Refinements of MDL and MML Coding

C. S. WALLACE AND D. L. DOWE

Computer Science, Monash University, Clayton, Victoria 3168, Australia
Email: csw@cs.monash.edu.au

We discuss Rissanen's scheme of 'complete coding' in which a two-part data code is further shortened by conditioning the second part not only on the estimates, but also on the fact that these estimates were preferred to any others. We show that the scheme does not lead to improved estimates of parameters. The resulting message lengths may validly be employed to select among competing model classes in a global hypothesis space, but not to select a single member of the chosen class. A related coding scheme is introduced in which the message commences by encoding an ancillary statistic, and then states parameter estimates using a code conditioned on this statistic. The use of Jeffreys priors in MDL codes is questioned and the resulting normalization difficulties and violations of the likelihood principle are discussed. We argue that the MDL objective of avoiding Bayesian priors may be better pursued by other means.

1. COMPLETE CODING

Rissanen [1] has introduced a technique of 'complete coding' to reduce description length estimates when the description of the data $x \in X$ is based upon a two-part message form for a model class with a fixed number K of real-valued parameters, parameter vector $\theta \in \Theta$ and probability model $f(x \mid \theta)$. (Note that x is all the data available, and may typically be a vector of values, each i.i.d. from some unknown distribution.) The first part names a parameter vector $\hat{\theta}$ which is one of a discrete set Θ^* of possible vectors and the second part encodes the data x using a code based on the probability distribution $f(x \mid \hat{\theta})$. However, unlike the case in minimum message length (MML), the code for the second part is not a Shannon-optimal code for the distribution $f(x \mid \hat{\theta})$. Rather, the code is optimized for the truncated distribution

$$f_c(x \mid \hat{\theta}) = C_c\, f(x \mid \hat{\theta}) \quad \text{if } m(x) = \hat{\theta}$$
$$\text{or } f_c(x \mid \hat{\theta}) = 0 \text{ otherwise}$$

where $m(x)$ is an estimator function mapping data values onto Θ^*, and C_c is a normalization constant.

That is, $f_c(\cdot \mid \hat{\theta})$ is $f(\cdot \mid \hat{\theta})$ with support restricted to just those data values which will be encoded using the estimate $\hat{\theta}$ and renormalized. The result of the restriction and renormalization is that $C_c \geq 1$, so for all x which will be encoded using $\hat{\theta}$, $f_c(x \mid \hat{\theta}) \geq f(x \mid \hat{\theta})$. The inequality will be strict if there is any x such that $f(x \mid \hat{\theta}) > 0$ and $m(x) \neq \hat{\theta}$. As this will usually be the case, the second part code length $-\log f_c(x \mid \hat{\theta})$ will be less than the 'incomplete' code length $-\log f(x \mid \hat{\theta})$ used, for example, in MML codes.

Clearly, the effect of this use of 'complete codes' will usually be to shorten the length of the two-part description and is therefore regarded as preferable in minimum description length (MDL) work.

In discussing this idea, it will be useful to introduce a probability density $h(\theta)$ on Θ and the resulting marginal distribution

$$r(x) = \int h(\theta)\, f(x \mid \theta)\, \mathrm{d}\theta.$$

The density $h(\theta)$ plays the role of a 'prior', but in MDL work is usually derived from considerations other than 'prior belief' [2]. One definition of the stochastic complexity (SC) of data x with respect to the model class described above is $SC_1(x) = -\log r(x)$. It gives the length of a message which encodes x using a code based on the marginal distribution of data. (Here, and elsewhere unless noted, message lengths are given in nits, one nit = $\log_2(\mathrm{e})$ bits.) We will use the symbol $I_0(x)$ for $SC_1(x)$ and define its expectation

$$I_0 = -\sum_{x \in X} r(x) \log r(x).$$

A second definition of the SC of x with respect to the model class is based on a two-part coding as described above. In the original derivation, which did not employ 'complete codes', the complexity was defined by [3]

$$SC_2(x) = -\log q(m(x)) - \log f(x \mid m(x))$$

where $m(.)$ is an estimator function from X onto Θ^*, usually but not necessarily approximating the maximum likelihood estimator, and $q(\hat{\theta})$ defines a 'probability' distribution over the members of Θ^*. The members of Θ^*, i.e. the estimates which may be used in the first part of the message, were chosen to be spaced in Θ with a spacing δ (or a Voronoi region volume δ) so that the probability $q(\hat{\theta})$ associated with an estimate $\hat{\theta} \in \Theta^*$ is $q(\hat{\theta}) = \delta \times h(\hat{\theta})$. The optimum value of δ (which may vary in different regions of Θ) is then chosen by balancing the reduction in the first term of $SC_2(.)$ yielded by a large δ against the increase in its second

term produced by the fact that a large δ causes an increasing disparity between the maximum likelihood estimate for x and the estimate $m(x) \in \Theta^*$ which will be used in the encoding. The optimum δ turns out, for sufficiently regular model classes, to be proportional to $F(\theta)^{-1/2}$ where $F(\theta)$ is the Fisher information.

Thus far, this second derivation of $SC_2(.)$ closely parallels the commonest approximation to message length used in the MML school. The differences are confined to the choice of the density $h(.)$, which is regarded as a Bayesian prior in MML, and the question of whether δ and the estimator $m(.)$ should be chosen to minimize the 'expected' or the 'worst-case' message lengths. For any two-part coding scheme, we generically define $I_1(x)$ as the length of the encoding of x with respect to the model class, and I_1 as its expectation. Thus, for this derivation of SC, we have

$$I_1(x) = SC_2(x),\ I_1 = \sum_{x \in X} r(x) I_1(x).$$

In MML, $m(.)$ (and hence Θ^* and $\delta(\theta)$) are chosen to minimize the expectation I_1, but in MDL it appears commoner to minimize $\max_{x \in X} (I_1(x) - I_0(x))$. For sufficiently regular model classes, e.g. those in the exponential family, and for similar choices of $h(.)$, the differences between the MML and MDL $SC_2(.)$ approaches are numerically small. Note that in both cases, $I_1 \geq I_0$, but for some x, it is possible that $I_1(x) < I_0(x)$.

The introduction of complete codes changes the picture, and leads to a third definition of stochastic complexity:

$$SC_3(x) = -\log q(m(x)) - \log f_c(x \mid m(x)).$$

Clearly if complete coding is introduced to an $SC_2(.)$ coding scheme without altering the estimator, then δ, Θ^* and $q(.)$ remain unchanged and for all x, $SC_3(x) \leq SC_2(x)$, with strict inequality being the usual case. An improvement in message length, both expected and worst-case, is obtained. However, the balancing of first-part and second-part message lengths leading to the choice of δ is no longer valid.

Suppose that the original $SC_2(.)$ choice of δ is now everywhere halved. This will double the number of estimate values in Θ^* and halve every $q(\hat{\theta}) \in \Theta^*$, with the result that the first term in $SC_3(.)$ is increased by $\log 2$. But also, the number of data values mapping to each estimate is halved on average, because the same number of values in X are now distributed by the estimator mapping $m(.)$ over twice as many estimates. Hence, for each estimate $\hat{\theta}$, $\sum_{x:m(x)=\hat{\theta}} f(x \mid \hat{\theta})$ is typically halved, leading to a doubling of the normalization constant C_c and so a reduction in the second term of $SC_3(.)$ of at least $\log 2$. In fact, the reduction will typically be greater, because the reduced number of data values mapping to $\hat{\theta}$ may now be chosen to be those most probable under $\hat{\theta}$. That is, if $\hat{\theta}$ was an original member of Θ^*, it can now shed its worst-case data values to a newly-introduced estimate chosen to give these values higher likelihood. Both on average and in worst case, the product $C_c \times f(x \mid \hat{\theta})$ will at least double, and typically more than double, leading to a reduction in the second term of $SC_3(.)$ of more than $\log 2$. Overall, the effect of halving the original δ cannot increase the SC of any x and will typically decrease it, i.e. give a shorter description length.

If halving δ after introducing complete codes reduces $SC_3(x)$, halving it again may lead to further improvement. In fact, for all x, $SC_3(x)$ decreases monotonically as more estimates are added to Θ^*. The minimum expected and worst-case values of $SC_3(x) - I_0(x)$ are sure to be reached only when so many estimates are included in Θ^* that for all $\hat{\theta} \in \Theta^*$, the pre-image of $\hat{\theta}$ under $m(.)$ contains only a set of data values with identical sufficient statistics.

If complete coding is carried to this logical limit of increased efficiency, a little thought shows that, for the set of data values mapped by $m(.)$ to some estimate $\hat{\theta}$, the actual value of $\hat{\theta}$ has no effect on the value of $f_c(x \mid \hat{\theta})$ for any x in the set, provided only that $f(x \mid \hat{\theta}) > 0$. Since all x in the set have equal sufficient statistics, if this proviso is satisfied for any member of the set it is satisfied for all. Thus, the full-blooded adoption of complete coding for a two-part description, while giving an efficient code for the data, destroys the basis on which two-part coding can be claimed to yield good parameter estimates.

If the intention of introducing complete coding is simply to make $SC_3(.)$ a closer approximation to the original definition $SC_1(.) = I_0(.)$ than was $SC_2(.)$, it is clearly a useful advance, whether applied just once to the estimator derived using $SC_2(.)$, or carried to its logical limit. However, if the intent of two-part coding is to obtain good estimates of the parameters of the model, as well as a measure of complexity with respect to the 'class' of models, complete coding seems less advisable. Even if applied only once, retaining the estimator derived under $SC_2(.)$, its soundness as a component of a model-selection process must be questioned because, if unleashed, it would vitiate the estimator. Of course, MDL was designed to select a model class, not a fully-specified model, and the above objection is nugatory in this context. However, some users of MDL appear to regard complete coding as yielding good parameter estimates [4, 5, Section 4], and this view appears mistaken.

1.1. 'Complete coding' in MML

In the MML approach, the intention is to find the best 'fully-specified' model of the data within some set Θ of possible models, regardless of how this set is classified. Without here discussing whether this aim is worthy, we simply note that for the reasons stated above, complete coding as used in MDL would conflict with this aim. However, this does not mean that MML has nothing to learn from the technique. In thinking about this discussion of the differences between MML and MDL approaches to coding data, we realized that a technique closely related to complete coding could, at least in some cases (see Section 1.2), overcome a serious deficiency in MML. The idea is new to us, arising only after writing our main paper in this issue, and so our presentation should be regarded as tentative, but we hope still useful.

For a model class with K parameters, let S be the smallest

integer for which there exists an S-dimensional vector 'sufficient statistic', i.e. an S-dimensional vector function of the data which captures all the data information relevant to the parameters. Most applications of MML (and MDL) have used data models which could be decomposed into components each having a fairly simple probability model, commonly a member of the exponential family. For models in this family, $S = K$. For all such models that we have used, the standard MML two-part code has a high efficiency, both in expectation and in worst-case.

It has been shown [7] that if for any x the log-likelihood function has an approximately quadratic form about its maximum, and the prior density $h(.)$ is slowly varying, the MML construction yields a two-part code with

$$I_1 - I_0 < (1/2)\log(2\pi(K+1)) + 2.$$

Further, if $S = K$, it can be shown that for all x,

$$I_1(x) - I_0(x) < (1/2)\log(2\pi(K+1)) + 4.$$

Well-behaved models stay well within these bounds. For instance a binomial model of a Bernoulli trial with unknown success probability p, $h(p) = 1$, $N = 1000$, gives $I_1 - I_0 = 0.175\ldots$ and $I_1(x) - I_0(x) < 1.14$ for all success counts. Hence, the MML two-part code comes within a very few nits of the 'ideal' code length $I_0(x)$. As any improvement due to complete coding would be limited to these few nits at best, we considered the technique to be not worth the risk of prejudicing the MML parameter estimation.

Not all model classes are so well behaved and when $S > K$, it is possible for $I_1(x) - I_0(x)$ to be large in the worst case, even if $I_1 - I_0$, the expected inefficiency, remains small. We now illustrate with a simple example and show how a version of complete coding may dramatically improve the performance of MML.

1.2. An awkward example

Consider the model class of a uniform distribution of known range. The data x consists of N values $\{v_n : n = 1\ldots N\}$ i.i.d. with a uniform distribution between $\mu - 1/2$ and $\mu + 1/2$, so the range of the distribution is known to be one, and there is only $K = 1$ free parameter, the mean μ. Assume that μ is known to lie in some range of length $R \gg 1$ and assume it has a uniform 'prior' over this range. Let the data values $\{v_n\}$ be represented to a least count or measurement quantum of $a \ll 1/N$. We assume R so large and a so small that end effects near the ends of the possible range of μ may be neglected and sums replaced by integrals where convenient. Define for each x the sufficient statistics

$$s = v_{\max} - v_{\min}; \quad c = (v_{\max} + v_{\min})/2.$$

Then the probability model is

$$f(x \mid \mu) = a^N \text{ if } |\mu - c| < (1-s)/2$$

$$\text{or } f(x \mid \mu) = 0 \text{ otherwise.}$$

There is no smaller set of sufficient statistics, so $S = 2 > K = 1$.

For given μ, the joint density of (s, c) is

$$j(s, c \mid \mu) = N(N-1)s^{N-2} \text{ where } f(x \mid \mu) > 0.$$

This gives the marginal density of s as

$$g(s \mid \mu) = g(s) = N(N-1)(1-s)s^{N-2}.$$

As $g(s)$ is independent of μ, it gives the marginal density of s marginalized over c and μ. The overall marginal density of c is $w(c) = 1/R$. Marginalized over μ, the joint density of (s, c) is $g(s)/R$.

The problem presents no difficulty for the $SC_1(.)$ definition. Integration over μ gives

$$r(x) = (1-s)a^N/R,$$

$$SC_1(x) = I_0(x) = -\log r(x) = D - \log(1-s)$$

where $D = \log R - N\log a$. Integrating over the density $g(s)$ gives

$$I_0 = E(SC_1(x)) = D + \sum_{n=2}^{N} 1/n.$$

For large N, I_0 is approximately $D + \log N + \gamma - 1 = D + \log N - 0.422\ldots$.

The simple two-part $SC_2(.)$ construction and the standard MML approximation do badly on this problem. The Fisher information does not exist and the estimate set M^* must have members spaced by at most a, since for any x with sample range $s = 1$, $\hat{\mu}$ must exactly equal the sample midpoint c. If all members of M^* are coded similarly, the first part of the message has length $\log(R/a)$. If, as in these two methods, the second part codes the data using simply the distribution $f(x \mid \hat{\mu}) = a^N$, the total length is

$$I_1(x) = \log(R/a) - N\log a = D - \log a.$$

For $a \ll 1/N$, this value may grossly exceed $I_0(x)$ for almost all x.

To see the effects of 'complete coding' of the second part using this M^*, note that for given μ, the density of c is

$$N(1 - 2|c - \mu|)^{N-1} \text{ for } \mu - 1/2 \le c \le \mu + 1/2$$

and consider the obvious estimator $\hat{\mu} = c$. The density of c given $\mu = \hat{\mu}$ at $c = \hat{\mu}$ is N, so given that c is specified to least count a, $\text{Prob}(c = \hat{\mu} \mid \hat{\mu}) = aN$. Hence,

$$\begin{aligned} f_c(x \mid \hat{\mu}) &= f(x \mid \hat{\mu})/\text{Prob}(c = \hat{\mu} \mid \hat{\mu}) \\ &= a^N/(aN) = a^{N-1}/N. \end{aligned}$$

So, for the completed code, the message length $I_1(x)$ is

$$\begin{aligned} SC_3(x) &= \log(R/a) - \log(a^{N-1}/N) \\ &= \log R - N\log a + \log N = D + \log N. \end{aligned}$$

Hence, for large N, $I_1 \approx I_0 + 0.422\ldots$, so the code is quite efficient on average. However, it gives the same complexity for any x, so differs greatly from $SC_1(x) = I_0(x) = D - \log(1-s)$ for data with unusually large or small sample range. Rissanen's recent suggestion of a 'normalized maximum likelihood' (NML) measure of SC [6] in this issue does not cope well with this problem. For all $x \in X$, the maximized likelihood is the same, viz. a^N, so this measure will also give the same complexity to all data.

The strict MML construction (SMML), to which standard MML is an approximation, is a two-part code in which the code used in the first part to specify the estimate $\hat{\theta}$, is not based on a notional allocation to $\hat{\theta}$ of the 'prior' probability $\delta h(\hat{\theta})$ in some interval of Θ containing $\hat{\theta}$. Instead, $q(\hat{\theta})$ is taken as the marginal probability that x will lie in the pre-image of $\hat{\theta}$ under the mapping $m(.)$. That is, $q(\hat{\theta}) = \sum_{x:m(x)=\hat{\theta}} r(x)$. This construction achieves the shortest expected message length for any two-part code in which the second part codes x using a code length $-\log f(x \mid m(x))$. The SMML code for this example uses, perforce, a set M^* with spacing a, but assigns unequal code lengths to its members. An 'order 0' set of R estimates has integer estimate values, all with the same first-part code length and all x which can be coded using some member of this set are mapped to it. An 'order 1' set of R estimates has half-integer values, equal but slightly larger code lengths and is used for all x which can use these values and are not mapped into order 0 estimates.

Proceeding recursively, an 'order i' set of $2^{i-1}R$ estimates has estimate values odd multiples of 2^{-i} and is used for all possible x not already mapped into lower-order estimates. The recursion stops when all x are mapped or when $i > -\log_2 a$. It can be shown that the first-part code lengths increase with increasing i and that an estimate of order $i > 0$ will be used only (but not always) for data values x with $s > 1 - 2^{1-i}$.

This SMML code achieves good expected efficiency. For all $N > 1$, it gives $I_1 - I_0 < \log 2$, with equality being approached for large N. However, its worst-case efficiency is poor and unbounded as a approaches zero. For order $i > \log_2 N$, there is an x mapping to an order-i estimate with $I_1(x) - I_0(x) > 2(i - \log_2 N)\log 2$. Since i can reach $-\log_2 a$ for some x, $I_1(x) - I_0(x)$ can reach $-2\log(aN)$, which can be large since $a \ll 1/N$. If such data occurs, the SMML code length gives a serious over-estimate of the data's complexity with respect to the model class, which could distort a comparison with alternative model classes for the same data. Equally seriously, it can give very different complexities to two data vectors with the same sample range s but slightly different sample midpoints. If s is close to 1, the estimate $\hat{\mu}$ will be a low-complexity order-0 value if the midpoint c lies within $(1-s)/2$ of an integer value, but a high-order high-complexity estimate if c lies just outside this range.

The excessive SMML code length for some (admittedly improbable) data is caused by the 'incomplete' second-part codes used for such data. The data incurring an excessive cost are those x which have sample ranges s much closer to 1 than would be expected from the density $g(s)$, i.e. data with $(1-s) \ll 2/(N+1)$. When the SMML code maps such an x into a high-order estimate, the use of this estimate implies that $s > 1 - 2^{(1-i)}$, but the coding in the second part of the code makes no use of this knowledge. MDL-type complete coding using the SMML M^* and first-part coding would greatly increase the worst-case efficiency. The expected code length I_1 would also be decreased slightly, but would remain greater than I_0, because the completed second part code for data mapped to $\hat{\mu}$ is still based on a distribution $f_c(x \mid \hat{\mu})$ which is proportional to $f(x \mid \hat{\mu})$ and so implies a joint density for the statistics (s, c) proportional to $j(s, c \mid \hat{\mu})$ or s^{N-2} whereas the actual joint density for data in any region of (s, c) space (and hence in the pre-image of $\hat{\mu}$) is $g(s)/R$ proportional to $(1-s)s^{N-2}$. The discrepancy would cause a large difference between $SC_3(x)$ and $SC_1(x)$ for x with $(1-s) \ll 1/N$.

We propose instead of the MDL complete coding a three-part modification of the two-part MML code. The probability distribution $g(s)$ of the sample range s is independent of μ. Hence a message encoding x can begin by encoding s using a code optimized for the distribution $g(s)$. No knowledge of μ is needed to encode or decode this segment, length $-\log(ag(s))$. The message continues by stating an estimate $\hat{\mu}$, but now the code used can employ knowledge of s, since the receiver of the message will have this knowledge before having to decode the code for $\hat{\mu}$. In this example, once s is known, it is clear that $\hat{\mu}$ need only be coded with precision $(1-s)$. That is, we may now use an estimate set M_s^* with spacing $\delta_s = (1-s)$ and all estimates in the set can be given the same code length $\log(R/(1-s))$. Finally, a third message segment can encode x using the (unique) $\hat{\mu} \in M_s^*$ for which $f(x \mid \hat{\mu}) > 0$.

The code used in the third part will employ a kind of 'complete coding', using a code for the distribution conditioned on $\hat{\mu}$ and s, with length $-\log f(x \mid \hat{\mu}, s)$ where

$$\begin{aligned} f(x \mid \hat{\mu}, s) &= \text{Prob}(x, s \mid \hat{\mu})/\text{Prob}(s \mid \hat{\mu}) \\ &= f(x \mid \hat{\mu})/(ag(s)) = a^N/(ag(s)). \end{aligned}$$

Summing the lengths of the three parts gives

$$\begin{aligned} I_1(x) &= (-\log(ag(s))) + (\log(R/(1-s))) \\ &\quad + (-\log(a^N/(ag(s)))) \\ I_1(x) &= \log R - N\log a - \log(1-s) \\ &= D - \log(1-s) = I_0(x). \end{aligned}$$

Thus the new three-part MML code gives ideal efficiency. It does no violence to the intent of the MML approach, since it differs from two-part MML only in beginning the message with an aspect of the data which is independent of, and conveys no information about, the unknown parameter. All results about the consistency, invariance and efficiency of the MML method seem to apply to the three-part form. In particular, it better conforms to one of the guiding principles of the MML method, that parameters should be specified to a precision consistent with the expected error in their estimation.

1.3. Three-part MML coding

The three-part coding scheme developed above may have application to other model classes having $S > K$. Suppose we have a model class with probability model $f(x \mid \theta)$, data space X of N dimensions and parameter space Θ of $K < N$ dimensions and that $N \geq S > K$. There may exist an invertible mapping $t(x) = (y, z)$ for all $x \in X$ where y and z have dimension $L \leq (S - K)$ and $(N - L)$ respectively, and such that the probability distribution of y is independent of θ, say $g(y)$. Then the value $y = y(x)$ may be optimally coded as the first part of the message without any knowledge or estimate of θ. However, knowledge of y may assist in constructing an efficient code for an estimate $\hat{\theta}$ of θ, which is then stated in the second part of the message. Finally, the remaining detail of the data, namely the value $z = z(x)$, may be encoded in the third part, using a code based on the conditional distribution $f_z(z \mid \hat{\theta}, y)$ or $f_x(x \mid \hat{\theta}, y) = f(x \mid \hat{\theta})/g(y)$.

The statistic y is not informative about θ, but it may imply information about the shape of the likelihood function and thus assist in choosing an appropriate precision for stating the estimate $\hat{\theta}$. In the uniform example of 1.2, the sample range s played the role of y and while it said nothing about μ, it indicated the shape of the likelihood function and hence allowed the optimum Θ_s^* spacing $\delta_s = (1 - s)$ to be used in the second part of the message.

The statistic y, if it exists, appears to fit Fisher's definition of an 'ancillary' statistic.

The generality and utility of this three-part MML coding scheme remains to be examined in future work. While in the present example it achieves the ideal result $I_1(x) = I_0(x)$, this cannot be generally true. It is interesting that, while the scheme was inspired by Rissanen's 'complete coding', it is strangely different. In both forms, the final coding of the data uses a probability distribution conditioned on the estimate and on a function of the data. In Rissanen's form, the conditioning function is the estimator $m(x)$ which is highly informative about θ, whereas in our form, the conditioning function is an ancillary statistic which, by definition, is independent of and uninformative about θ.

2. SOME PROBLEMS WITH MDL

Earlier comparisons between earlier versions of MDL and MML are given in [3, 7] and the accompanying discussion in that 1987 issue of *J. R. Statist. Soc.*; and also in [8].

2.1. Partitioning models into 'model classes'

We see some difficulties in a program which aims to select a model class in that some sets of models permit several different plausible partitions into model classes. We provide two examples where the partition into model classes seems rather unclear: one being the family of computable functions and the other being the family of polynomial functions.

Is it legitimate to consider the class of functions, C_n, which can be described by n bits of input to a universal Turing machine (UTM) as a model class? If so, this definition and the corresponding model class structure will depend upon the choice of UTM, T. In general, for UTMs T_1 and T_2, C_{n-1}^{T1} is not contained in C_n^{T2} and C_{n-1}^{T2} is not contained in C_n^{T1}.

For the family of polynomials, do we (in the spirit of TMs) consider C_n to be the class of polynomials computable in n arithmetic steps? Or perhaps we consider C_n to be the polynomials of degree n. In the latter case, x^{23} would be considered less complex than x^{32}, whereas we would expect the reverse in the former case. Or perhaps we should consider C_n to be the class of polynomials with just n non-zero co-efficients. But, is such a definition with respect to the basis $\{1, x, x^2, \ldots\}$ or with respect to some orthonormal basis for a given scalar dot product? For rational polynomials, we might even consider C_n to be the class whose sum of integer coefficients in the numerator and denominator is equal to n.

In summary, it is not always clear how a well-defined family of models should be partitioned into classes.

2.2. Completing the code over the 'model class'

As described in Section 1, the idea behind complete coding [1] is that, given a model class, when we encode the data, we may assume the receiver of the message to know the estimator $m(\cdot)$. Hence, on receiving a message stating estimate $\hat{\theta}$, the receiver may deduce that the data x lies within the pre-image of $\hat{\theta}$ under m. So, in the second part of the message, based on the distribution $f(\cdot \mid \hat{\theta})$, all code words encoding data not in the pre-image of $\hat{\theta}$ may be eliminated, allowing a shortening of the remaining code words using the truncated distribution $f_c(\cdot \mid \hat{\theta})$ from Section 1. However, if we are to complete the code within model classes so that data is encoded using this truncated distribution, then we ask the question of why we do not complete the code over model classes within the family. In other words, when data is encoded as coming from a certain model class, why do we not encode this data conditional on its lying within the pre-image of that model class?

2.3. Jeffreys prior and normalized maximum likelihood

2.3.1. Interpretation of the Jeffreys prior

The Jeffreys prior depends upon the sensitivity of the measuring instruments and observational protocol used to obtain the data [9, 10, 11, p. 217]. While such a prior is clearly mathematically convenient, it is formally equivalent to using a Bayesian prior which will favour parameter values (or models) around the values where the measuring instruments are most sensitive. Consider, for example, an observer inside a circular train track trying to estimate the position of a train moving at fixed speed along the track. The Jeffreys prior advocates that if the observer is standing away from the centre of the circle nearer the track, then the observer should have a higher prior probability of the train being at a part of the track near the observer than at a corresponding length of track diametrically opposite. A

related example is given elsewhere [11, p. 217] regarding estimating the strength of a magnetic field.

From a Bayesian perspective, use of a Jeffreys prior is tantamount to saying that our prior belief about the strength of a field or the value of a parameter depends upon our measuring instrument and even upon its location, which is clearly rather silly.

2.3.2. Normalization of Jeffreys prior

Even from a non-Bayesian perspective, we have reservations about the use of a Jeffreys prior. What are we supposed to do in the situation when we cannot normalize the Jeffreys prior? The negative binomial distribution arises from a modification to the protocol in the binomial distribution—whereas the binomial distribution sees us take a fixed number of trials, the negative binomial distribution sees us sample until we have had a fixed number of successes and is but one distribution for which we are unable to normalize the Jeffreys prior.

2.3.3. Negative binomial distribution and Jeffreys prior

Given i.i.d. probability of binomial trial successes p, the negative binomial distribution for parameter n is the probability distribution of the number of trials, N, needed to get n successes and is written $N \sim Nb(n, p)$, with the geometric distribution, $G(p) = Nb(1, p)$, being a special case.

The negative log-likelihood function, L, is given by

$$\begin{aligned} L &= -\log f(N \mid n, p) \\ &= -\log \binom{N-1}{n-1} - n \log p - (N-n)\log(1-p). \quad (1) \end{aligned}$$

The Fisher information can be shown to be $n/(p^2(1-p))$, from which we obtain its square root, the Jeffreys prior, to be $\sqrt{n/(p^2(1-p))}$, which exceeds $1/p$ and therefore has an infinite integral and cannot normalize.

2.3.4. Negative binomial and normalized maximum likelihood

From equation (1), given n and N, the maximum likelihood (ML) estimate $\hat{p}_{\mathrm{ML}} = n/N$, which gives a maximum value of

$$f(N \mid n, \hat{p}_{\mathrm{ML}}) = \binom{N-1}{n-1} (n/N)^n \, (1-n/N)^{N-n}.$$

Choose $n_0 \geq n$ such that for all $N \geq n_0$ we have that $N!/(N-n)! \geq N^n/2$ and that $(1-n/N)^N \geq \mathrm{e}^{-n}/2$. Then

$$\begin{aligned} &\sum_{N=n}^{\infty} f(N \mid n, \hat{p}_{\mathrm{ML}}) \\ &\quad = \sum_{N=n}^{\infty} \binom{N-1}{n-1} (n/N)^n (1-n/N)^{N-n} \\ &\quad \geq n^n \sum_{N=n_0}^{\infty} n/N \binom{N}{n} (1/N)^N (1-n/N)^{-n} (1-n/N)^N \\ &\quad \geq n^n/n! \sum_{N=n_0}^{\infty} n/N \times 1/2 \times 1 \times \mathrm{e}^{-n}/2 \\ &\quad = \frac{n^{n+1}\,\mathrm{e}^{-n}}{4n!} \sum_{N=n_0}^{\infty} 1/N = \infty \end{aligned}$$

and so, for the negative binomial distribution, we are unable to do the normalization necessary for normalized maximum likelihood.

2.3.5. A mixed binomial protocol

Consider now a protocol of binomial trials in which our sampling experiment terminates when either 100 successes have occurred and been registered, or when 3000 trials have taken place, whichever comes first. The maximum likelihood can be normalized with this protocol limit on the number of trials, so NML is applicable as a measure of complexity. Suppose the experiment in fact concludes with 100 successes achieved in 469 trials, so we compute the NML complexity of this result. But just as we are about to send off the results for publication, our laboratory assistant informs us that he has found a design flaw in the experimental apparatus which, while not affecting the outcome of a trial, would inevitably cause irreparable damage to the equipment after just 500 trials and that our grant could not fund a replacement. So in fact, although we did not know it, our experiment was conducted under the protocol of 100 successes or 500 trials. Using these numbers gives a substantially different NML complexity. Which value is correct? And why should the non-event of equipment failure change anything much? And, indeed, were the number 3000 above changed to 300,000, the difference between the original and revised NML complexities would be even more substantial.

3. IS MDL MISDIRECTED?

There are two aspects of MDL with which we disagree, but will accept for the purposes of this section. The first is the reluctance of the MDL school to accept that any human investigator brings prior knowledge to every scientific study and the course of the work is informed by prior expectations. Indeed, we are born with strong prior expectations and would not survive without them. The second is the belief that the selection of a 'model class' for data is somehow more fundamental than and logically prior to the selection of a fully-specified model. A third aspect seen in the work of some followers of the school is the apparent belief that the best way to select a fully-specified model is first to select the model class and then to estimate its parameters by conventional means. This belief does not seem to be universally held and is certainly not an essential of the MDL program, so we will discuss it no further.

Accepting these two tenets, we may characterize the MDL program as we see it by the following points.

(i) We wish to select one of several parametrized model classes in the light of the data, but are uninterested in parameter values.

(ii) We will make no assumption of a Bayesian prior over the unknown parameters of a model class.
(iii) The selection of a class should be made by deriving a single number for each class and choosing the class with the smallest number.
(iv) The length of an efficient data description with respect to the class is a good choice for this number.

We may now ask whether the tools currently used in MDL are well suited to this program. It seems to us that there is reason for doubt.

First, several of the MDL tools, including the SC_1, SC_3 and NML measures, require for their application use of a normalized density measure over the parameter space of a class. Since no prior knowledge is allowed, this density must somehow be conjured from what is available, which is only the data space X, the parameter space Θ and the functional form of the class $f(x \mid \theta)$. The available choices seem limited. The 'Jeffreys prior' (if it exists at all) and the more general, but often similar, density implicitly defined by NML cannot be normalized for many simple and regular classes, as in Sections 2.3.3 and 2.3.4.

The response to this problem [6], namely artificially to restrict Θ to make normalization possible, seems arbitrary and distasteful. We are given no general guidance as to how this restriction should be done and different restrictions will give different results. If, as has been suggested, the restriction is made 'in the light of the data', we have no workable data description method at all, contrary to point (iv).

Second, any method reliant on a density on Θ constructed from X and $f(x \mid \theta)$ is likely to lead to a serious violation of the likelihood principle. Both the Fisher information on which the Jeffreys prior is based, and the normalization constant in NML, depend on the data space over which expectation or integration is performed. Thus, it becomes possible that exactly the same information may lead to different judgements depending on the observational protocol under which it was obtained. These MDL measures depend significantly on the data which was not obtained as well as on the data which was. (We note in passing that MML might be seen as equally in violation of the likelihood principle, but study of examples such as the binomial versus negative binomial comparison shows that its violations are insignificant [12].)

Finally, the SC_3 and NML methods involve an estimator, i.e. a selection of specific parameter values, which should be unnecessary in the light of point (i).

Let us take points (i), (ii) and (iv) seriously. An efficient encoding is one which exploits regularities in the data encoded. If we wish to encode our data with respect to some model class, we should aim to exploit only those regularities which we expect to find in the data by virtue of its being sourced from this class. If we encode our data with respect to the class of univariate normal distributions (with absolutely no prior assumptions about mean or variance), we should use those regularities arising solely from the 'Normal-ness' of the data and not from the nomination of any specific member of the class. Neither SC_1, SC_3 nor NML conforms to this obvious requirement.

It does not seem impossible to devise coding schemes based purely on the regularities to be expected of the class rather than of a specific member. However, it is in general the case that even when the data is a vector of i.i.d. values, the coding will not encode these individual values directly. Rather, the code will aim to provide efficient encodings of various functions of the data, the functions being chosen to exhibit the regularities of the 'class'. Necessarily, this means that the joint distribution of these functions must be independent of the parameter values. Only such distributions are characteristic of the class, rather than of some member model. For instance, if x is a set of N scalars $\{v_n : n = 1 \ldots N\}$, and the model class is that the scalars are i.i.d. according to some $f(v \mid \mu, \sigma)$ where μ is a location and σ a scale parameter, then the functions $u_n = (v_n - \bar{v})/\mathrm{Av}(|v_n - \bar{v}|)$ have distributions independent of the parameters and so can be efficiently coded assuming only the model class. Of course, these N functions are insufficient to reconstruct the data, since only $N - 2$ of them are independent. The MDL description would also have to include, e.g., the sample mean and standard deviation, and these could best be encoded using whatever 'code' or representation was used in the presentation of the raw data. We certainly have no warrant to suppose any better code.

We conclude that the MDL program (according to our understanding of its aims) might be better pursued by considering the coding of those functions of the data whose distributions are characteristic of the model class and independent of the parameters. This course would make it unnecessary ever to postulate densities over the parameter space, or equivalently marginal distributions over the data space. With them would go the ghost of the Reverend Bayes and any dependence on an estimator. We do not pretend that this course would be easy to follow. It is not obvious that the required number of functions of the data (ideally $(N - K)$) with a joint distribution independent of θ can easily, or always, be found.

The 'Predictive' coding technique described in [6] seems better adapted to the MDL program than the other methods mentioned, since it avoids definition of a density on Θ, but it still involves (admittedly ephemeral) estimates, which should not be needed by the program.

ACKNOWLEDGEMENTS

This discussion paper was supported by Australian Research Council (ARC) Large Grants A49330656, A49703162 and A49602504.

REFERENCES

[1] Rissanen, J. (1996) Fisher information and stochastic complexity. *IEEE Trans. Inform. Theory*, **42**, 40–47.
[2] Rissanen, J. J. (1989) *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore.

[3] Rissanen, J. J. (1987) Stochastic complexity. *J. R. Statist. Soc.*, B, **49**, 223–239.

[4] Dom, B. E. (1996) *MDL Estimation for Small Sample Sizes and its Application to Linear Regression*. Technical Report RJ 10030 (90526), IBM Almaden Research Division, CA, USA.

[5] Grünwald, P., Kontkanen, P., Myllymäki, P., Silander, T. and Tirri, H. (1998) Minimum encoding approaches for predictive modeling. In *Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI'98)*, Madison, Wisconsin, USA, pp. 183–192. Morgan Kaufmann.

[6] Rissanen, J. (1999) Hypothesis selection and testing by the MDL principle. *Comput. J.*, **42**, 260–269.

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

[8] Baxter, R. A. and Oliver, J. J. (1995) *MDL and MML: Similarities and Differences*. Technical Report TR 207, Department of Computer Science, Monash University.

[9] Lindley, D. V. (1972) *Bayesian Statistics, A Review*, p. 71. SIAM, Philadelphia, PA.

[10] Bernardo, J. M. and Smith, A. F. M. (1994) *Bayesian Theory*. Wiley, Chichester.

[11] Dowe, D. L., Oliver, J. J. and Wallace, C. S. (1996) MML estimation of the parameters of the spherical Fisher distribution. In *Proc. 7th Conf. Algorithmic Learning Theory (ALT'96), LNAI 1160*, Sydney, 1996, pp. 213–227. Springer-Verlag, Heidelberg.

[12] Baxter, R. A. (1996) The likelihood principle and 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. 292–303. World Scientific, Singapore.