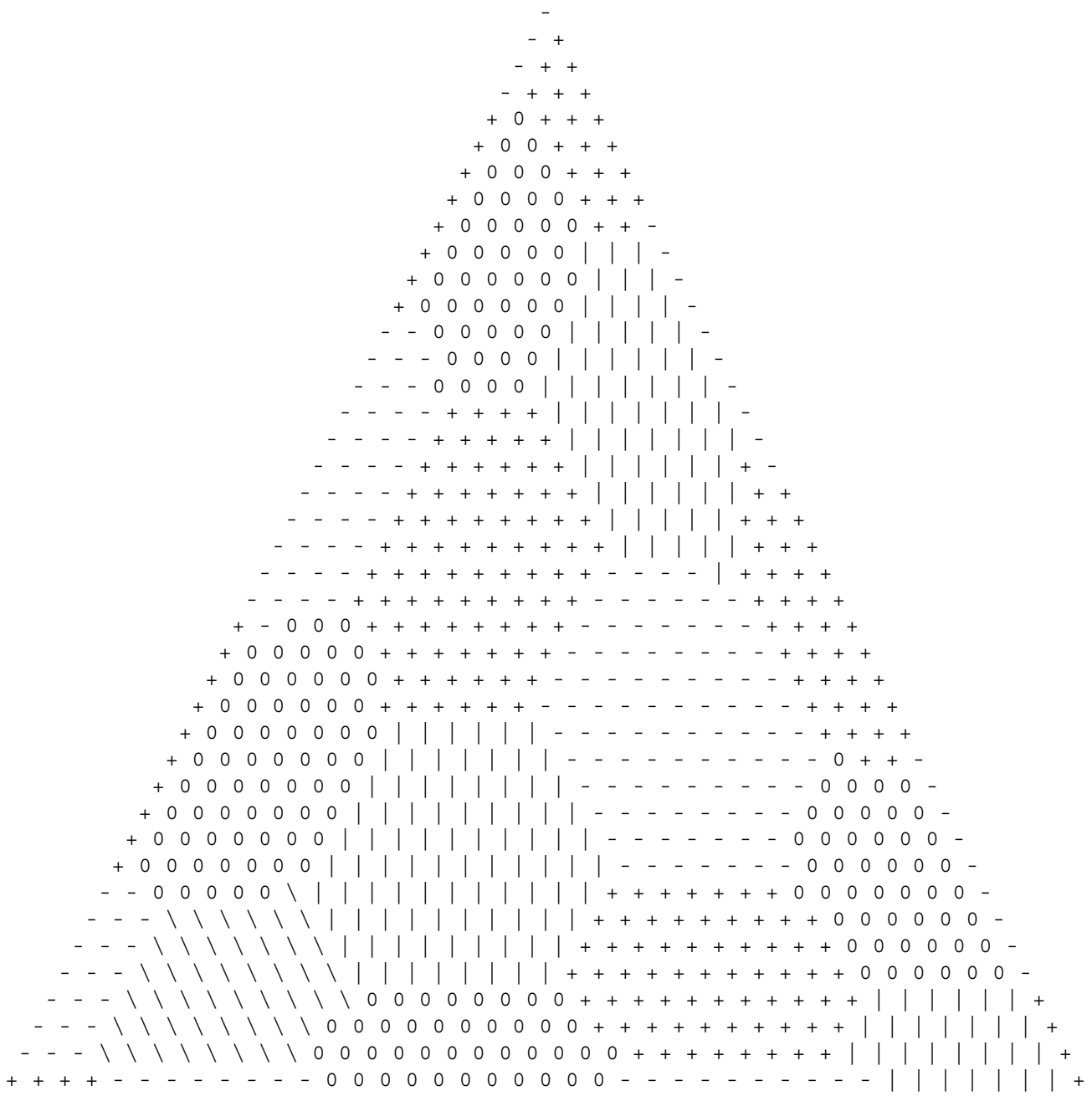

FIGURE 1. Partition of the data space, found by a heuristic argument, for a trinomial distribution, $N = 40$, with uniform prior.

which can be thought of as an equilateral-triangular array of $(N + 1)(N + 2)/2$ points, with $x = N$ at one corner, $x = 0$ along the opposite side, etc. Nor do we have any non-trivial bounds on the complexity of finding the optimal partition, although we suspect it is NP-hard. The heuristic suggests that a part (of the partition) with estimate (p^*, q^*, r^*) should have a size of about $10N(p^*q^*r^*)^{1/2}$ data points and should 'ideally' have a circular shape near the centre of the triangle, becoming elliptical near the sides with long axis parallel to the side. The need for parts to partition the data set suggests that the 'ideal' circles and ellipses will be modified to hexagons. Using the heuristic to grow parts can give different results depending on which border point of the part is tried next for inclusion, and on whether one or several parts are grown at a time. The partition yielded by growing one part at a time, and treating its border points in a systematic order, tends to give convex borders to the grown part, which leads to parts subsequently grown out from this border having shapes far from elliptical, being partly concave. A greedy refinement algorithm, which moves points from near a border from one part to another if this is advantageous, can noticeably improve the partition.

An example of a partition found for $N = 40$, with uniform prior on (p, q, r)-space, is shown in Figure 1. (Each part is indicated by a region of identical symbols.) The sizes and shapes of the parts are in reasonable conformity to the

predictions of the heuristic argument. It has 24 parts and gives an expected message length of 52.13201 bits. The entropy of the marginal prior distribution for the data, $r(\cdot)$, is 51.72164 bits, and our expected message length exceeds this lower bound by only 0.41037 bits, which suggests it is close to optimum.

5. NP-HARDNESS

We now turn to results of a negative character. We will prove that SMML inference is NP-hard in general, even in some quite restricted situations, treating in turn the cases of discrete and continuous parameter spaces.

THEOREM 1. *If the parameter space Θ is discrete, then SMML inference is NP-hard, even for the special case in which the likelihood function is two-valued and the prior distribution is uniform.*

Proof. We prove that SMML inference is NP-hard by reduction from a special case of the well-known NP-hard problem EXACT COVER BY 3-SETS [21] (typically abbreviated X3C).

X3C
Input: a set X (the 'ground set') whose cardinality is a multiple of three and a collection C of three-element subsets of X.
Output: a subset $C^* \subseteq C$ such that each element of X is contained in exactly one member of C^*.

Let $n = 3q = |X|$ and let k be the number of 3-sets in C. If $x \in X$ then the *degree* of x in C, written $\deg_C x$, is the number of members of C which contain x.

It is known that X3C remains NP-complete when restricted to instances in which each element of X is in at most three members of C. It is then not difficult to show that it is still NP-complete if each element of X must belong to *exactly* three members of C.

This can be shown by just adding some extra 3-sets containing elements of degree one or two in C. Let n_i denote the number of elements of X which have degree i in C and observe that $3n_3 + 2n_2 + n_1 = 3k$, so that $3 \mid n_2 + 2n_1$. Let $k' = (n_2 + 2n_1)/3$. Form a list of all the elements of X of degree one or two in C with elements of degree one appearing twice in the list and elements of degree two appearing once in the list. Let this list be $y_1, \ldots, y_{3k'}$. For each $i = 1, \ldots, k'$, add the 3-sets $\{z_{i1}, z_{i2}, z_{i3}\}$, $\{z_{i1}, z_{i2}, y_{3(i-1)+1}\}$, $\{z_{i1}, z_{i3}, y_{3(i-1)+2}\}$, $\{z_{i2}, z_{i3}, y_{3(i-1)+3}\}$ to C, forming a new collection C'. Note that none of the z_{ij} are in X (while all of the y_k are). Write X' for the set obtained by adding all the z_{ij} to X and note that every element of X' has degree three in C'. It is routine to prove that (X', C') has an exact cover by 3-sets if and only if (X, C) does.

We use the name CUBIC X3C for this restriction of X3C to cases where all elements of the ground set have degree three.

We now show that CUBIC X3C $\propto$ SMML.

Suppose (X, C) is now an instance of CUBIC X3C. (Note that $|X| = |C| = n$.) Put $\Theta = C$ and $\mathcal{X} = X$. For all $\theta \in \Theta$, $x \in \mathcal{X}$, define

$$f(x|\theta) = \begin{cases} 1/3, & x \in \theta, \\ 0, & \text{otherwise.} \end{cases}$$

Let h be the uniform distribution over Θ: $\forall\theta$, $h(\theta) = 1/n$. The marginal prior $r(x)$ over the data is then also uniform, with value $1/n$.

Note that in order for the expected message length to be finite it is necessary to have $f(x|m(x)) \neq 0$, which implies that $x \in m(x)$. In that case, $f(x|m(x)) = 1/3$ and the second part of the message has expected length $\log_2 3$. Thus it suffices to minimize the expected length of the first part of the message (subject to $x \in m(x)$, which implies that no more than three of the $x \in \mathcal{X}$ can map to any single $\theta \in \Theta$). This is just the entropy of the distribution q on Θ^* and here this can never be smaller (subject to $x \in m(x)$) than when each $\theta^* \in \Theta^*$ has exactly three preimages under m. This may not always be possible. If it is possible, then the range Θ^* of such an m is an exact cover by 3-sets for (X, C). If it is not possible, then the entropy of q will be larger. (We are using elementary properties of entropy here; see, e.g. [22, Section 6].) The total expected message length in the former case is easily calculated to be $\log_2 n$ bits.

Thus, (X, C) has an exact cover by 3-sets if and only if the inference problem we constructed has a map m which yields a strict minimum message length of $\log_2 n$ bits.

The construction is easily seen to be polynomial time computable. □

Many practical problems concern the inference of parameters which are continuous rather than discrete, and for which both the prior density and the likelihood are mathematically well-behaved functions of the parameters. The above result can be used to show that SMML inference is still NP-hard in general for such problems.

COROLLARY 2. *If the parameter space is a continuum (and even if it must be $[0, 1]$), then SMML inference is NP-hard. Both the prior density and likelihood may be required to be infinitely differentiable.*

Proof. (Outline) We reduce from the problem we have just shown to be NP-hard: SMML with discrete parameter space Θ with n elements, likelihood function $f(x|\theta)$ taking just two values 0 and $1/3$ and uniform prior distribution $h(\theta) = 1/n$.

We construct a new, continuous parameter space Θ', a prior density h' on it and a likelihood f' such that SMML partitions and estimates for this new problem correspond to those for the original problem. The approach is to construct h' and f' by piecing together sufficiently well-behaved (and, in particular, infinitely differentiable) curves to reproduce the structure of the discrete problem in this continuous context.

Suppose $\Theta = \{\theta_1, \ldots, \theta_n\}$. Let $0, t_1, t_2, \ldots, t_n, 1$ be a strictly increasing sequence of members of $[0, 1]$ and let

ϵ be half the minimum distance between two consecutive members of this sequence. It would suffice to take $t_j = j/(n+1)$ for all j and $\epsilon = 1/(2(n+1))$. We intend that t_j corresponds to θ_j. Put $\Theta' = [0,1]$, so that $t_j \in \Theta'$ for all j.

Let $s : [0,1] \to [0,1]$ be an infinitely differentiable function which (a) is monotonic increasing, (b) has $s(0) = 0$ and $s(1) = 1$, (c) satisfies $s(1-\theta) = 1 - s(\theta)$ and (d) has all derivatives 0 at $\theta = 0$ and $\theta = 1$. s is thus an S-shaped curve with 180° rotational symmetry and very 'flat' ends so that copies of it may be pieced together properly.

Define the new prior h' on Θ' by:

$$\begin{array}{ll} \text{on } (t_j - \epsilon, t_j): & h'(t_j - \eta) = 1 - s(\eta/\epsilon)/n; \\ \text{on } (t_j, t_j + \epsilon): & h'(t_j + \eta) = 1 - s(\eta/\epsilon)/n; \\ \text{elsewhere:} & h'(\theta') = 0. \end{array}$$

Define $f'(x|\theta')$ for $x \in \mathcal{X}$, $\theta' \in \Theta'$ as follows:

$$\begin{array}{ll} \text{on } (t_j - \epsilon, t_j): & f'(x|t_j - \eta) = f(x|\theta_j)s(\eta/\epsilon); \\ \text{on } (t_j, t_j + \epsilon): & f'(x|t_j + \eta) = f(x|\theta_j)s(\eta/\epsilon); \\ \text{elsewhere:} & f'(x|\theta') = 0. \end{array}$$

It is routine to prove that our new, continuous SMML problem with prior density h' on $[0,1]$ and likelihood f' has all the required properties and in particular that SMML partitions of $\mathcal{X}$ for this new problem are precisely the SMML partitions of $\mathcal{X}$ for the original, discrete problem. □

6. CONCLUSIONS

SMML inference is NP-hard in general, easy for the binomial distribution and other one-dimensional cases, and of unknown complexity for the trinomial distribution. It raises interesting algorithmic and combinatorial problems. The unsolved trinomial case resembles a geometric packing problem and would appear to be especially worthy of further investigation.

ACKNOWLEDGEMENTS

We thank David Dowe for his comments. This work is supported in part by Australian Research Council Large Grant A49703170.

REFERENCES

[1] Bartoszyński, R. and Niewiadomska-Bugaj, M. (1996) *Probability and Statistical Inference*. Wiley, New York.

[2] Solomonoff, R. (1964) A formal theory of inductive inference I, II. *Inform. Control*, **7**, 1–22, 224–254.

[3] Wallace, C. S. and Boulton, D. M. (1968) An information measure for classification. *Comp. J.*, **11**, 185–194.

[4] Boulton, D. M. and Wallace, C. S. (1970) A program for numerical classification. *Comp. J.*, **13**, 63–69.

[5] Wallace, C. S. and Boulton, D. M. (1975) An invariant Bayes method for point estimation. *Classification Soc. Bull.*, **3**, 11–34.

[6] Wallace, C. S. and Freeman, P. R. (1987) Estimation and inference by compact coding. *J. R. Stat. Soc. B*, **49**, 223–265.

[7] Rissanen, J. (1976) Parameter estimation by shortest description of data. In *Proc. Joint Automatic Control Conf.*, 1976, pp. 593–597. American Society of Mechanical Engineers, New York.

[8] Rissanen, J. (1978) Modelling by shortest data description. *Automatica*, **14**, 465–471.

[9] Rissanen, J. (1987) Stochastic complexity. *J. Royal Statist. Soc. B*, **49**, 223–239, 252–265.

[10] Rissanen, J. (1989) *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore.

[11] Allison, L., Powell, D. R. and Dix, T. I. (1999) Compression and approximate matching. *Comp. J.*, **42**, 1–10.

[12] Allison, L. and Wallace, C. S. (1994) The posterior probability distribution of alignments and its application to parameter estimation of evolutionary trees and to optimization of multiple alignments. *J. Molec. Evol.*, **39**, 418–430.

[13] Allison, L., Wallace, C. S. and Yee, C. N. (1992) Finite-state models in the alignment of macromolecules. *J. Molec. Evol.*, **35**, 77–89.

[14] Dowe, D. L., Oliver, J. J., Allison, L., Wallace, C. S. and Dix, T. I. (1993) A decision graph explanation of protein secondary structure prediction. In *Proc. 26th Hawaii Int. Conf. on System Sciences (HICSS)*, Hawaii, January 1993, pp. 669–678. IEEE Computer Society Press, Los Alamitos, CA.

[15] Dowe, D. L. and Krusel, N. (1994) Decision tree models of bushfire activity. *AI Applic.*, **8**, 71–72.

[16] Kissane, D. W., Bloch, S., Burns, W. I., Patrick, J. D., Wallace, C. S. and McKenzie, D. P. (1994) Perceptions of family functioning and cancer. *Psycho-oncology*, **3**, 259–269.

[17] McKenzie, D. P., McGorry, P. D., Wallace, C. S., Low, L. H., Copolov, D. L. and Singh, B. S. (1993) Constructing a minimal diagnostic decision tree. *Methods Inform. Med.*, **32**, 161–166.

[18] Patrick, J. D. and Wallace, C. S. (1982) Stone circle geometries: an information theory approach. In Heggie, D. (ed.), *Archaeoastronomy in the Old World*, pp. 231–264. Cambridge University Press, Cambridge.

[19] Wallace, C. S. (1996) False oracles and Strict MML estimators, In Dowe, D. L., Korb, K. B. and Oliver, J. J. (eds), *Information, Statistics and Induction in Science: Proc. ISIS '96*, Melbourne, 20–23 August, 1996, pp. 304–316. World Scientific, Singapore. Earlier version (1989) *Technical Report 89/128*, Department of Computer Science, Monash University, Australia.

[20] Wallace, C. S. and Dowe, D. L. (1999) Minimum Message Length and Kolmogorov complexity. *Comp. J.*, **42**, 270–283.

[21] Garey, M. R. and Johnson, D. S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco.

[22] Shannon, C. E. (1948) A mathematical theory of communication. *Bell Syst. Tech. J.*, **27**, 379–423.