Intrinsic classification by MML - the Snob program

Christopher S. Wallace and David L. Dowe,
Department of Computer Science,
Monash University, Clayton, Victoria 3168, Australia
e-mail: {csw,dld}@cs.monash.edu.au

Abstract: We provide a brief overview of Minimum Message Length (MML) inductive inference (Wallace and Boulton (1968), Wallace and Freeman (1987)). We then outline how MML is used for statistical parameter estimation, and how the MML intrinsic classification program, Snob (Wallace and Boulton (1968), Wallace (1986), Wallace (1990)) uses the message lengths from various parameter estimates to enable it to combine parameter estimation with model selection in intrinsic classification. We mention here the most recent extensions to Snob, permitting Poisson and von Mises circular distributions. We also survey some applications of Snob (albeit briefly), and further provide some documentation on how the user can guide Snob's search through various models of the given data to try to obtain that model whose message length is a minimum.

Keywords: Machine learning, mathematical foundations, classification, intrinsic classification, numerical taxonomy, clustering, unsupervised learning, Minimum Message Length, MML, Snob, induction, coding, statistical inference, information theory.

1. Introduction - About Minimum Message Length (MML)

The information-theoretic Minimum Message Length (MML) principle[23(p185),27] (and, e. g.[4,24]) of inductive inference variously states that the best conclusion to draw from data is the theory with the highest posterior probability or, equivalently, that theory which maximises the product of the prior probability of the theory with the probability of the data occuring in light of that theory. We quantify this immediately below.

Letting D be the data and H be an hypothesis (or theory) with prior probability Pr(H), we can write the posterior probability Pr(H|D) = Pr(H&D)/Pr(D) = Pr(H).Pr(D|H)/Pr(D), by repeated application of Bayes's Theorem. Since D and Pr(D) are given and we wish to infer H, we can regard the problem of maximising the posterior probability, Pr(H|D), as one of choosing H so as to maximise Pr(H).Pr(D|H) .

Also, elementary information-theoretic coding tells us that an event of probability p can be coded (e.g. by a Huffman code) by a message of length $-\log_2 p$ bits . (Negligible or no harm is done by ignoring effects of rounding up to the next positive integer.)

So, since $-\log_2 (\Pr(H).\Pr(D|H)) = -\log_2 (\Pr(H)) - \log_2 (\Pr(D|H))$, maximising the posterior probability, Pr(H|D), is equivalent to minimising $-\log_2 (\Pr(H)) - \log_2 (\Pr(D|H))$, the length of a two-part message conveying the theory and the data in light of the theory. Hence the name "minimum message length" (principle) for thus choosing a theory, H, to fit observed data, D . The principle seems to have first been stated by Solomonoff[19,p20], and was re-stated and apparently first applied in a series of papers by Wallace and Boulton[23(p185),3,4,5,6,7,24] dealing with model selection and parameter estimation (for Normal and multi-state variables) for problems of intrinsic classification. An important special case of the Minimum Message Length principle is an observation of Chaitin[8] that data can be regarded as "random" if there is no theory, H, describing the data which results in a shorter total message length than the null theory results in.

2. Parameter Estimation by MML

Given data $\underset{\sim}{x}$ and parameters $\underset{\sim}{\theta}$, let $h(\underset{\sim}{\theta})$ be the prior probability distibution on $\underset{\sim}{\theta}$, let $p(\underset{\sim}{x}|\underset{\sim}{\theta})$ be the likelihood, let $L = -\log p(\underset{\sim}{x}|\underset{\sim}{\theta})$ be the negative log-likelihood and let $F = E\left(\frac{\partial^2 L}{\partial \underset{\sim}{\theta} \partial \underset{\sim}{\theta}'}\right)$ be the Fisher information, the determinant of the (Fisher information) matrix of expected second partial derivatives of the negative log-likelihood. Then the MML estimate of $\underset{\sim}{\theta}$ is[27(p245)] that value of $\underset{\sim}{\theta}$ minimising the message length, $-\log\left(\frac{h(\underset{\sim}{\theta})p(\underset{\sim}{x}|\underset{\sim}{\theta})}{\sqrt{F(\underset{\sim}{\theta})}}\right)$ + a constant. (This is elaborated upon elsewhere[25(pp1–3)].)

The two-part message describing the data thus comprises first, a theory, which is the MML parameter estimate(s), and, second, the data given this theory. While it is reasonably clear to see that a finite coding can be given when the data is discrete or multi-state, we also acknowledge that all recorded continuous data must only be stated to finite precision by virtue of the fact that it was able to be (finitely) recorded. In practice, we assume that, for a given continuous attribute, all measurements are made to some precision, ε. For the Snob program (see Section 3), this precision is stated by the user. The precision should be a measure of the repeatability of a measurement. For a physical measurement, it is presumably the accuracy of the instrument being used. For a psychological experiment, it is (loosely speaking) how much the measured value would be expected to change if we had made the measurement yesterday or to-morrow rather than to-day.

2.1 Continuous Attributes

For a Normal distribution (with sample size, N), assuming a uniform prior on μ and a "colourless", scale-invariant, $1/\sigma$ prior on σ, we get that the Maximum Likelihood (ML) and MML estimates of the mean concur, i.e., that $\hat{\mu}_{MML} = \hat{\mu}_{ML} = \bar{x}$. Letting $s^2 = \sum_i (x_i - \bar{x})^2$, we get that $(\sigma^2)_{ML} = s^2/N$ and[23(p190)] that $(\sigma^2)_{MML} = s^2/(N-1)$ corrects this minor but well-known bias in the Maximum Likelihood estimate.

(In practice, the Snob program makes the reasonable assumption that $\sigma \geq 0.3\varepsilon$.) Snob assumes continuous attributes to come from a Normal distribution.

2.2 Multi-State Attributes

Since multi-state attributes are discrete, the above issues of measurement precision do not arise.

For a multi-state distribution with M states, a ("colourless") uniform prior is assumed over the (M-1)-dimensional region p(1) + p(2) + ... + p(M) = 1 .

Letting n(i) be the number of things in state m and N = n(1)+...+n(M), the MML estimate of p(m) is given[23:p187(4),pp191–194] by $\hat{p}(m) = (n(m) + 1/2)/(N + M/2)$.
This nominally gives rise to a (minimum) message length[23:p187(4),p194(28)] of $((M-1)/2).\log(N/12 + 1) - \log(M-1)! - \sum_m (n(m) + 1/2).\log \hat{p}(m)$ for both stating the parameter estimates and then encoding the things in light of these parameter estimates.

2.3 Circular (von Mises) Attributes

Earlier versions of Snob[23,20,22] permitted models of classes whose variables were assumed to come from a combination of either (discrete) multi-state or (continuous) Normal distributions. Since then, Snob has been augmented by permitting Poisson distributions and is currently being augmented to permit von Mises circular distributions[25,26].

The von Mises distribution (see, e. g.[10]), $M_2(\mu, \kappa)$, with mean direction μ, and concentration parameter, κ, is a circular analogue of the Normal distribution - both being maximum entropy distributions. Letting $I_0(\kappa)$ be the relevant normalisation constant, it has p.d.f. $f(x|\mu, \kappa) = \dfrac{1}{2\pi I_0(\kappa)} e^{\kappa . \cos(x-\mu)}$, and corresponds to the distribution of the angle, x, of a circular pendulum in a uniform field (at angle μ) subjected to thermal fluctuations, with κ representing the ratio of field strength to temperature. For small κ, it tends to a uniform distribution and for large κ, it tends to a Normal distribution with variance $1/\kappa$. Circular data arises commonly[10] in (e.g.) biology, geography, geology, geophysics, medicine, meteorology and oceanography; and we are currently involved in joint work using Snob to perform a cluster analysis of protein dihedral angles.

MML estimation of the von Mises concentration parameter, κ, is obtained by minimising the formula earlier in this section (plus a constant) for the message length, using[25] a uniform prior on μ in [0, 2π) and variously using the priors $h_3(\kappa) = \kappa/(1+\kappa^2)^{3/2}$ and $h_2(\kappa) = 1/(1+\kappa^2)$ on κ. Monte Carlo simulations[25,pp12–18] show a very impressive performance by the MML estimator. We are currently using the $h_3(\kappa) = \kappa/(1+\kappa^2)^{3/2}$ prior for Snob.

The contrast between MML and ML estimation is perhaps a little sharper for the von Mises distribution than it is for the Normal, multi-state and Poisson distributions. Regardless of how similar or otherwise the ML and MML estimates might or might not sometimes be, as we will elaborate upon in the next section, being able to associate a message length with each class enables us to use (the minimisation of) the message length as a natural metric for model selection.

3. Applying MML to Intrinsic Classification - the Snob Program

Given a set of objects or "things", it is a common problem to wish to intrinsically classify (or cluster) these things into would-be natural groupings (or "classes"). Such intrinsic classification can be thought of as concept formation, and (e.g.) common nouns within natural language presumably arose as a result of an intrinsic classification. The Snob program[23,20,22] is an application of MML to this problem of numerical taxonomy. Each continuous-valued attribute is assumed to come from a Normal distribution and each discrete attribute is assumed to come from a multi-state distribution. Snob uses MML for both the model selection (number of classes and assignment of things to classes) and parameter estimation (estimating means and standard deviations, etc.). Essentially, Snob tries to discern the structure in the data. Snob will prefer to hypothesise the existence of an additional class in the data precisely when the information cost of stating the parameter estimates for this additional class is more than offset by the information gain in stating the things assigned to this new class in terms of the newer, more appropriate, parameter estimates.

3.1 Stating the message

Following earlier work[23,20,22], we suppose the data (to be intrinsically classified) to be given as a matrix of D attribute values for each of N "things", with some attribute values possibly missing. Rather than use a hierarchic classification[5], we opt here for a "flat" classification.

The first part of the message, stating the hypothesis, H, comprises several concatenated message fragments, stating in turn:

a. The number of classes. (All numbers are considered equally likely, although this could easily be modified.)

b. The relative abundance of each class. (Creating names or labels for each class of length $-\log_2$ of the relative abundance, via a Huffman code, gives us a way of referring to classes later when, e.g., we wish to say which class a particular "thing" belongs to.)

c. For each class, the distribution parameters of each class (as discussed in Section 2). Each parameter is considered to be specified to a precision of the order of its expected estimation error or uncertainty (see, e. g.[25,pp3–4]). For a larger class, the parameters will be encoded to greater precision and hence by longer fragments than for a smaller class.

d. For each "thing", the class to which it is estimated to belong. (This can be done using the Huffman code referred to in b. above.)

Having stated in part 1 of the message above, our hypothesis, H, about how many classes there are and what the distribution parameters (μ, σ, etc.) are for each attribute for each class, in part 2 of the message we need to state the data in light of this hypothesised model.

The details of the encoding and of the calculation of the length of part 1 of the message may be found elsewhere[23]. It is perhaps worth noting here that since our objective is to minimise the message length (and maximise the posterior probability), we never need construct a message - we only need be able to calculate its length.

Given that part 1d. of the message told us which class each thing was estimated to belong to and that, for each class, part 1c. gives us the (MML) estimates of the distribution parameters for each attribute, part 2 of the message now encodes each attribute value of each thing in turn in terms of the distribution parameters (for each attribute) for the thing's class.

3.2 Stating the message more cleverly and more concisely - partial assignment

The form of message described in Section 3.1 implicitly restricted us to hypotheses, H, which asserted with 100% certainty which class each thing belonged to. Given that the population that we might encounter could consist of two different but highly over-lapping distributions, forcing us to state with conviction which class each thing belongs to is bound to cause us to mis-classify outliers from one distribution as belonging to another. In the case of two over-lapping (but distinguishable) 1-dimensional Normal distributions, this would cause us to over-estimate the difference in the class means and under-estimate the class standard deviations.

If what we seek is a message which enables us to encode the attribute values of each thing as concisely as possible, then a probabilistic (or partial) assignment of things to classes will enable us to produce a shorter message than that of Section 3.1. The reason for this is that[20,Section3;22,p77] if p(j,x), j=1, ..., J, is the (prior) probability of class j generating datum x, then the total assignment of x to its best class results in a message length of $-\log(Max_j\, p(j,x))$ to encode x whereas, letting $P(x) = \sum_j p(j,x)$, a partial assignment of x having probability p(j,x)/P(x) of being in class j results in a shorter message length of $-log(P(x))$ to encode x. These are the underlying ideas behind the partial assignment discussions in earlier work[20,Section3;22,Section3].

The other issue that arises is how to successfully carry out a coding trick to take advantage of this. If the outcomes of any random process are encoded using a code that is optimal for that process, the resulting binary string forms a completely random process[27,p241]. Since our Minimum Message Length theory is (by definition) optimal, our message (if it were to be constructed) would be a completely random string. Starting at thing N and reading the message backwards would give us a way of (pseudo-)randomly assigning data things x to class j with probability $p(j, x)/P(x)$. Using the weights $p(j, x)/P(x)$, which are available during iterations of Snob, we can safely partially assign x to the various classes since this turns out to have the same expected message length as the (pseudo-)random assignment just described[20,Section3]. Our message thus consists of a first part, the theory, describing the number of classes and the parameter estimates for each class, and a second part which optimally encodes the data given this theory.

4. Statistical consistency of estimates

The quotation[27,p241] above and the fact that general MML codes are (by definition) optimal implicitly suggest that, given sufficient data, MML will converge as closely as possible to any underlying model. Indeed, MML can be thought of as extending Chaitin's idea of randomness[8] to always trying to fit given data with the shortest possible computer program (plus noise) for generating it. This general convergence result for MML has been explicitly re-stated elsewhere[21,1].

The problem of model selection and parameter estimation in intrinsic classification can, at its worst, be thought of as a problem for which the number of parameters to be estimated grows with the data. It is well known[13] that Maximum Likelihood can come unstuck with such problems.

Lastly, without using the partial assignments described in Section 3.2, estimation would be guaranteed to be weakly inconsistent. This presents no problem if the underlying classes are sufficiently well-separated, but[20,Section3] if the means of two Normal distributions are separated by less than 2.5 times the true standard deviation of each component, the shortest (and also maximum likelihood) explanation would be incorrectly given by a 1-component model, no matter how large the data sample.

An extensive discussion of alternative algorithms for intrinsic classification has been given by Boulton[2], and a more recent discussion by Wallace[22,pp78−80].

5. Using the program, interpreting the output

Once the program has been compiled and the input file has been correctly formatted (and named), Snob can be used. At the completion of any iteration cycle, Snob will have a hypothesised model consisting of classes with various parameter estimates for the various attributes, and (as from Section 3.2) a partial assignment of things to classes. For any given class and attribute, one could code the value of this attribute for each thing in the class either using the population estimate(s) for this attribute's parameter(s) or by first stating class-specific estimates and then coding the values using these class-specific estimates. The amount by which the latter code is shorter than the former is deemed to be the significance of the attribute in the class.

For each (sufficiently large) class (or "mainclass"), Snob will store two sub-classes, which it will modify at regular intervals, in the hope of being able to reduce the mesage length by splitting a class into its subclasses. If its current model contains at least two mainclasses, then Snob will also carry a candidate join class, which it will also modify at regular intervals, as it also looks to reducing the message length by combining two mainclasses. For each class, Snob will report facts including its size, its relative abundance, its age (the number of iteration cycles since its creation), significance of each attribute and parameter estimates. If an attribute is deemed insignificant for a class (as discussed above), Snob will use the population estimates of the relevant parameters.

Typing "help" brings up a menu of options, some of which are discussed in[20,Section5].

"adjust n", for n a positive integer, asks Snob to carry out n iteration (or "adjust") cycles. During an adjust cycle, Snob will try to split or combine classes as discussed above. Also, given parameter estimates, Snob will make a partial assignment of things to classes. This partial assignment will give rise to new parameter estimates for the next iteration cycle until the estimates stabilise and a local minimum of the message length is found. "adjust" will stop after about 30 iteration cycles without improvement.

Typing "split" or "spliton" can induce a variety of class splits.

"wipe" can destroy one class or all classes.

"random" creates random classes, and can be seeded with "seed".

"sum", "pratt" and "prclas" are some of the reporting options, and "repatt" and "repclas", etc. write to files.

Also, a list of Snob commands can be automated by storing them in a file and using the "file" command. Letting the file "cycle" be the following list of commands, one per line (delineated here by commas to save space) : "random 7, adjust 1000, split 0, adjust 1000, random 6, adjust 1000, split 0, adjust 1000, random 5, adjust 1000, split 0, adjust 1000, random 9, adjust 1000, split 0, adjust 1000, random 8, adjust 1000, split 0, adjust 1000, setcross best, adjust 1000, file cycle", typing the command "file cycle" will commence execution of this list. Since the last command is "file cycle", the list of commands will continually repeat.

One hopes that, given the various random starting points, a global minimum to the message length will eventually be found.

In the input data, it is possible to declare either attributes or things or both to be "inactive". An inactive thing will not affect the classification or the message length, but will be (partially) assigned to its optimal class(es). An inactive attribute also has no affect on the classification or the message length, but the user can determine the parameter distributions for such attributes over the various classes. Digressing, observations of attribute values for things can also be input as missing - Snob is indeed still able to execute its message length calculations in this case.

6. Applications

Earlier applications of Snob include several to medical data[18,12,29,16,17,11]. Surveys of Snob applications to other data (such as[23,9,14]) are given in[15,29], the former survey being extensive and the latter survey providing an update. A study of families with a parent terminally ill with cancer[11] led to clusters of family members based on a response to several questionnaires. In ongoing joint work using data obtained post-bereavement, it appears that the message length of fitting the data several weeks post-bereavement given the optimal pre-bereavement cluster (i.e. the null theory) is longer than that of the MML post-bereavement model. This nominally suggests that, in this study, the post-bereavement data is significantly different from the pre-bereavement model. Here, a difference of more than 5 to 6 bits[27,p251] or of more than 10 bits[20] might be deemed to be statistically significant under certain modelling conditions.

The Poisson module has been used to analyse word-count data in 17th Century texts. The von Mises module seems to be accurately able to discriminate between pseudo-randomly generated classes from different von Mises distributions, and is currently finding clusters in data of several thousand sets of protein dihedral angles currently being analysed in joint work.

7. Notes on further work and program extensions

The theory behind MML single linear factor analysis[28] and multiple linear factor analysis (in preparation) has been completed. The program currently implicitly assumes that variables are uncorrelated and does not yet use the MML factor analysis[28]. Where there is correlation, linear factor analysis (which permits axis rotation) should enable the data to be better compressed.

The search for the minimum in the message length currently uses a (slightly conservative) greedy algorithm, only choosing to split or combine when a saving in message length can be guaranteed. With the message length as the objective function, using simulated annealing as a heuristic should accelerate the search.

It would not be too difficult to permit the user to modify the colourless priors (see Section 2) used by Snob to better represent the user's prior beliefs (or knowledge, or bias). However, in order that the classification obtained by the user might be better defended against disputes, it seems somewhat safer and perhaps wiser that the prior assumptions used by Snob be as colourless as possible.

8. Availability of the Snob program

The current version of the Snob program (written in Fortran 77 and complete with detailed but slightly out-of-date documentation file, snob.doc) is freely available for not-for-profit, academic research, and not for re-distribution, from C.S. Wallace. Published or otherwise recorded work using Snob should cite Wallace[22] and Wallace and Boulton[23].

Acknowledgements: This work was supported by Australian Research Council (ARC) Grant A49330656 and ARC 1992 small grant 9103169 .

References:

[1] A. R. Barron and T. M. Cover, Minimum Complexity Density Estimation, *IEEE Transactions on Information Theory*, **37**, 1991, pp1034-1054.
[2] D. M. Boulton, The Information Criterion for Intrinsic Classification, Ph. D. thesis, Dept. of Computer Science, Monash University, Australia, 1975.
[3] D. M. Boulton and C. S. Wallace, 'The Information Content of a Multistate Distribution', *J. Theoret. Biol.*, **vol 23**, 1969, pp269-278.
[4] D. M. Boulton and C. S. Wallace, 'A Program for Numerical Classification', *The Computer Journal*, **vol 13, no 1**, 1970, pp63-69.
[5] D. M. Boulton and C. S. Wallace, 'An Information Measure for Hierarchic Classification', *The Computer Journal*, **vol 16, no 3**, 1973, pp254-261.
[6] D. M. Boulton and C. S. Wallace, 'A Comparison Between Information Measure Classification', ANZAAS Congress, Perth, August 1973.
[7] D. M. Boulton and C. S. Wallace, 'An Information Measure for Single-Link Classification', *The Computer Journal*, **vol 18, no 3**, August 1975, pp236-238.

[8] G. J. Chaitin, On the length of programs for computing finite sequences, *J. Assoc. Comp. Mach.*, **13, 4**, 1966, pp547-549.
[9] Y. H. Chong, B. Pham, M. Manton and A. Maeder, Automatic nephanalysis from infrared GMS data, Tech Rept #89/125, Dept of Computer Science, Monash University, Australia, 1989.
[10] N. I. Fisher, 'Statistical Analysis of Circular Data', Cambridge Univ. Press, Cambridge, 1993.
[11] D. W. Kissane, S. Bloch, W. I. Burns, J. D. Patrick, C. S. Wallace and D.P. McKenzie, "An empirical study of family functioning and cancer", submitted, to appear.
[12] C. R. Latimer, "Eye-movement data: cumulative fixation time and cluster analysis", *Behav. Res. Meth. Instr. Computers* **20**, 1988, pp437-470.
[13] J. Neyman and E. L. Scott, Consistent estimates based on partially consistent observations, *Econometrika*, **16**, 1948, pp1-32.
[14] E'. Papp, D. L. Dowe and S. J. D. Cox, "Spectral classification of radiometric data using an information theory approach", *Proc. Advanced Remote Sensing Conference*, **Vol 2**, UNSW, Sydney, July 1993, pp223-232.
[15] J. D. Patrick, 'Snob: A program for discriminating between classes', Tech Rept #91/151, Dept of Computer Science, Monash University, Clayton, Australia, 1991.
[16] I. Pilowsky and M. Katsikitis, The classification of facial emotions : a computer-based taxonomic approach, *Journal of Affective Disorders*, **30**, 1994, pp61-71.
[17] I. Pilowsky and M. Katsikitis, A classification of illness behaviour in pain clinic patients, *Pain*, **57**, 1994, pp91-94.
[18] I. Pilowsky, S. LeVine and D.M. Boulton, The classification of depression by numerical taxonomy, *British Journal of Psychiatry* **115**, 1969, pp937-945.
[19] R. Solomonoff, A formal theory of inductive inference I, II. *Information and Control*, **7**, 1964, pp1-22 and pp224-254.
[20] C. S. Wallace, 'An Improved Program for Classification', *9th Australian Computer Science Conference (ACSC-9)*, **vol 8, no 1**, 1986, pp357-366.
[21] C. S. Wallace, 'False Oracles and SMML Estimators', Technical Report #89/128, Department of Computer Science, Monash University, Australia, June 1989.
[22] C. S. Wallace, 'Classification by Minimum-Message-Length Inference', S.G. Akl et al (eds.) Advances in Computing and Information - ICCI'90, Niagara Falls, *LNCS* **468**, Springer-Verlag, 1990, pp72-81.
[23] C. S. Wallace and D. M. Boulton, 'An Information Measure for Classification' *Computer Journal*, **Vol.11, No.2**, 1968, pp185-194.
[24] C. S. Wallace and D. M. Boulton, 'An Invariant Bayes Method for Point Estimation', *Classification Society Bulletin*, **vol 3, no 3**, 1975, pp11-34.
[25] C. S. Wallace and D. L. Dowe, "MML estimation of the von Mises concentration parameter", Tech Rept #93/193, Dept of Computer Science, Monash University, Australia, 1993.
[26] C. S. Wallace and D. L. Dowe, "Estimation of the von Mises concentration parameter using Minimum Message Length", *Proc. 12th Australian Statistical Society Conference*, Monash University, Melbourne, Australia, 1994.
[27] C. S. Wallace and P. R. Freeman, 'Estimation and Inference by Compact Coding', *Journal Royal Statistical Society, Series B, Methodology*, **49, 3**, 1987, pp240-265.
[28] C. S. Wallace and P. R. Freeman, 'Single Factor Analysis by MML Estimation', *J.R. Statist. Soc. B*, **54, No.1**, 1992, pp195-209.
[29] J. D. Zakis, I. Cosic and D. L. Dowe, "Classification of protein spectra derived for the Resonant Resonant Recognition model using the Minimum Message Length principle", *17th Australian Computer Science Conference (ACSC-17)*, Christchurch, NZ, Jan 1994, pp209-216.