Intrinsic Classification of Spatially Correlated Data

C. S. WALLACE

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

Intrinsic classification, or unsupervised learning of a classification, was the earliest application of what is now termed minimum message length (MML) or minimum description length (MDL) inference. The MML algorithm 'Snob' and its relatives have been used successfully in many domains. These algorithms treat the 'things' to be classified as independent random selections from an unknown population whose class structure, if any, is to be estimated. This work extends MML classification to domains where the 'things' have a known spatial arrangement and it may be expected that the classes of neighbouring things are correlated. Two cases are considered. In the first, the things are arranged in a sequence and the correlation between the classes of successive things modelled by a first-order Markov process. An algorithm for this case is constructed by combining the Snob algorithm with a simple dynamic programming algorithm. The method has been applied to the classification of protein secondary structure. In the second case, the things are arranged on a two-dimensional (2D) square grid, like the pixels of an image. Correlation is modelled by a prior over patterns of class assignments whose log probability depends on the number of adjacent mismatched pixel pairs. The algorithm uses Gibbs sampling from the pattern posterior and a thermodynamic relation to calculate message length.

1. INTRODUCTION

A basic learning problem is discovering the class structure of a population by observation of some of its members. This process underlies the invention of language and of all scientific theory, since it is possible to talk about a group of similar things only after the group has been recognized as important and a word or concept agreed to identify it.

In many cases, the things observed, from which a classification is to be induced, are observed in no particular order and have no important spatial relationships. However, in some domains, the things to be classified have a known spatial arrangement and there may be prior grounds to expect that the classes of neighbouring things are likely to be correlated. Traditional methods for intrinsic classification are not designed to detect or use such correlation.

In this paper, we describe two rather different methods for modelling class correlation. Both derive from the minimum message length method first described by [1], but use different means to incorporate a correlation model into a classification model. In both cases, significant extensions to the information measure calculations of the original method are needed.

2. MML CLASSIFICATION

The minimum message length (MML) approach to intrinsic (or unsupervised) classification introduced by [1, 2] views the problem of deciding whether a sample of things may usefully be regarded as a mixture of several different 'classes' of thing as a data-compression problem. Given data comprising attribute values for each thing, it considers a coding of this data based on the hypothesis that there are K different classes represented in the sample. The coded message first states K, then for each of the K classes specifies its relative abundance and the parameters of a simple model for the probability distribution of attribute values within the class. This first part of the message represents a hypothesized mixture model for the attribute values distribution for the whole population.

The second part of the message has a section for each thing in the sample. The section first states the hypothesized class of the thing, then gives its attribute values using a Huffman or similar code chosen to be optimal (in the sense of least expected length) given the parameters of the attribute distribution for the named class, as stated in the first part of the message.

The MML classification algorithm searches for that number K and those within-class distribution parameters and assignment of things to classes which jointly minimize the message length. In so doing, it is not necessary actually to construct the message, one need only calculate its length. The length of the second part is essentially the negative log of the probability of the class assignments and data given the population model. The coding, and hence length, of the first part requires adoption of a prior density over the parameters of the model and determination of the best precision to which these parameters should be stated. Its length may be taken as the negative log of the prior probability of the population model. Thus the total length is the negative log of the joint probability of population model, assignment of things to classes and data. Its minimization is equivalent to choosing the model and assignment of highest posterior probability, where the choice is made among a set of possibilities discretized so that no two possible choices are so similar that the data could not be expected to distinguish them.

Since its introduction, the algorithm has been refined and extended in various ways. A hierarchic specification of classes was introduced by [3], unfortunately in a way which made it impossible to remove a weak inconsistency in the estimates. An economy in the coding of assignments of things to classes introduced by [4] removed this weak inconsistency in the class parameter estimation for non-hierarchic classifications. Besides the real-valued and unordered discrete attribute types catered for originally, Poisson and von Mises distributions for within-class attribute distribution models were added by [5]. The very similar Autoclass algorithm developed by [6] has likewise been refined and extended to provide for within-class correlation among attributes and a limited form of class hierarchy. Both the MML algorithm (called 'Snob') and Autoclass have been used successfully in a wide range of domains.

Some theoretical grounds for using an MML approach to classification (and other model-selection and estimation problems) have been given by [7, 8], and other researchers. Autoclass is justified by Bayes-evidence arguments, but a numerical approximation used in the algorithm in fact makes it closer to MML in operation.

2.1. The core algorithm

The algorithms used by Snob and Autoclass have a two-phase core iterative procedure. Given some class model of the population, i.e. given K and the relative abundances and attribute distribution parameters for each class, the algorithm looks at each thing in turn and computes its probability of being found as a member of each class. That is, it computes the posterior distribution of the thing's class. The thing is then assigned to a class and its attribute values contribute to statistics gathered for the class. When all things have been looked at, the algorithm re-estimates the relative abundances and attribute distribution parameters for each class from the statistics gathered from the things assigned to it and begins another iteration.

The earliest version of Snob assigned each thing to the class of highest posterior probability for that thing. Later, it was realized that the message could actually be made shorter by employing any uncertainty in the classification of the thing to convey advance information about the next thing. This trick has the effect that a thing is partially assigned to all classes, in proportion to their posterior probability, and so contributes with partial weight to the accumulated statistics for every class. Autoclass uses the same partial-assignment, if with different motivation. If, however, each thing is assigned wholly to a single class chosen at random from the posterior distribution over classes for that thing, the expected contribution of the thing to the statistics of each class is the same as for partial assignment. In fact, the more economical coding of classes now used in Snob has this behaviour: a thing is assigned to a class by pseudo-random choice from the posterior, the bits encoding the next thing being used as a source of pseudo-random digits. Partial assignment is used to reflect the expected consequences of this behaviour without having to construct the coded message in full detail.

If random assignment is used in the iteration, the iteration will not converge, but will eventually generate in its successive steps a sample of classifications drawn randomly from the posterior over classifications given the entire data set. Such a random algorithm is a form of Gibbs sampling of plausible classifications.

2.2. Determining the number of classes

The core iteration, whether using partial, pseudo-random or random assignment, may effectively reduce K by reducing the estimated abundance of some class to zero, but cannot increase K. Snob uses a heuristic to detect when a class may be split in two with a profitable reduction in message length, and also to detect profitable class mergers. Autoclass relied on starting the iteration with an excessive number of classes and allowing some of them to be eliminated by the core iteration. Neither algorithm is guaranteed to find the globally optimum model. Davidson (private communication) and [9] have demonstrated extensions of the Gibbs-sampling random-assignment version of the core algorithm which can randomly sample K in proportion to its posterior distribution, despite the change in population model dimensionality with changing K. In principle, given sufficient time, these methods are sure eventually to find the optimal model and to sample from its neighbourhood, but as yet neither has been demonstrated on problems of realistic size.

3. SPATIAL CORRELATION

In the classification algorithms described earlier, the 'things' to be classified are viewed as having been drawn randomly from some underlying population. Although the data are presented (and in MML encoded) thing by thing as an ordered sequence, the order is assumed to be an irrelevant by-product of the sampling process.

This work concerns cases when the 'things' have a spatial ordering which may be expected to be relevant to their classification and how the classification algorithms may be modified to accommodate this expectation. Two problems will be discussed, one with a one-dimensional ordering of things, the other with two-dimensional ordering. In both cases, the work described is intended merely to demonstrate that the message length criterion can be applied to the intrinsic classification of spatially ordered data. The two-dimensional work is in a very early stage and would require much more development to realize the full potential of the MML approach.

Note that spatial correlation is here introduced by expecting and exploiting short-range spatial correlation among the classes of neighbouring 'things'. An alternative is to introduce spatial correlation through the statistical model used for the within-class distribution of attribute values by making the model distribution for the attributes of a pixel conditional on the attributes of its neighbours. This is not the kind of model considered here.

4. THE SECONDARY STRUCTURE OF PROTEINS

The work on this problem was done by Edgoose *et al.* [10], and is only briefly discussed here.

In this problem, the 'things' are sugar units on the backbone of a protein molecule (or sequence of molecules) and are naturally ordered in their linear sequence along the molecule. The attributes of a unit are two 'dihedral' angles between chemical bonds of the unit and the type of amino acid attached to the unit. It is hoped that an intrinsic classification of the units will yield classes corresponding roughly to the type of secondary structure in the vicinity of the unit, for example, helical, beta sheet, turn, etc.

The number of classes and their relative abundances are to be discovered from the data. However, there are good grounds for expecting the classification of one unit to be correlated with the classes of its neighbours. For instance, if the classes inferred indeed corresponded to secondary structure classes, we would not expect to find a 'helical' unit in the middle of a string of 'beta sheet' units: a single turn of a protein alpha helix takes several units. Similarly, isolated 'sheet' units would be unexpected, as would long strings of 'hairpin bends'.

The exact nature of the correlations however is not assumed *a priori*, as we do not wish to guide or bias the class discovery towards the usual secondary-structure nomenclature. The aim was to include in the model a fairly general model of neighbour dependence, leaving the message-length minimization to determine just what sort of dependence was found in the data. The model class assumed was a Markov model of order zero or one. That is, the prior probability that the next unit would be in class C could be taken as simply the relative abundance of class C (order zero) or from a conditional distribution over classes conditioned on the class of the previous unit (order one).

There is an obvious asymmetry in this model: taking the units in reverse order gives a different class of models. However, proteins are asymmetric: there is a front end and a back end and in living organisms they are usually assembled by adding units from the front to the back. The asymmetry in the Markov model is thus not unreasonable.

4.1. The protein sequence algorithm

The Edgoose algorithm for this problem is an extension to the current Snob partial-assignment algorithm. It deals with one order of class correlation at a time, and separate runs are done for orders zero and one, the order giving the shortest message length being preferred. The order-zero run just uses the normal Snob iteration, since no correlation model is assumed.

For non-zero Markov order, the class relative-abundance distribution of the uncorrelated model is replaced by a set of distributions, being the distributions conditioned on the class of the previous unit. During the first phase of the core iteration, the cost of encoding a site as a member of each class and hence the posterior distribution over classes for this unit, used the appropriate conditional abundance distribution as the prior distribution over the classes. However, the unbiased assignment of a particular thing to a class requires the influence of this choice on the coding of subsequent things to also be considered. A simple dynamic programming algorithm (DPA) can calculate the message length of stating the entire sequence conditional upon the assignment of any one thing to a particular class. Abundance statistics are accumulated conditioned on this unbiased assignment. In the second phase of the iteration, the conditional distributions are re-estimated from their corresponding conditional abundance statistics.

Note that, unlike ordinary Snob, the order in which things are treated in the first phase is important and the cost of encoding each observation conditional upon each class must be stored in memory for later traversal by the DPA.

This description conceals a complication. Having calculated the posterior probability of a unit belonging to each class, the core iteration makes a partial assignment of the unit to each class with weight proportional to class posterior probability, as in Snob. This involves nothing new in the accumulation of the statistics from which class distribution parameters will be estimated. However, in adding these weights to the conditional abundance statistics, the conditioning class, i.e. the class of the previous unit, is not known exactly, as this unit has also been partially assigned. Thus, the 'weights' for the current unit contribute partially to the conditional abundance statistics for all conditions, i.e. all possible classes of the previous unit.

The upshot is that the program needs to keep track of K Markov chains simultaneously, involving some rather complicated bookkeeping.

Further bookkeeping complications are required to implement the Snob class-splitting and class-merging heuristics, but these raise no new problems of principle.

4.2. Summary of results

A dataset containing over 41,000 such sugar units was modelled using the first order Markov classification model with the best class structure describing 19 types of thing. This model achieved compression of 6.40 bits/thing on this noisy data which represents a saving of 0.66 bits/thing over the zero-order model and 2.56 bits/thing over the one-class model.

As expected the class structure correlated well with the common helix, beta-sheet and turn classification with three sub-classes of helix and two sub-classes of beta-sheet. The remaining classes described various varieties of turn structure. This 19 class classification found is considerably simpler than competing unsupervised classification models [11, 12] and hence more amenable to further analysis and model building in this area.

Of particular interest was the second-most abundant class

(12% of the data) which described a turn structure with large variance and an expected segment length of about five things. It seems that the specific conformation of residues in this class is most likely completely determined by non-local structural effects. This surprising result is a good example of unexpected structure emerging from the data.

5. INTRINSIC CLASSIFICATION OF IMAGE PIXELS

The second case to be discussed is the classification of things arranged in a regular square grid in two dimensions. An example is the pixels of a multi-spectral image, where each pixel has as attributes several intensities at different wavelengths. We will henceforth use this example.

Consider a multi-spectral image of part of the earth. *A priori*, we expect to find spatial clusters or regions of similar pixels: a patch of water, a patch of forest, patches of cultivated fields, etc. That is, we expect the class of a pixel to be positively correlated with the class of its neighbours. We will consider the problem as purely intrinsic, that is, we address it without prior expectations of how many classes exist, or what these classes might be, but we do expect to find some degree of positive correlation.

The kind of population model we seek and which we will encode in the first part of the MML message, will, as in Snob, state the (hypothesized) number of classes K and for each class, its relative abundance and the parameters of its intensity distribution in each spectral band. In this study, we have assumed the pixels of a class to have independent normal distributions in each band, so for each band we have mean and standard deviation parameters. However, besides these standard Snob components of the population model, we state a parameter which specifies the (estimated) strength of neighbour class correlation.

Once the class abundances and correlation strength have been stated, they together define a ‘prior’ probability distribution over all possible assignments of pixels to classes, that is, a prior joint distribution over the classes of all pixels. We will call a class assignment of all pixels a ‘pattern’, so the joint distribution over pixel classes is a distribution over patterns.

Given some specified ordering of the pixels, for example, a raster scan, the prior probability of a pattern may, of course, be decomposed as the probability of the class of the first pixel, times the probability of the second pixel class conditioned on the first, times the probability of the third pixel class conditioned on the first and second and so on. So, in principle, we could proceed as in the one-dimensional case, encoding classes and data one pixel at a time and collecting statistics for re-estimation of the population model using partial assignment. However, whereas in the protein model, each class prior was conditioned on just the one previous class, the prior for the class of a pixel might be conditional on the classes of all previous pixels. In fact, this seems to be the case for all plausible pattern distributions which have N–S and E–W symmetry on the pixel grid. As these symmetries are clearly desirable in an unbiased prior, we reject the one pixel at a time or raster-scan approach as infeasible, as it replaces the simple fixed-order Markov chain of the one-dimensional problem with a process whose order increases as the scan progresses.

5.1. Gibbs sampling of the class pattern

Suppose for the moment that the number, abundances and attribute distribution parameters of the classes and the parameter(s) of the correlation model (pattern ‘prior’) are known or have been estimated. If the form of the pattern prior is such that the prior distribution over the possible classes of pixel X depends only on the classes of its four immediate neighbours, one may use Gibbs sampling to generate samples from the posterior pattern distribution given the data. Starting from some initial pattern, one re-samples the class of each pixel in turn. If N, E, S and W denote the classes of a pixel’s immediate neighbours and D denotes the attribute values of the pixel, then the probability that the pixel is class c is proportional to $\text{Prior}(c|N, E, S, W) \times \text{Prob}(D|c)$. A new value for the pixel class is sampled from this distribution and the process repeated on all pixels of the image.

Subject to mild restrictions on the regularity of the probability distributions involved, continued re-sampling of the pixel classes in this way leads in time to each possible pattern being generated with a frequency proportional to its posterior probability given the image data. Sampling from the pattern prior is also possible, by omitting the $\text{Prob}(D|c)$ factor.

The sampling process may also be viewed as a Monte Carlo Markov chain (MCMC) process [13] where the pattern distribution sampled is the stationary distribution of the pixel reclassification process. However, in our case the parameters of the Markov process are themselves altered as sampling proceeds.

During this sampling from the pattern posterior, statistics are accumulated from the pixels assigned to each class from which the class attribute distribution parameters are periodically re-estimated. Statistics are also accumulated for the re-estimation of the class abundances and the parameter(s) of the pattern prior. These new estimates define a new posterior, and hence a new Markov process. We re-estimate these global parameters only after several pixel reclassification passes over the image, so the statistics on which the new estimates are based represent the average of several plausible patterns and hence are relatively stable. (Full Gibbs sampling over the joint posterior of patterns and global parameters would be achieved if, following each reclassification pass, the global parameters were replaced by values sampled randomly from their respective posterior distributions given the current pattern. However, we have found in other (non-spatial) classification work that full Gibbs sampling leads to rather slow and erratic convergence.)

After a sufficient time, the re-estimated global parameters will approximate the MML estimates of the population model. As for the MML estimate of the pattern, it has been shown in [14] that the MML estimate of a high-dimensioned vector of parameters, such as the pattern, is very close to a pseudo-random choice from the posterior distribution of the vector. Thus, the pattern finally reached after some large, arbitrary number of sampling steps may be taken as an indication of the MML estimate.

5.2. A pattern prior

To enable simple Gibbs sampling, we have chosen a pattern prior such that the prior class distribution for a pixel is conditioned only by the classes of its four immediate grid neighbours, and has E–W and N–S symmetry. A simple pattern prior having this property gives a pattern s a prior probability

$$p_s = (1/Z)\exp(-q_s)$$

where

$$q_s = G_m \times N_m + \sum_c (B_c \times N_c),$$

N_m is the number of inter-pixel boundaries between pixels of different classes (termed 'strains'), N_c is the number of pixels of class c and G_m, $\{B_c : c = 1 \ldots K\}$ are parameters of the pattern prior. Z is a normalization constant depending on these parameters. Note that B_c parameters are determinable only up to an additive constant. Adding the same constant to all B_c parameters does not affect their prior probabilities. To remove the indeterminacy, we require

$$\sum_c (\exp(-B_c)) = 1.$$

With this constraint, if $G_m = 0$, then for each class B_c is the negative logarithm of the class relative abundance, but this relation does not hold for non-zero G_m.

With this pattern prior, the prior for the class of one pixel conditional on the classes of all others depends on the classes of its immediate neighbours only and gives class c a prior proportional to

$$\exp(-B_c - G_m \times (4 - n_c))$$

where n_c is the number of neighbours of class c, or

$$\exp(-B_c + G_m \times n_c).$$

We do not suggest this prior is very realistic or appropriate for the intrinsic classification of pixel images, as it captures only a primitive form of local class correlation. It was chosen purely to provide an experimental situation within which we could test the ability of the MML measure to estimate the number of classes.

We will have occasion to use the analogy between pattern distributions of the form

$$p_s = (1/Z)\exp(-q_s)$$

and the thermodynamic equilibrium, or Boltzmann distribution

$$p_s(T) = (1/Z(T))\exp(-q_s/T)$$

which gives the probability that a physical system in thermal equilibrium at temperature T will be found in a state s of energy q_s. Here, $Z(T)$ is a function of T, known as the partition function, satisfying

$$\sum_s (p_s(T)) = 1$$

and we use units such that the Boltzmann constant = 1. At temperature $T = 1$ the Boltzmann distribution becomes the pattern prior.

5.3. The core algorithm for 2D

Using the previously discussed prior, the core iteration to find the best model with K classes is straightforward. Some initial estimates of class distribution parameters, G_m and abundances are assumed. Also, some initial pattern of pixel classes is set up. In an inner loop of the iteration, several passes are made over all pixels of the image. For each pixel, the probability of its data D as a member of class c is computed and thence the unnormalized posterior

$$\exp(-B_c + G_m \times n_c) \times \mathrm{Prob}(D|c).$$

The posterior is normalized and a new class c' drawn from it for the pixel. The attribute values of the pixel are added to statistics for class c'.

The order in which the pixels are processed is not vital. We have chosen to do first all the pixels which would be white squares were the grid a chequerboard, then all the black squares. As the class prior for a white square depends only on black squares and *vice versa*, the ordering of white squares is immaterial, as is the ordering of black squares. We believe this ordering of a pass promotes rapid diffusion of information throughout the grid.

After several such passes, the attribute distribution parameters for each class are re-estimated from the accumulated statistics. The pattern prior parameters G_m and $\{B_c : c = 1 \ldots K\}$ are re-estimated by a maximum-likelihood method using counts of class assignments and neighbour strains accumulated during the passes.

The outer loop repeats the above process until the population model and pattern energy appear to be stable.

The techniques (Gibbs or MCMC sampling of class pattern, with periodic re-estimation of global parameters) used in the core algorithm are well known in the image analysis literature, for example [15, 16], as are the kinds of class model and pattern prior used here. The contribution of the present work is to introduce to this type of analysis a well-founded criterion (MML) for estimating the number K of classes from the data.

5.4. Calculation of message length

This core iteration yields a classification model with K classes (or perhaps fewer if one or more classes have been eliminated). To estimate K we need to calculate the message length for this model and compare it with the lengths given by models of different K, preferring the number of classes yielding the shortest message.

The calculation of message length needed to state the number of classes and their attribute distribution parameters presents no problem and follows the calculations used in the uncorrelated case, i.e. follows Snob. The message length to encode the pattern prior parameters is small and we have neglected it *pro tempore*. The final component of the message, encoding the pixel values given the population model and pixel classes, also follows Snob calculations exactly and for each pixel just has length the negative log of the probability of the pixel values given the attribute distribution of the pixel class.

It remains to consider the encoding of the class pattern. The total energy Q of the final pattern gives its prior probability as $(1/Z)\exp(-Q)$, and hence the message length needed to encode the pattern as $\log(Z)+Q$. However, the calculation of the normalization constant Z from the pattern prior parameters and K is difficult and we have resorted to a Monte Carlo estimation.

5.5. A useful thermodynamic relation

We will use a standard result of thermodynamics, which is outlined here for those unfamiliar with it. Define $H(T)$, the 'entropy' of the system at temperature T, as

$$H(T) = H = -\sum_s (p_s \times \log(p_s)).$$

Define the expected total energy of the system as

$$Q(T) = Q = \sum_s (p_s \times q_s)$$

and define $U = -\log(Z)$, so

$$\begin{gathered} p_s = \exp(U - q_s/T), \\ p'_s = \mathrm{d}p_s/\mathrm{d}T,\ H' = \mathrm{d}H/\mathrm{d}T, \\ Q' = \mathrm{d}Q/\mathrm{d}T,\ U' = \mathrm{d}U/\mathrm{d}T. \end{gathered}$$

Then, since $p_s = \exp(U - q_s/T)$:

$$\begin{gathered} p'_s = p_s(U' + q_s/T^2) \\ \sum_s (p'_s) = 0 = U' + Q/T^2. \end{gathered}$$

(Because $\sum_s (p_s) = 1 = \text{constant}$.) Hence:

$$U' = -Q/T^2. \tag{1}$$

Also:

$$\begin{aligned} H &= -\sum_s (p_s \times \log(p_s)) \\ &= -\sum_s (p_s \times (U - q_s/T)) \\ &= -U + Q/T \end{aligned}$$

hence

$$H' = -U' + Q'/T - Q/T^2.$$

Using (1):

$$H' = Q'/T, \quad \mathrm{d}H = \frac{\mathrm{d}Q}{T}.$$

That is, the change in the entropy of the system due to a change in energy is the change in energy divided by the temperature at which the change occurs.

5.6. The prior entropy

The usefulness of this relation in our context is thus: the entropy of a distribution is the average negative log probability of a state, or class pattern in our case. We may argue that the final pattern is a 'typical' realization of the pattern prior, as we have adjusted the pattern prior parameters to fit the sampled patterns. Hence, the entropy of the pattern prior at $T = 1$ is a reasonable estimate of the negative log prior probability of the final pattern. This relation enables us to estimate this entropy. To do so, we set up the final class pattern, and commence Gibbs sampling from the pattern prior. This is done simply by dropping the $\mathrm{Prob}(D|c)$ factor from the distribution over the class of a pixel. During this sampling, the pattern prior parameters are left unchanged. As the sampling proceeds, the temperature T is gradually lowered from one. That is, the relative probability of two states s, r of energies q_s, q_r is taken as

$$p_s/p_r = \exp((q_r - q_s)/T).$$

As T drops, the total energy Q of the pattern tends to decrease, and we accumulate the entropy decrements $-\mathrm{d}Q/T$. At a sufficiently low temperature, the pattern will 'freeze', typically with all pixels having the most abundant class. As this pattern is completely ordered, it has no entropy, so the accumulated decrement estimates the initial entropy and hence the message length needed to encode the estimated class pattern.

This method of estimating the entropy of the pattern prior was programmed and tested for various values of the prior parameters. It worked, but on repeated runs with fixed parameters, showed a rather large variation in estimated entropy unless the cooling was very slow. We devised an alternative scheme which, for the same computational effort, gives a standard deviation of estimates typically a factor of about 10 less than the original.

In the alternative, we again sample from the pattern distribution, beginning with the final pattern found in the core iteration and again keeping the prior parameters fixed. Instead of cooling the system from $T = 1$, it is gradually heated from $T = 1$. As the temperature increases, the pattern becomes less ordered and increases in energy and its increase in entropy is accumulated. The temperature is gradually increased to infinity. At this point, each pixel is equally likely to be assigned to any class, independently of its neighbours, so the entropy of the pattern distribution at infinite temperature is simply

$$H(\infty) = (\text{number of pixels}) \times \log(\text{number of Classes}).$$

Subtracting the entropy accumulated during the heating from $H(\infty)$ gives the entropy at $T = 1$, which is the desired prior entropy.

The final pattern found at the end of the core algorithm is a random sample from the posterior over patterns, given the population model and data, but may not be typical of the prior over patterns given G_m and B_c, so it may appear that either of these methods of estimating the entropy of the pattern prior is inappropriate for what we actually want, which is the log prior probability of the final pattern. In fact, by starting the entropy estimation process from the final pattern, we do get what we want. The Gibbs sampling at the initial temperature $T = 1$ will rapidly result in the average energy of the pattern approaching its average value under the prior. If the final pattern had an energy greater than this average, the energy loss as the average approached would be accumulated in our estimate of entropy change and if the average is achieved while the temperature is still close to one, it accumulated and divided by $T = 1$. The final estimate of prior entropy, or average log prior probability of patterns, is therefore corrupted by the difference in energy between the final pattern and average patterns. However, this difference is just the difference between the log probability of the final pattern and the average log probability. Thus, our estimate of the prior entropy is automatically modified to give an estimate of the negative log prior probability of the final pattern, rather than the average.

5.7. Pattern precision

We now have estimates for the lengths of all components of the message: population model, class pattern and pixel data values and so can compute a total message length. However, this length is not the shortest which can be achieved by a K-class model, because the encoding of the class pattern is too precise. The message as constructed previously states an estimated pattern exactly, but if there is any overlap in the class attribute distributions, there will be many pixels whose class is somewhat uncertain and hence many, perhaps many millions, of patterns any one of which would have led to an equally short message. The freedom to choose among these plausible patterns can be exploited to reduce the message length [4].

The means to do so can be seen most easily by supposing that we wished to encode and transmit not only the image data, but some other unrelated information such as a personal letter. We could choose among the equally plausible patterns in such a way as to encode some of the initial binary digits of the letter. After receiving the coded image, the receiver can decode the original pixel data and hence herself reconstruct the set of plausible class patterns. By noting which of these has actually been used in the message, she can recover the initial digits of the following letter, so these need not be included in the image + letter message. The economy can be credited to the encoding of the image, thereby reducing its effective length by the log of the number of plausible patterns.

This coding device has been re-discovered and well described by [17], but without the development described in the next paragraph.

It is less obvious that this economy can be realized within the image message itself, rather than by shortening the length of some following messages. However it is possible, by recasting the image message so that, after stating the population model, the message encodes each pixel in turn, say in raster-scan order. To encode a pixel, its class is stated using a code based on its class prior conditioned on the classes of all previous pixels and then the pixel data values are encoded using the attribute distributions of the stated class. Any uncertainty in the class to which a pixel should be assigned is exploited to encode the initial digits of the code for the next pixel, which may therefore be omitted. Such a construction is, in principle, possible, although computationally infeasible because of the difficulty of computing the conditional class prior for each pixel. The details are given in [4].

Either approach leads to the same conclusion: the effective or actual length of the image message can be reduced by the log of the number of reasonably plausible patterns, or more exactly, by the entropy of the posterior distribution over patterns, given the population model and image data. It is just this distribution which is sampled in the core iteration, at least after estimates of the class distribution parameters etc. have stabilized. The entropy of the distribution can be estimated by the same means used for the pattern prior. That is, sampling is continued while the temperature is gradually lowered from one, and the $\mathrm{d}Q/T$ decrements accumulated. But now, the data probability $\mathrm{Prob}(D|c)$ for each pixel is also interpreted as the negative log of an energy term. The cooling will eventually freeze on a class pattern close to the maximum-likelihood estimate of the pattern, which is the minimum-energy pattern. In this case, there is no worry about the typicality of the final pattern from which the cooling is started, as it is indeed a random sample from the posterior over patterns, which is the distribution whose entropy we want.

For estimating the entropy of the pattern posterior, cooling till the pattern freezes is better than heating to a high temperature, as is used for the pattern prior. The variance of the estimate is small because the data prevents large changes in the pattern. Were heating used, a temperature would have to be reached so high that the differences in energy resulting from different class assignments for a pixel would be small compared with the temperature. For the pattern prior, the maximum energy difference is of order $4 \times G_m$, typically less than 20, so an 'infinite' temperature around 200 suffices. For the posterior, many pixels typically have a very low chance of being a member of some class and hence a very large energy if assigned to that class. Heating would have to continue to extremely high temperatures to reach a pattern distribution showing no trace of the pixel likelihoods.

The posterior pattern entropy so estimated is then subtracted from the total message length to give an approximation to the MML message length for a K-class classification. The present program makes no attempt to split or combine classes during the core iteration, so the whole process of core iteration, estimation of prior and posterior entropies, must be repeated for differing K to find the optimum number of classes, i.e. the number giving the shortest message length. There is no reason in principle why the automatic K-selection technique of [9] could not be used instead.

6. PRELIMINARY RESULTS

As yet, we are still experimenting with the algorithm to learn the most efficient deployment of computing effort among the phases of the algorithm: preliminary annealing of the class model, core iteration at $T = 1$, cooling of posterior and heating of the prior. Before attempting to analyse real data, where the 'real' classification is unknown, we wish to establish that the algorithm can at least recover a good model of data which has been artificially drawn from a population consistent with the assumptions of the algorithm. This is of course shooting fish in a barrel, but an algorithm which cannot pass such a test would be unfit for real use.

A typical test was done on a (32×32) pixel array, where each pixel has four real-valued attributes. Four classes were created with mean attribute vectors $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 0)$ and $(0, 0, 0, 1)$ and in each class each attribute has standard deviation 0.4. The pattern prior parameters were set at

$$G_m = 1.0, \quad B_1 = B_2 = B_3 = B_4 = -\log(4).$$

To generate a data sample, first the class of each pixel was chosen by Gibbs sampling patterns from the pattern prior and choosing the 1000th pattern found. Then pixel attributes were chosen randomly from the appropriate class Gaussian densities. It is worth noting that although the population from which the 1024 pixels are drawn has equal abundance for all classes (all B_c equal), the data sample tends to have notable differences in class counts, for example (301, 241, 175, 307) for the data on which the results outlined later were found.

6.1. Reproducibility

Any algorithm based on random sampling may give varying results on the same data when different pseudo-random number streams are used, so one needs to examine the extent of this variation. We conducted 10 runs on the data described above for each of the class numbers two, three, four and five (the 'true' number being four) and found the maximum, minimum, mean and standard deviation (SD) of message length for each class number. The maximum for four classes was below the minimum for any other number, by a margin of at least 17 (versus five classes). This difference may appear too small to indicate a clear preference for four classes, although the standard deviation of the mean lengths was much smaller, about one. However, the difference in message length between two models for the same data may be interpreted as the log posterior odds ratio in favour of the model giving the shorter message. Thus the odds in favour of the worst four-class model over the best five-class one are nominally over 24 million. This figure must be taken with a pinch of salt as our class-model priors are fairly arbitrary, but a strong and reliable preference for four classes is evident.

6.2. Accuracy of estimated population model

The means of the attribute vectors are recovered well. For the four-class model, the means were in error by about 0.02 and the standard deviation estimates also by about 0.02 (the true value being 0.4). The class abundances expected on the model were, for the first run, (301, 245, 178, 297), close to the true counts (301, 241, 175, 307).

The estimate of the correlation parameter G_m averaged 0.95 with little variation over the 10 runs. This is somewhat below the population value 1.0, but runs on other data sets from the same population gave estimates ranging from 0.90 to 1.05. Evidently, this parameter of the population is not estimated very accurately and may have a slight bias to low values.

6.3. Accuracy of class pattern

The four-class population from which the data was drawn has four equidistant classes in four-space. Their distributions overlap substantially, so there are many pixels whose class cannot be determined with confidence. A run of the algorithm provides two estimated patterns. One ('TYP') is found at the end of the core sampling and is intended as a random draw from the posterior over patterns, given the estimated population model. The other ('ML') is found at the end of the posterior-cooling process and should approximate the most likely pattern given the same population model. For one run with four classes, TYP misclassified 98 pixels, showing 853 strains. ML made fewer errors (76) but showed only 811 strains, the true value being 843.

We also ran the algorithm with G_m held at zero, simulating the normal Snob algorithm which takes no account of correlation. These runs still showed a preference for four classes by a margin of 14 in message length, but gave worse estimates of both population and pattern. The four-class TYP pattern had 146 errors, the ML had 103. The number of strains was badly over-estimated at 1067 (TYP) or 988 (ML).The message length for this uncorrelated model exceeded that for the correlated model by 295, so the MML criterion showed an overwhelming preference for a model with spatial correlation.

Further runs on data generated with all attribute SDs $=$ 0.7 and $G_m = 1.1$ showed similar results. With the more diffuse class distributions, the overlap among classes increased, resulting in more misclassified pixels (295 and 214 in the TYP and ML patterns respectively). The message lengths still show a preference for four classes, but only by a difference of six over the best three-class run. Runs with $G_m = 0$ (no correlation) gave the shortest message length with two classes (a one-class model was not tried) so did not detect the presence of four classes. The best four-class model gave 486 (TYP) or 395 (ML) misclassified pixels. Clearly, the correlated algorithm is superior in recovering

```
= = = = = o | | % % % % % % % o o | | o o o o o o o | | o | % o
= = = = = % % | % % % % % % | | | | | | o o o o o o o | | | | |
= = = = | % % % % % % % % % | | | % = = = o o o o o o = | | | |
= = = = = = % % % % % % % % % % % % = = = o o o o o o o | | = |
= = = = = = % % % % % % % % % % % % % = = o o o o o % o % % = |
= = | | | | | % | | % % % % % % % % % o o o o o o o o o % % | |
= = o | | | | % % | | o | % % % % % % % o o % % % = o o % % | |
% | o o | | | | | | | o | % % o % % % % % | % % % | = % % % | |
| | o o | | | | | | o o | = % % % % % % % % % % % | | | % o o o
o o o o | | | | | | o = = = = = % % % % % % % % % % % % % % % o
o o o o | o | | | = = = = = = = % % % % % % % % % % % % % % % %
| | o = % o | | | | % = = = = % % % % % % % % % % % % % % | | |
| | | = % o | % % % % = = = | % o % % % % % % % % % % % % | | |
| | | = = | | % % % % % % % % % o | % | | % | % | | | | | | | |
| | | = = o % % % % % % % % % % o | | | | | | | | | | | | | | |
| | = = = = % o % % % % | | | | | | | | | | | | | | | | | | | |
= % % % % = = = % = = | | | | | | | | | | | | | | | | | | | | |
= % % | % = % % % | | | | o | | | | | | | | | | | | | | | | | |
= % | o = % % % % | | % | | o = = | = = | | | | | | | | | | | |
= = o o | | | % % % | % o o o = = = = = | | | | | | = = | | | |
o o o o | | | | | | | o o o o o = = = = | | | | = = = = = | | |
o o o o o | | | | | | o o o o | | | | | | | | | | = = = | | | o
o o o o o | | | | | o o o o o | | | | % | | | | | | = = | | % o
o o o o o o | | | | | | | | | | | | | % | | | | | | = = | | % |
o o o o o o | | | | | | | | | | | | o | | | | | | | | | | | % %
o o o o o | | | | | | | | | | | | | o = | | | | | | | | | | | %
o o o o o o o | | | | | | | | | | = o = = = = = = | | | | | | |
o o o o o o o o o | | | | | | | | = = = = = = = = = | | | | | |
o o o o o o o o o | | | | | | | | = = = = | % = = o o | | | | |
o o o o o o | o o | | | | | | o o | = = = | % = = o | | | % % %
= % % % % o = o o | | | | | % o = o = = = | | | | | | % % % % %
= % | | % o o o o | | | | % % = = o o | | | | | | | | % % % % %
```

FIGURE 1. Most likely pattern from data with all attribute SDs = 0.7 214 errors. Symbols represent the four classes.

```
| = = % = o | | % = | = = = % o o % | % = % = % o o | o o | % o
| = % = = % % | o % % | | % | | | o o | o | o o = o o % | | | o
= = % = | = = = % o o | = % | | | % = = = o = o o | o = o | o |
o = = % % = = | % % o % = % % % = o = = = o = o | o o o o | o =
= % = % % | % | o % o = | = % % o o | = = o = = o | % o % = = %
= | | | | | | o | % = % % | o % % | o = o = o o | = o o | % | =
= = o | o | = % % | | o | % = % = % % = = o % = % = o o % % % %
% | o = % | | o = | | o | % % o % % = % % | % = % | = % o % | |
% | o o % | o = | = o o | = % % o o % % % = | % % | | | = o o o
o o o = | | | | | | o = = = | = % % | % % | % % = % % % o % % o
o | o o | o | | = = = = = = o = % % % = = | % = % o % | = = = %
| | o = = = | o | | % = = = = % % % | o | % % | = | % | o | % =
| = | = % o | = = = | = = = o % o % % % % o = % % % % = % | | %
| % o = % | | % % % % % | = = | = | o | | % | o | = | | % % | %
% | o = = o % = % % o % = % % | o | | o | | % = | o | | = = | |
| | = = = = % o % % % % % o | | = o | | | % | | % | = | | = = |
= % o % o = = = % = = | | % % = | = | = o | | | | | % % | | | o
% o % | % = % % % % | | | o | | = % % % = | o | o o o | | | | |
= % | o = % = % % | | % | | o = o | = = o o | = | | | | = | o =
= | o = | % | % % % | % o o o = = % = = | % | % % o o = o o | |
o o o o | = | o o | % = o o o o = = = = o | | = = | = % = | | %
| o | o = % | o % o | = % = o = | | | % | o | % | = = = o | | =
o % o = o | | | % = o % = = o % = | | % | | o | = | | o | | % o
o o o o o o = | | o = | = | | | o | | % % | | | = % = = | | % |
% o o o o o | | % % | o | | o = % = o | % | | | = | | | | | % =
= % % o o | = o | = = | | | | | = | o = o | % | % % = | | = | =
o o o o % o o | | o % = | | | o | = o = = = = | = | | | | | | %
= o = % % o = = = | | | o % % | % | = = = = = o | = o | | | | |
o | o % o o o o o | | | = | | | = = = = % | % = = o o | % = o |
o o o o = = | o | = | | % | % o o | = = = | % = = o = | | % % %
| % % % = o = = o = | | % | % o = o = = = | | | | | | % o % % o
= % o | % o = o o % % o % % o = = o o | o | o o % % % o % % o %
```

FIGURE 2. Most likely pattern (SDs = 0.7) found without correlation. 396 errors.

the population model and in classifying the pixels when correlation is indeed present. Figure 1 shows the ML 4-class pattern found using the correlation algorithm. Figure 2 shows the corresponding result found without correlation. Figure 3 is the 'true' pattern from which the data was generated.

```
= = = = = o | | % % % o % % % o o | | % o o o o o o | | o | = o
= = % = = | | | % % % % % % | | | | | | o o o o o o | o | | | |
= = = = = o % % % % % % % % | | | % % = o o o o o o o o | | | |
= o o o % = o % % % % % % % o % % = = = = o o o o o o o o | = =
= = o % % o o o o % % % % | | % % o = = = o o o o o % o % % = =
= = | | | | = o | | % % % | | % % % % o o o | o o o o o % % % %
= | | | | | | % % | | o = % % % % % % % % o | o o o o o % % % %
% | o o | | | o = | | o = = o o % % % % % | % % % | = % % % %%
% | o o | | | | | o | o = = % % o o % % | = % % % % | | % o o o
| | o o | | | | | = o % = = = = % % % % | % % % % % % % % % % o
| | o o % o | | | = o = = = = = % % % % % % % = % % % % % % % %
| | | = o o | | | = = = = = = % | % % % | % % % % % % % % | | |
| | | = o o | = % % % % = = = = o % % | % % % % % % % % o | | %
| | | = | | | % % % % % | % = = | | o | | % | % % % | | | | | |
| | | = = % % % % % % % % % o = | | | | | | = | | o | | | | | |
| | = = = % % % % % % % % % | | | | | | | | | | | | | | | | | |
| % | | % = = = % = = | | % % % | | | = | | | | | o | | | | | o
= % % | % = % % % | | | | o | | | % % % % | | | | | | | | | | |
= | | o = % % % % | | | | = = = = = = = o | | | % | = = = | | |
o | o o | | | % % % | % o o o o o = = = | | | = % % = = = | | |
o o o o | | | | % | | o o o o o o = = = | | | | % | = = o | | =
| o o o o o | | | | | o o = o | | | | | | | | | | o = | | | = o
o % o o o | | | | % o o o o | | = | | | | | | | = % = | | | % o
o o o o o o = | | o | | = | | | o | | % | | | = = % = = | | % |
o o o o o o | | | | | | | | | = | o o = | | | = = | | | | | % %
% o o o o | | | | | | | | | | | | | o = | = % | = | | | | | | %
o o o o | o o | | | % o | | | | | o o = = = = = = | | | | | | |
| o o o | o o o o | | | | | | | = = = = = = = = | = | | | | | o
| % o o o o o o o | | | | | | | % = = = = = = = = o o | | | | |
| o o o % o | o o | | | | | | o o | = = = | % % % | | | % % % %
= % % % % o = = % | | | % % % o o | = = | | | | | | | % % % % %
= % | | % o o o o % % | % % o o % | | = | | | | | % % % % % % %
```

FIGURE 3. True pattern for data for Figures 1 and 2.

6.4. A real image

A five-band satellite image of the Lake Eyre area has been used as a test image. It has (256×256) resolution. The area is mostly desert and salt pan, with some scattered areas of vegetation. Runs seeking between four and 20 classes have been done and showed a high level of neighbour correlation ($G_m \approx 2.3$). The smallest message length was found with 20 classes. This may seem a large number. It is well known that correlation among the attributes can lead a classifier whose class model does not admit in-class correlation to over-estimate the number of classes. We transformed the data to remove the most obvious inter-band correlation, a general 'brightness' component, but found no great change in the results. The low entropy of the posterior pattern distribution, about 0.08 per pixel, shows that the classification of most pixels is not in much doubt and the different classes are quite distinct. However, it would be desirable to provide for an unrestricted multivariate Gaussian distribution of attributes within each class, as is done in AUTOCLASS II [6].

The classifications with between four and 20 classes showed good agreement with terrain type, but proper assessment of the method on real data will require much more testing.

The analysis time for six classes on the (256×256), five-band image is about 25–30 min on a 133 MHz Pentium PC.

7. EXTENSIONS

The one-dimensional method has given useful results on protein structure using a first-order Markov class correlation model. Extension to second-order models raises no difficulties, in principle, and will be attempted.

Obviously the 2D algorithm requires more extensive testing to discover its performance in more diverse problems.

If it proves useful, it can be extended to allow more flexible correlation models. At present, a mismatch between any two adjacent pixels incurs the same energy penalty G_m. It could be extended to provide a different penalty for every pair of different classes and to estimate the values of these penalties just as the present program estimates G_m. In some domains this could be useful: *a priori* we may be more surprised to find a 'rainforest' class adjacent to a 'desert' class than to find a 'mangrove swamp' class next to an 'estuary' class. This scheme would require inference of $K^2/2$ penalty parameters, so it would work well only on large images.

Another possible extension would be to allow mismatches in an E–W direction to incur a different penalty from those in a N–S direction. This could allow for the situation, common in aerial survey data, where pixels are more closely spaced in one dimension than in the other. Again, the algorithm could discover the difference for itself, or prior knowledge could be used to require G_m(E–W) to be some specified multiple of G_m(N–S).

In principle, the main contribution of our 2D algorithm, the use of a message-length measure to determine an appropriate number of classes, could be incorporated in the more sophisticated spatial-correlation methods which have been developed by others, for example [15, 16].

ACKNOWLEDGEMENTS

The one-dimensional method was conceived by Lloyd Allison and Tim Edgoose and was largely developed by the latter with technical assistance on MML from the author.

This work was supported in part by Australian Research Council grant Nos A49330656 and A49703162. We are grateful to the anonymous referees for helpful suggestions.

REFERENCES

[1] Wallace, C. S. and Boulton, D. M. (1968) An information measure for classification. *Comp. J.*, **11**, 185–195.

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

[3] Boulton, D. M. and Wallace, C. S. (1973) An information measure for hierarchic classification. *Comp. J.*, **16**, 254–261.

[4] Wallace, C. S. (1986) An improved program for classification. *Proc. ACSC-9*, **8**, 357–366.

[5] Dowe, D. L. and Wallace, C. S. (1997) MML mixture modelling of multi-state, Poisson, von Mises circular and Gaussian distributions. *Comput. Sci. Stat.*, **28**, 608–613. Available on http://www.cs.monash.edu.au/∼dld/Snob.html.

[6] Cheeseman, P. C. (1988) AUTOCLASS II conceptual clustering system. In *Proc. Machine Learning Conf.*, 54–64. Available on http://ic-www.arc.nasa.gov/ic/projects/bayes-group/autoclass.

[7] Wallace, C. S. and Freeman, P. R. (1987) Estimation and inference by compact coding. *J. R. Stat. Soc.*, B, **49**, 240–265.

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

[9] Richardson, S. and Green, P. J. (1997) On Bayesian analysis of mixtures with an unknown number of components. *J. R. Stat. Soc.*, B, **59**, 731–792.

[10] Edgoose, T., Allison, L. and Dowe, D. L. (1998) An MML classification of protein structure that knows about angles and sequence. In *Proc. 3rd Pacific Symp. on Biocomputing*. World Scientific, Singapore.

[11] Hunter, L. and States, D. J. (1992) Bayesian classification on protein structure. *J. IEEE Expert*, **7**, 67–75.

[12] Dowe, D. L., Allison, L., Dix, T. I., Hunter, L., Wallace, C. S. and Edgoose, T. (1996) Circular clustering of protein dihedral angles by minimum message length. In *Proc. 1st Pacific Symp. Biocomp.*, pp. 242–255. World Scientific, Singapore.

[13] Besag, J. and Clifford, P. (1991) Sequential Monte Carlo p-values. *Biometrika*, **78**, 301–304.

[14] Wallace, C. S. (1996) False oracles and strict MML estimators. In Dowe, D. L., Korb, K. B. and Oliver, J. J. (eds), *ISIS Information Statistics and Induction in Science*. World Scientific, Singapore.

[15] Daily, M. J. (1988) Colour image segmentation using Markov random fields. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 304–312.

[16] Geman, S. and Geman, D. (1992) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE PAMI*, **6**, 721–742.

[17] Frey, B. J. and Hinton, G. E. (1996) Free energy coding. In *Proc. Snowbird Data Compression Conf.* IEEE, Morristown, NJ.