Statistics and Computing (1999) **9**, 269–278

MML Markov classification of sequential data

T. EDGOOSE and L. ALLISON

Computer Science Department, Monash University, Clayton, VIC 3168, Australia
e-mail: {time, lloyd} @cs.monash.edu.au

Received October 1997 and accepted May 1999

General purpose un-supervised classification programs have typically assumed independence between observations in the data they analyse. In this paper we report on an extension to the MML classifier Snob which enables the program to take advantage of some of the extra information implicit in ordered datasets (such as time-series). Specifically the data is modelled as if it were generated from a first order Markov process with as many states as there are classes of observation. The state of such a process at any point in the sequence determines the class from which the corresponding observation is generated. Such a model is commonly referred to as a Hidden Markov Model. The MML calculation for the expected length of a near optimal two-part message stating a specific model of this type and a dataset given this model is presented. Such an estimate enables us to fairly compare models which differ in the number of classes they specify which in turn can guide a robust un-supervised search of the model space. The new program, tSnob, is tested against both 'synthetic' data and a large 'real world' dataset and is found to make unbiased estimates of model parameters and to conduct an effective search of the extended model space.

Keywords: Classification, Hidden Markov Modelling, MML, spatial data

1. Introduction

Classification, also known as mixture modelling and clustering, is the building of models from sets of observations where each observation is assumed to have been generated from one of a finite number of classes. A particular classification model specifies the number of such classes and a distribution over the observations expected for each. Un-supervised classification programs attempt to find the most appropriate class structure and parameterisation given a set of observations.

However, selecting the most appropriate number of classes requires that a balance to be struck between model complexity and explanatory power. The best model will be sufficiently complex as to avoid discarding information implicit in the observed data, but not so complex as to be overly specific to the observed data (over fitting). A partial solution to this problem was presented in earlier un-supervised classification work by Wallace and Boulton (1968) and subsequently generalised by Wallace (1987, 1990). A Minimum Message Length (MML) information measure was proposed that would estimate the length of an optimal two-part message stating a model and a set of observations given the model stated. Such a message length gives an information measure by which any two competing classification models can be fairly compared.

This earlier MML classification work was designed to model randomly sampled data and hence assumed independence between observations in a dataset. This assumption fails to take advantage of some of the extra information implicit in datasets where the observations are not randomly sampled such as sequences.

In this paper we present an MML based approach to the un-supervised classification of a sequence of observations which takes advantage of some of the extra information contained in such data. Specifically the data is modelled as if it were generated from a first order Markov process with as many states as there are classes of observation. The state of such a process at any point in the sequence determines the class from which the corresponding observation is generated. Such a model is commonly referred to as a Hidden Markov Model (HMM) and is able to capture the sequential structure of datasets where the probability of an observation in a certain class depends only on the class of

the immediately preceding observation (i.e. first order). Such a model has only one probabilistic state variable (with as many possible states as there are classes) that influences the assignment of the next observation and which is then updated and propagated along the Markov chain. For sequences with a more complicated sequential structure (perhaps a simple grammar) the best first order approximation will be found. Although not appropriate for all types of sequential data, this sequence model is none the less of significant practical interest. For a good introduction to these models which are rich in mathematical structure and have been used extensively in the area of speech recognition refer Rabiner (1989). An iterative solution for these models was first proposed by Baum *et al.* (1970). The technique applied was an Expectation Maximisation (EM) method later generalized by Dempster and Rubin (1977). Later work by Leroux and Puterman (1992) improved on the work by Baum by using a Bayesian Information Criteria (BIC) to estimate the complexity of a given HMM model and hence they were able to compare two HMMs with a different number of states. However, these works suffered from the lack of a suitable search method for larger model spaces and also from the surprising notion that there can ever be enough observational evidence to justify a probability of zero (or one) when estimating model parameters. This in turn led to zeros being preserved in the transition matrix and the possibility of a search being trapped in such a solution. We extend this earlier work by deriving an MML information estimate for such a model, an improvement on the approximate BIC estimate, and we specify an effective search method of this complex model space which is guided by this measure.

The MML classification program Snob of Wallace (1990) has been re-implemented and extended in order to model first order Markov processes. The new program, tSnob, is a cut-down variant of the MML classifier written in the C programming language. It includes most of the existing functionality of the 1993 Snob implementation with the exception of poison distributed attributes. The program is designed to model multi-variate data with a fixed number of attributes. The type of these attributes can be discrete, continuous or angular these being modelled by multi-state, Gaussian or von Mises distributions respectively. Attribute values are assumed to be independently distributed within a class and for any particular observation in a dataset any or all attribute values may be tagged as missing.

The MML modelling approach is Bayesian in nature and strong parallels exist between Snob and the Bayesian classifier Autoclass produced by Cheeseman (1988). The methods are contrasted in Wallace (1990). MML contrasts with the Minimum Description Length (MDL) information measure subsequently proposed by Rissanen (1987) in that MML states a specific model parameterisation and the encoding is based on domain specific prior knowledge rather than a universal prior.

This paper includes a short introduction to MML encoding, the construction of a MML code for this class of Markov classification model, a specific model space search procedure and some experimental results based on 'synthetic' and 'real-world' data. The MML information measure of the Markov classification model yields improved class model selection results when modelling sequential datasets even where there is only a modest dependence between adjacent observations. Such a measure also results in unbiased parameter predictions and can guide a search of the model space thus making a difficult search problem more tractable.

2. MML background

Within the MML paradigm, models are judged by their ability to reduce the length of a message transmitting all our observations to a receiver who initially shares only our prior beliefs. We calculate the expected length of an optimal message sending a model (i.e. an hypothesis) to an optimal precision and our observations given this model (i.e. the evidence). The best model will minimise the length of this two-part message. No model that fails to compress the evidence can be considered superior to the empty model (null hypothesis).

These two message parts are used as estimates for $P(H)$ and $P(D\,|\,H)$ in Bayes Theorem.

$$P(H\,|\,D) = \frac{P(H)\,\cdot\,P(D\,|\,H)}{P(D)}$$

For any one particular dataset $P(D)$ is constant, so the best hypothesis will maximise

$$P(H\,\&\,D) = P(H)\,\cdot\,P(D\,|\,H)$$

This gives an fair criterion which we can use to compare any two hypotheses based on a given set of observations.

Wallace and Freeman (1987) gives the general form of such an MML estimate (given an appropriate likelihood function and a stable prior)

$$\begin{aligned} ML(H\,\&\,D) \approx &-\ln h(H) + \tfrac{1}{2}\ln\det(F(H)) \\ &-\ln f(D\,|\,H) + g(n_p) \end{aligned}$$

where $h(H)$ is a prior distribution over parameter values, $F(H)$ is the Fisher Information matrix, $f(D\,|\,H)$ is the likelihood function for the model, $g(n_p)$ is a function of the number of parameters being estimated, and the unit of the result is natural bits or nits (divide by ln 2 to convert to bits).

The objective function for Snob is constructed using three such MML optimal code length estimates. The estimates we use are the same as those in the original Snob program. They predate the derivation of the general form, but are close approximations.

The first of these is the multi-state distribution where observed values are discrete and have come from a finite

unordered set of possibilities. From Wallace and Boulton (1968) the optimal MML code length to transmit K values from an M state multi-state distribution (assuming a uniform prior over the possible combinations for the frequencies of the observed values) is:

$$ML(H\,\&\,D) \approx \frac{M-1}{2}\left(\ln\frac{K}{12}+1\right) - \ln(M-1)! - \sum_{m=1}^{M}\left(n[m]+\frac{1}{2}\right)\ln p[m]$$

where $n[m]$ is the number of values in state m and $p[m]$ is the probability stated for state m and is re-estimated as

$$p[m] = \frac{n[m]+\frac{1}{2}}{K+\frac{M}{2}}$$

The second is the Normal distribution where values are continuous reals stated to a specified accuracy. From Wallace and Boulton (1968) the optimal MML code length to transmit K values from a normal distribution with mean, μ, standard deviation, σ, and measurement accuracy, ε, from a global distribution with mean, μ_p, and standard deviation, σ_p, (assuming μ has a uniform prior in the range $\mu_p \pm 2\sigma_p$ and $\ln\,\sigma$ has a uniform prior in the range $\ln\,\varepsilon$ to $\ln\,\sigma_p\sqrt{2\pi}$) is:

$$ML(H\,\&\,D) \approx -\ln\left(4\sqrt{\frac{K}{12}}\frac{\sigma_p}{\sigma}\right) - \ln\left(\ln\left(\frac{\sqrt{2\pi}\sigma_p}{\varepsilon}\right)\sqrt{\frac{K-1}{6}}\right) - K\ln\left(\frac{\sigma\sqrt{2\pi}}{\varepsilon}+\frac{1}{2}\right)+\frac{1}{2}$$

and σ is re-estimated as $\sqrt{\frac{v}{K-1}}$ where v is the sample variance.

Finally, we consider the von Mises distribution detailed in Fisher (1993) for modelling angular values stated to a known accuracy. It has mean direction μ and concentration parameter κ. Letting $I_0(\kappa)$ be the relevant normalisation constant, it has probability density function

$$f(x\,|\,\mu,\kappa) = \frac{1}{2\pi I_0(\kappa)}\,\mathrm{e}^{\kappa.\cos(x-\mu)}$$

which for small κ tends to a uniform distribution and for large κ tend toward a Normal distribution with variance $1/\kappa$. This is a circular analogue of the Normal distribution – both being maximum entropy distributions. The MML estimate for the von Mises distribution is less compact and can be found in Wallace and Dowe (1993).

These three expected message length functions will be minimised when the model parameters stated best fit the data to be encoded.

3. Calculating the message length of model and data

In this section we define the calculation for the length of a near optimal message encoding a specific parameterisation of our Markov classification model (stated to an appropriate accuracy) and a sequence of observations given this model. We derive the length of such a message by first stating the length of a non-optimal encoding and then deriving the length of the optimal encoding by argument.

Our encoding consists of the following components:

(a) The number of classes (N)
(b) The relative abundance of each class
(c) For each class
- The distribution parameters for each significant attribute
- The relative abundance for the next class conditional upon being preceded by this class.

(d) For each observation
- An assigned class
- The attribute values given this class

Parts (a), (b) and (c) constitute our hypothesis, H, and part (d) is one possible encoding of our data, D, given this hypothesis.

In part (a) of our message all values for the number of classes, N, are considered equally likely so stating N is assumed to have some unknown constant cost. As this calculation is only used to compare models we can safely omit this as part of our message length.

The length of part (b) of our message is the cost of sending the description of a multi-state distribution which could be used to assign each observation (K in all) to a particular class (N possibilities).

The code length required to describe a class, c_i, can be closely approximated as the sum of the optimal code lengths required to state the parameters describing each attribute. The message length of part (c) of our message is the sum of these individual class message lengths and additionally another N multi-state distributions specifying the class distribution for the next observation given the class of this observation. Each of these additional multi-state distributions encodes a proportion of the total number of observations as specified by the class relative frequency stated in part (b).

One caveat of note in this current implementation of tSnob is that this calculation is a conservative encoding of the N^2 transition matrix (the optimal message is slightly shorter). An optimal encoding of this transition matrix is not known to the authors. However, one should be able to save something like one row of the matrix. Specifically our estimate is calculated assuming independence between rows in the transition matrix. This is clearly not the case (there are fewer degrees of freedom), however, assuming independence yields a close approximation that has slight bias toward a more conservative model.

Once parts (a), (b) and (c) which constitute our hypothesis, H, have been transmitted we can transmit part

(d), the actual observations we have, by selecting a class for each observation and encoding each observation accordingly. An observation is coded as the sum of the optimal encoding for each attribute value using the stated class with missing attribute values coding as zero length messages (i.e. the receiver is assumed to know a-priori which attributes are missing).

The class of the first observation is specified using the unconditional multi-state distribution stated in part (b) of the message and the class of each successive observation is specified using the appropriate conditional multi-state distribution as stated in part (c) of the message (i.e. based on the class of the preceding observation). In this way it is possible to calculate the length of one possible decodable message.

As the only messages we consider are prefix codes (i.e. uniquely decodable) we can, for notational convenience, define a mapping from message lengths to probabilities

$$P_{ml}(x) = \mathrm{e}^{-ML(x)}$$

where $P_{ml}(x)$ is the probability that we will send a message, x, of length $ML(x)$ (in nits).

We can consider any one assignment of observations to classes as a path through our data (see Figure 1) and note that with N classes and K observations there are N^K such paths. A message stating one such specific path will not be an optimal encoding of the data given the model. However, we can now calculate the probability (and hence the length) of the optimal message by summing over all of these N^K sub-optimal encodings.

Summing these N^K probabilities appears to be a formidable task. However, if we consider our encoding process as a state machine, we find that our model is left in only one of N possible states after the encoding of any observation. So for any observation, we can calculate the sum over all paths that lead to one of our N states based on the N sums calculated for the preceding observation.

We define $P_{ml}(c_i \mid c_j)$ to be the probability associated with a message stating that an observation from c_i follows an observation from c_j and $P_{ml}(o_k \mid c_i)$ to be the probability associated with a message stating the attribute values associated with observation k using class i.

If we now define $F(o_k \in c_i)$ to be the sum over all paths (messages) that lead to an encoding of o_k as a member of class c_i then

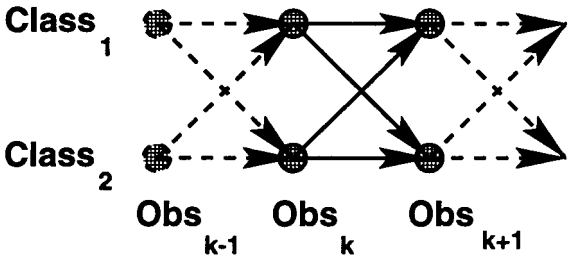

Fig. 1. *Possible observation encodings given 2 classes*

$$F(o_1 \in c_i) = P_{ml}(c_i) \cdot P_{ml}(o_1 \mid c_i)$$

$$F(o_k \in c_i) = \sum_{j=1}^{N} F(o_{k-1} \in c_j) \cdot P_{ml}(c_i \mid c_j) \cdot P_{ml}(o_k \mid c_i)$$

for $1 < k \leq K$ and finally

$$P(D \mid H) = \sum_{i=1}^{N} F(o_K \in c_i)$$

which is the sum over all the possible encodings of our data given our model.

The message length of parts (a), (b) and (c) give us $P(H)$, and we can calculate $P(D \mid H)$ so we now have $P(H \,\&\, D)$. This is the objective function that the tSnob program maximises by minimising it as a message length.

It is important to note that the class transition matrix is part of our hypothesis and thus will be penalised on information-theoretic grounds for it's complexity. The best class structure found using the Markov modelling may well differ markedly from the best class structure without the Markov modelling (in the number of classes and the specific class parameters). Thus a model obtained using this program will, most likely, be quite different from a model obtained by first doing a classification that assumes independence between observations and then observing the class transitions (i.e. partitioning the problem space into two independent sub-problems).

4. Searching the model space

In this section we define the un-supervised search of the first order Markov classification model space implemented in the tSnob program. The complexity of this first order Markov classification model space increases dramatically with the number of classes as does the probability of finding locally optimal solutions.

Our search of this model space attempts to avoid these local optima by limiting the complexity of our model (the number of classes) at any one time to that justified by our MML information measure. To this end we divide the search into two subproblems. Searching for the best parameterisation of a model with N classes, and finding the best value for N.

4.1. *Improving the parameter estimates*

We improve the model parameterisation given a particular class structure by the repeated application of an EM re-estimation step. In order to apply the EM algorithm on this problem it is necessary to consider the optimal assignment of each observation to the N classes independently of the assignment of any of the other observations in the sequence. In fact we wish to calculate the sum over all

the N^{K-1} possible encodings of our dataset that specify any one of the N states for any particular observation.

To achieve this we define a backward sum over all possible paths that lead from a classification of c_i for observation o_k to the end of the data sequence.

$$B(o_K \in c_i) = 1$$

$$B(o_k \in c_i) = \sum_{j=1}^{N} P_{ml}(c_j \mid c_i) \cdot P_{ml}(o_{k+1} \mid c_j) \cdot B(o_{k+1} \in c_j)$$

for $0 < k < K$. We can now define the contribution of the class c_i for observation o_k to the final $P(D \mid H)$ to be

$$P(D \mid H, o_k \in c_i) = F(o_k \in c_i) \cdot B(o_k \in c_i)$$

and note that

$$P(D \mid H) = \sum_{i=1}^{N} P(D \mid H, o_k \in c_i)$$

for $0 < k \leq K$.

Once we have calculated these N sums for any particular observation we can calculate the relative contribution of each of the N states to the encoding of the entire sequence, $P(D \mid H)$. With this information we can correctly re-estimate all the class distribution and transition parameters. This calculation differs from the usual forward-backward maximum log likelihood calculation in that the appropriate MML message lengths used may also include small penalty terms which depend on the accuracy to which the corresponding distribution was specified in the hypothesis.

4.2. *Selecting the best number of classes*

In order to select the best number of classes we employ a variation on the class splitting and merging search procedure implemented in the original snob program described in Wallace (1990). At any one time we consider a specific N class model and we move toward the best solution in this model space. However, it turns out that by calculating this we can also easily search a useful subset of the $N+1$ and $N-1$ class models. If a model in either subset turns out to be more likely than our current N class model then we switch our focus, N, to the better model space. In this way a simple model will shift to the more complex hypothesis spaces only when this is justified by the MML objective function. Having an accurate information measure to guide this shift in focus is essential to get good initial parameter estimates in the more complex model spaces and thus avoid the worst of the locally optimal solutions.

We consider one $N+1$ class model and one $N-1$ class model at any one time. These models are constructed from the current N class model and given a limited number of improvement cycles in which to yield a better solution than the N class model. If a better solution is not soon found then alternative $N-1$ and $N+1$ models are constructed. The selection of candidate models in these other model spaces is guided by message length estimates based on the current N class model. These estimates give a upper bound on the true message length in the alternate model spaces.

We calculate N estimates in the $N+1$ model space. These being where any of the current N classes is split to form two new classes while all the other $N-1$ classes are kept the same. This is achieved by maintaining a hidden two class split model for each of the N classes. These split models are initialised by a random assignment of the observations from the corresponding model class and then re-estimated on each pass of the dataset. To speed up this re-estimation process the observations are assigned to one or the other split class for the first three cycles and thereafter they are assigned probabilistically by EM. The split models are also periodically re-initialised in order to search for different asymmetries in the data. These N hidden two class mixture models are used to guide the selection and initialisation of candidate $N+1$ class models.

We calculate ${}^{N}C_2$ estimates in the $N-1$ class model space. These being where any two classes are combined into one class while all the other classes remain unchanged. These estimates can be derived by adding the observation statistics for candidate merge classes and then calculating the revised expected message length of the new class.

This class splitting and merging search differs from that of the original Snob program in that the message length estimates are only used to select candidate split or merge models. The complete model message length evaluation is still required before such a model is selected as the new focus for the search.

When we split or merge classes to generate a new model, care must be taken that the starting values for the class transition probabilities are reasonable. We first consider the case where class, c_i, is split to form two new classes, c_i' and c_i'' with the weight given to each class defined by a probability, w ($0 < w < 1$), which is typically 0.5. The relative abundances of these two new classes are defined as

$$P(c_i') = P(c_i) \cdot w$$
$$P(c_i'') = P(c_i) \cdot (1 - w)$$

The conditional probabilities independent of c_i remain unchanged. The new conditional probabilities required are defined as

$$P(c_i' \mid c_j) = P(c_i \mid c_j) \cdot w$$
$$P(c_i'' \mid c_j) = P(c_i \mid c_j) \cdot (1 - w)$$
$$P(c_j \mid c_i') = P(c_j \mid c_i'') = P(c_j \mid c_i)$$
$$P(c_i' \mid c_i') = P(c_i' \mid c_i'') = P(c_i \mid c_i) \cdot w$$
$$P(c_i'' \mid c_i') = P(c_i'' \mid c_i'') = P(c_i \mid c_i) \cdot (1 - w)$$

where $j \neq i$. This is the starting point for our search in the $N+1$ model space.

We now consider the case where classes, c_i and c_j, are merged to form a new class c_{ij}. The initial relative abundance of this new class is defined as

$$P(c_{ij}) = P(c_i) + P(c_j)$$

The conditional probabilities independent of c_{ij} remain unchanged and the new conditional probabilities are defined as

$$P(c_{ij} \mid c_k) = P(c_i \mid c_k) + P(c_j \mid c_k)$$

$$P(c_k \mid c_{ij}) = \frac{P(c_i) \cdot P(c_k \mid c_i) + P(c_j) \cdot P(c_k \mid c_j)}{P(c_{ij})}$$

$$P(c_{ij} \mid c_{ij}) = \frac{P(c_i)}{P(c_{ij})}(P(c_i \mid c_i) + P(c_j \mid c_i)) + \frac{P(c_j)}{P(c_{ij})}(P(c_i \mid c_j) + P(c_j \mid c_j))$$

where $k \neq i$ and $k \neq j$. This is the starting point for our search in the $N-1$ model space.

From these starting points we improve the parameter estimates of these other models and if one model then turns out to be more likely than our current N class model, we switch our focus, N, to the better model space.

Practically speaking, the repeated application of these two model search methods is an effective search strategy (the EM algorithm may of course converge to a local minimum).

5. Experimental results

In this section we compare classification models with and without the first order Markov modelling on a variety of 'synthetic' two class datasets. We also consider a difficult 'real-world' dataset.

We compare model compression and parameter estimation on three models: the 'one class' (1C) model (this result is the same with or without Markov modelling and corresponds to the null hypothesis), 'two class' (2C) model (the best two class standard classification model (assuming independence between observations)), and the 'Markov two class' (M2C) model (the best two class Markov classification model).

The compression and estimated parameter values shown are the median values as determined from 100 separate trials on independently generated datasets.

5.1. *Generating synthetic data*

The test data has been generated from a simple two class model with the observations having a single continuous attribute specified to a measurement accuracy of 0.05.

Three parameters describe the datasets generated: the class separation (s), the class auto-transition (t), and the dataset size (K). The class separation, s, is defined by fixing one class, $N(0,1)$, and varying the mean of the other, $N(s,1)$. The class auto-transition parameter, t, specifies the probability of generating an observation from the same class as that of the preceding observation, $P(o_k \in c_i \mid o_{k-1} \in c_i)$. Finally, K, is simply the number of observations generated.

The test data is pseudo-randomly generated from a model with these parameters but, any particular dataset thus generated, may well be slightly better described by another similar model.

5.2. *Comparing model compression*

We compare the compression obtained by each of the three models when varying each of the three parameters.

In Fig. 2 the auto-transition parameter, t, is varied while the dataset size, K, is held constant at 1000 observations and the class separation, s, is held constant at 2.0 sd.

The performance of the 1C and 2C models is approximately equivalent (i.e. more observations are required to choose the 2C model over the 1C model) and, as one would expect, their performance is independent of the auto-transition parameter, t, in the generating model.

When $t = 0.5$ we expect no benefit to be gained from using the Markov Model (as adjacent observations are completely independent) and indeed we find that M2C model yields slightly worse compression than the other models (about 0.02 bits/observation). However, for $t < 0.4$ and $t > 0.6$ the M2C model is the preferred hypothesis.

When $t = 0.0$ the M2C model gives a saving of about 0.5 bits per observation. The most that we could ever expect to save with two class data is 1 bit per observation, so 0.5 bits is not bad considering the significant overlap between classes separated by only 2.0 sd.

So, even in this extreme case where there is significant overlap between the classes and it is unclear as to whether the best standard model has one or two classes, we find that the penalty for using the M2C model on truly independent

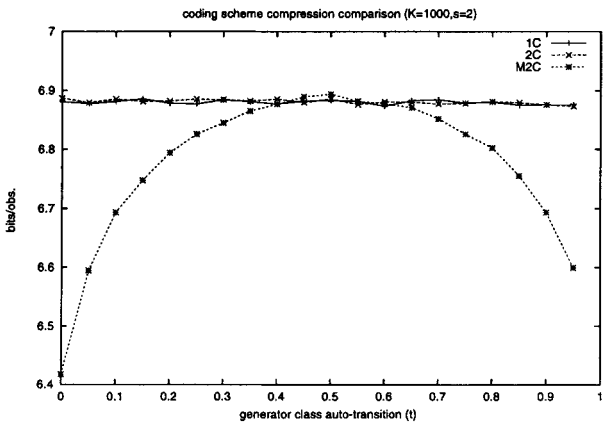

Fig. 2. *Compression vs. auto-transition*

data is only slight and that when there is a modest dependence between adjacent observations the M2C model quickly becomes the most probable.

In Fig. 3 the class separation, s, is varied while the dataset size, K, is held constant at 1000 observations and the auto-transition parameter, t, is held constant at 0.8.

When $s = 0.0$ sd (i.e. really one class) the 1C model is the most probable (as one would expect). For $s < 1.2$ sd (i.e. approximately one class) 1C model is still the preferred model for this moderately sized dataset. Although seemingly competitive for these small values of s the M2C yields a longer message due to extra class and transition parameters stated in the hypothesis (22 bits longer than the 1C model for 1000 observations when $s = 0.0$ sd). For $s > 1.2$ sd the M2C model is the most probable. It is not until $s > 2.1$ sd that the 2C model becomes significantly more probable than the 1C model.

In this instance the 2C model is never the best of the three and the M2C model is preferred over the 1C model with about 0.9 sd less separation between the generating classes than that required by the 2C model.

In Fig. 4 the dataset size, K, is varied while the class separation, s, is held constant at 2.0 and the auto-transition parameter, t, is held constant at 0.8.

For $\log_{10} K < 2.3$ ($K < 200$) the 1C model is clearly the most probable. Above 200 observations the M2C model is most probable of the three. The 2C model is preferred over the 1C model at about $\log_{10} K > 3.2$ ($K > 1580$).

So the M2C model is preferred over the 1C model with nearly an order of magnitude less data than would be required by the 2C model.

5.3. *Evaluation of parameter estimates*

In the following figures only results from the M2C model are presented. The estimated auto-transition parameter is defined as the auto-transition probability found by the program for $class_1$ (i.e. $P(o_k \in c_0 \mid o_{k-1} \in c_0)$) with the other three transition probabilities being ignored. The median, and the 10th and 90th percentiles for the values of this parameter over 100 trails are also shown.

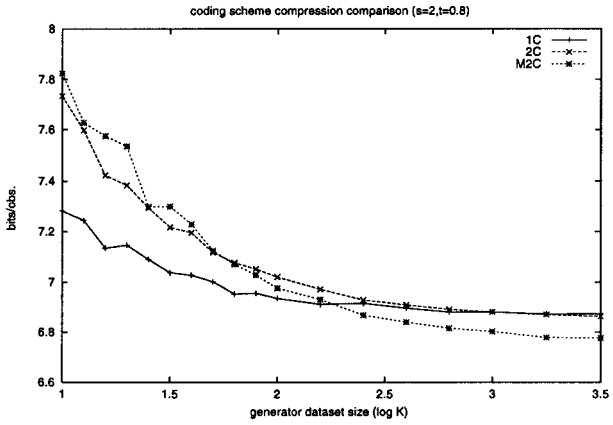

Fig. 4. *Compression vs. log_{10} dataset size*

In Fig. 5 the auto-transition parameter, t, is varied while the dataset size, K, is held constant at 1000 observations and the class separation, s, is held constant at 2.0 sd.

The median of the estimated auto-transition parameters is the same as the true value in the generating model (i.e. unbiased). The larger variance in our predicted value when $t \approx 0.5$ is to be expected as less accuracy is required when stating the model parameter at this point (i.e. slightly different generating models will produce similar data). This illustrates an interesting point about MML which will encode this model parameter at a reduced cost in this same region.

In Fig. 6 the class separation, s, is varied while the dataset size, K, is held constant at 1000 observations and the auto-transition parameter, t, is held constant at 0.8.

When $s = 0.0$ sd (i.e. really one class) the median auto-transition estimate is 0.5 (i.e. independence between observations). For $s > 1.5$ sd the median estimate for, t, is 0.8 (unbiased around the generating model). Incidentally,

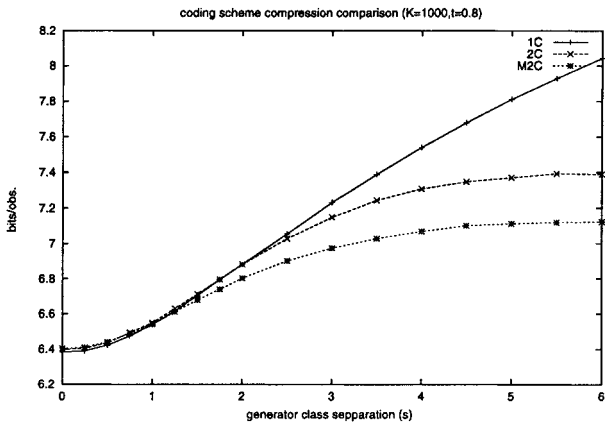

Fig. 3. *Compression vs. class separation*

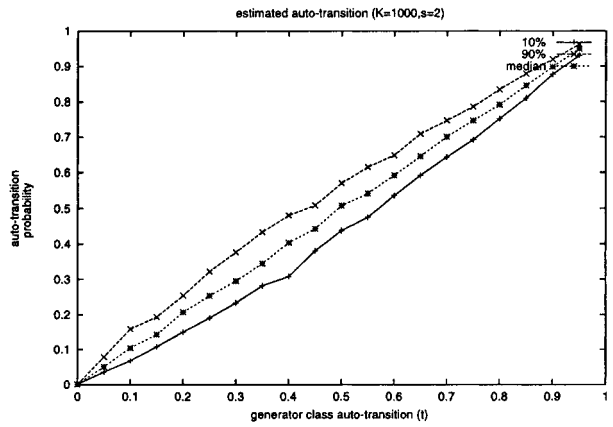

Fig. 5. *Estimated auto-transition vs. model auto-transition*

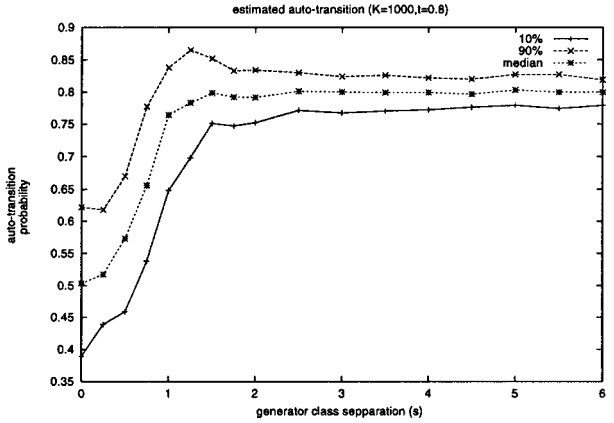

Fig. 6. *Estimated auto-transition vs. class separation*

this is the same point as the M2C model is preferred over the 1C model (see Fig. 3).

In Fig. 7 the dataset size, K, is varied while the class separation, s, is held constant at 2.0 sd and the auto-transition parameter, t, is held constant at 0.8.

At a $\log_{10} K = 1.0$ ($K = 10$) the median estimate for, t, is about 0.56 (i.e. the observations are best considered as being near independent). The median estimate for t asymptotes to 0.8 (same as in the generating model) as K increases. For $\log_{10} K > 2.4$ ($K > 250$) no significant further improvement in the median estimate occurs although the variance of the estimate continues to diminish. Note that for $K < 250$ it is not surprising, given the class separation of 2.0 sd, that many of the datasets generated do not provide enough evidence to accurately recover the transition parameter of the generating model.

5.4. *EM settling time comparison*

The settling time is the number of EM algorithm iterations required to minimise the models objective function with the stopping criteria defined as: no significant change (less than 0.01 bits) in message length in the next 30 iterations. The last iteration that produced a change in message length is taken as a measure for the settling time for the model on the given data.

The three models (i.e. 1C, 2C, M2C) are again plotted in the three figures that follow, however, the 1C model has a constant settling time of 1 iteration.

In Fig. 8 the auto-transition parameter, t, is varied while the dataset size, K, is held constant at 1000 observations and the class separation, s, is held constant at 2.0 sd. The 2C model takes no account of the order in the dataset and so the settling time is of constant cost. The M2C model takes slightly longer to settle for $0.15 < t < 0.75$, but is significantly faster otherwise.

In Fig. 9 the class separation, s, is varied while the dataset size, K, is held constant at 1000 observations and the auto-transition parameter, t, is held constant at 0.8. Both methods settle more rapidly as, s, increases. For $s > 2.5$ sd the M2C model settles in about half the iterations required by the 2C model.

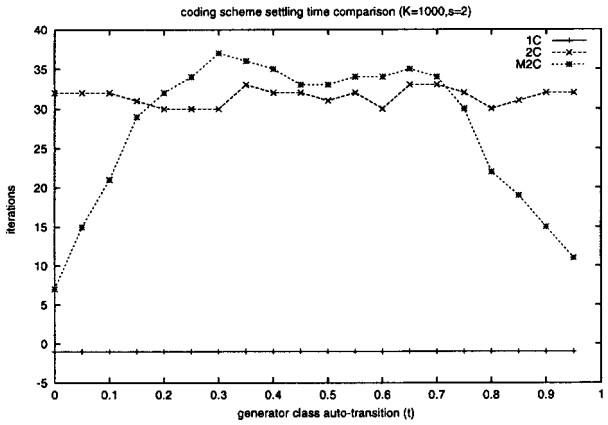

Fig. 8. *Settling time vs. auto-transition*

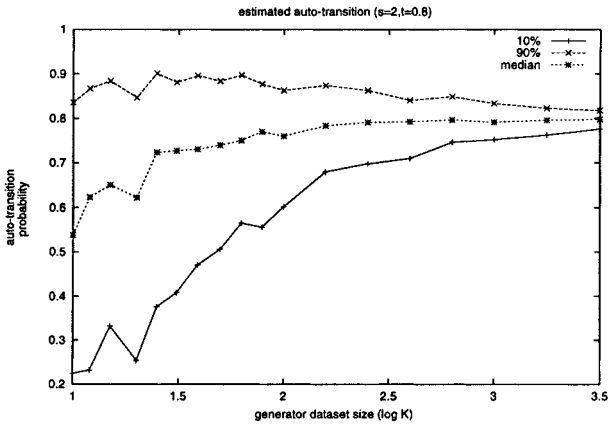

Fig. 7. *Estimated auto-transition vs. $\log_{10}$ dataset size*

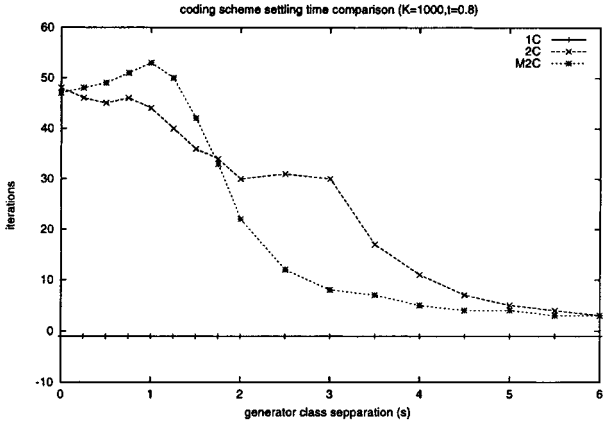

Fig. 9. *Settling time vs. class separation*

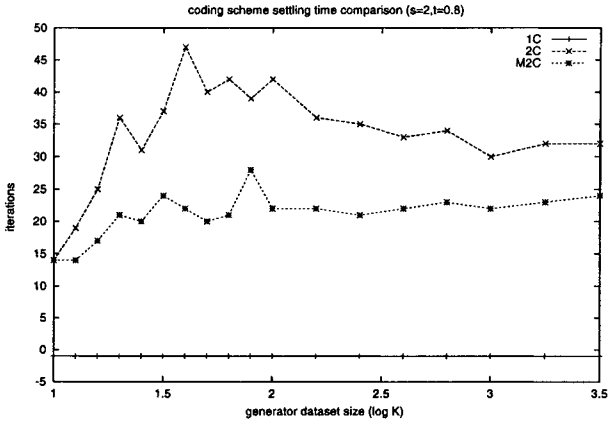

Fig. 10. *Settling time vs. log_{10} dataset size*

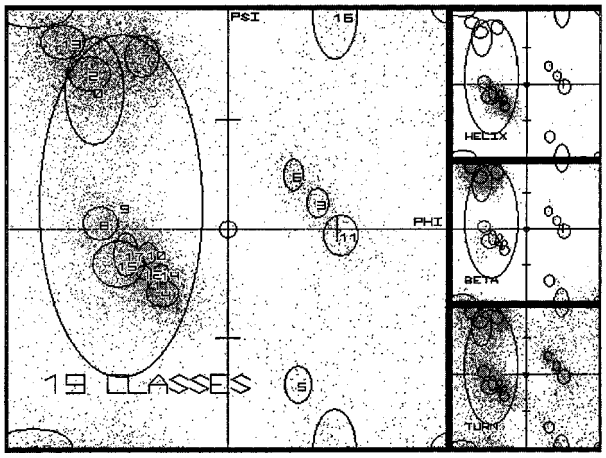

Fig. 11. *19 class protein structure model*

In Fig. 10 the dataset size, K, is varied while the class separation, s, is held constant at 2.0 sd and the auto-transition parameter, t, is held constant at 0.8. We find that M2C model utilises the extra auto-transition information and converges faster than the 2C model in all cases.

5.5. *Real world data*

In order to evaluate the algorithms performance on more a complex problem space a difficult 'real-world' dataset of protein structure data was selected. The protein dataset selected consists of 41731 pairs of protein dihedral angles (ϕ, ψ). Secondary structure classification of such data is of significant interest in the area of protein modelling.

The angle pairs are constructed from approximately 230 separate proteins as detailed in Edgoose *et al.* (1998) and the program was modified to encode each protein segment independently (i.e. the assignment of the first observation in a protein to a class is encoded from the un-conditional class distribution). The von Mises distribution was used to model both the ϕ and ψ angle attributes.

The best Markov classification model found had 19 classes and a message length of 266 973 bits. This 19 class structure is shown in Fig. 11 with each class depicted by an ellipse with centre (μ_ϕ,μ_ψ) and dimensions $(\frac{1}{\sqrt{\kappa_\phi}},\frac{1}{\sqrt{\kappa_\psi}})$. The actual observations are overlaid to create a scatter plot which is a square depiction of the surface of a torus and hence wraps around at the edges.

The class model found correlates well with known biological structures such as Helix, Beta-sheet and Turn with different flavours of each represented in the different classes. Of particular interest were some classes of statistical significance due to their relationship with their neighbours in the sequence. The classes 10 and 14 were found to occur only at the transition between a run of Beta and a run of Helix residues. Class 9, the most populous class, was found to have very small kappa values (i.e. a large variance), but a high probability of occurring in runs. The explanation seems to be that certain small regions of the protein are quite flexible and take on a structure almost entirely determined by non-local residue interactions. For further discussion refer Edgoose *et al.* (1998).

The best standard classification model (i.e. assuming independence between observations) found 27 classes with a message length of 294 700 bits.

Thus, the Markov classification model found represents a 9.4% improvement in terms of compression as well as having a simpler class structure. The search procedure for the Markov model space was found to be effective and consistent on this large and complex 'real-world' dataset.

6. Conclusion

We have extended the MML un-supervised classifier Snob to model ordered datasets where the best classification of an observation need not be independent of the classification of neighbouring observations. Specifically we model the data as if it had been generated from a first order Markov process with the state at any point specifying the class of the corresponding observation. Such a model is commonly referred to as a Hidden Markov Model.

We define a near optimal information measure for the cost of stating such a model and a set of observations given the stated model. This gives an objective criteria by which we can judge two competing models which differ in the numbers of classes they contain given a specific dataset. This measure is used to guide a robust un-supervised search of the Markov classification model space that correctly balances model complexity against explanatory power.

Experimentally it has been shown that the MML information measure for the Markov classification model yields improved class model selection results when modelling sequential datasets even where there is only a modest dependence between adjacent observations. This MML

implementation is shown to yield unbiased estimates of model parameters and the number of EM iterations required to search the more complex Markov model space is not significantly different from the number required for the standard classification model.

Finally, the Markov classification model has been used with consistent success on a large and difficult 'real-world' protein dataset indicating that the search heuristics are effective and the model search robust.

Acknowledgments

Special thanks to Rohan Baxter for discussions and thanks also to David Dowe for assistance with the implementation of the von Mises distribution.

References

Laird, N. M., Dempster, A. P. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* (*Series B*), **39**, 1–22.

Soules, G., Baum, L. E., Petrie, T. and Weiss N. (1970) A maximisation technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Mathematical Statistics*, **41**, 164–171.

Cheeseman, P. C. (1988) Autoclass II conceptual clustering system. *Proceedings Machine Learning Conference*, pp. 54–64.

Edgoose, T., Allison, L. and Dowe, D. L. (1998) An MML Classification of Protein Structure that knows about Angles and Sequence. *Forthcoming in the Proceedings of the 3rd Pacific Symposium on Biocomputing*.

Fisher, N. I. (1993) em Statistical Analysis of Circular Data. Cambridge University Press, Cambridge.

Leroux, B. G. and Puterman, M. L. (1992) Maximum-Penalized-Likelihood Estimation for Independent and Markov Dependent Mixture Models. *Biometrics*, **48**, 545–558.

Rabiner, L. R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257–286.

Rissanen, J. (1987) Stochastic complexity. *Journal of the Royal Statistical Society* (*Series B*), **49**, 223–239.

Wallace, C. S. (1990) Classification by minimum length inference. *AAAI Spring Symposium on the Theory and Application of Minimum Length Encoding*, *Standford*, pp. 5–9.

Wallace, C. S. and Boulton, D. M. (1968) An information measure for classification. *Computer Journal*, **11**, 185–194.

Wallace, C. S. and Freeman, P. R. (1987) Estimation and inference by compact coding. *Journal of the Royal Statistical Society* (*Series B*), **49**, 240–252.

Wallace, C. S. and Dowe, D. L. (1994) Estimation of the von Mises concentration parameter using Minimum Message Length. In proceedings of the Twelfth Australian Statistical Society Conference, Monash University, Melbourne, Australia.