An introduction to Minimum Message Length inference

David L. Dowe
School of Computer Science and Software Engineering,
Monash University, Clayton, Vic. 3168, Australia

e-mail: dld@cs.monash.edu.au
WWW: http://www.csse.monash.edu.au/~dld/

April 11, 2006

7 Some re-capping and revision

7.1 Invariance of MML under 1-to-1 re-parameterisation

As[1] has[2] been mentioned before and as will be repeated below, MML is invariant under 1-to-1 re-parameterisation. This holds not just for 1-dimensional estimation problems, but also for general n-dimensional estimation problems.
One can show invariance formally mathematically, but a basic outline is as follows.
First recall from Section 4.3, that the MML estimate of $\vec{\theta}$ is the value of $\vec{\theta}$ which maximises

$$h(\vec{\theta})f(x|\vec{\theta})/\{F(\vec{\theta})\}^{1/2}.$$

The likelihood function, $f(x|\vec{\theta})$ [3], is invariant under 1-to-1 re-parameterisation. A bit less obvious is that fact that $h(\vec{\theta})/\{F(\vec{\theta})\}^{1/2}$ is also invariant under 1-to-1 re-parameterisation. Although one should formally show this mathematically, we note that $h(\vec{\theta})$ has the dimensions of "per something", or " something^{-1} " . E.g., if θ were to be a length in metres,

[1] This document contains some revisions of earlier material by David Dowe alone or by David Dowe and Graham Farr (variously entitled 'Introduction to Minimum Encoding Inference', 'An introduction to MML Inference', etc.); the rest is new. Typographical feedback on that earlier material from Hons and other students in 1997-1998 are gratefully acknowledged.

[2] As Graham Farr has understandably revised some of his earlier 1997-1998 material, I apologise in advance for any subsequent out-of-date and incorrect cross-references to page numbers (in particular), section numbers or equation numbers in the material distributed by him in 1997-1998.

[3] or $p(x|\vec{\theta})$ if we accidentally confuse notation.

then $h(\vec{\theta})$ would have the dimensions of per metre, or m^{-1}. $F(\vec{\theta})$, being the expectation of a second derivative, has the dimensions of a second derivative, namely " per something^2 ", or " something^{-2} ". So, $F(\vec{\theta})^{1/2}$ has the dimensions of "per something", the same as $h(\vec{\theta})$. This at least tells us that $h(\vec{\theta})/\{F(\vec{\theta})\}^{1/2}$ is dimensionless, which is a necessary condition for being invariant under 1-to-1 re-parameterisation.

In fact, it turns out (H. Jeffreys, 1946) that $F(\vec{\theta})^{1/2}$ has the same mathematical form as a prior, and indeed that $h(\vec{\theta})/\{F(\vec{\theta})\}^{1/2}$ is invariant under 1-to-1 re-parameterisation.

As well as being invariant, we also see that the likelihood function and the log-likelihood function are dimensionless.

Since $f(x|\vec{\theta})$, $L = -\log f(x|\vec{\theta})$ and $h(\vec{\theta})/\{F(\vec{\theta})\}^{1/2}$ are invariant, it therefore follows that $h(\vec{\theta})f(x|\vec{\theta})/\{F(\vec{\theta})\}^{1/2}$ and $-\log(h(\vec{\theta})f(x|\vec{\theta})/\{F(\vec{\theta})\}^{1/2})$ are invariant.

More complete expressions for the message length, such as on section 4.4, involve dimensionless constants (such as the number of parameters, n).

It therefore follows that the message length, and the minimum of the message length, are invariant under 1-to-1 re-parameterisation.

A sketchy alternative argument for the invariance of MML appeals to directly to information theory, saying that information and information content are independent of the framing of the problem, and so the message length and its minimum should not change just because we transform the parameter axes.

7.2 Interpreting the Fisher information in MML estimation

7.2.1 Interpreting the Fisher information in one dimension

Recalling Section 4.3, we have that the optimal "spacing" or precision to which we state our parameter estimates is given by

$$s(\vec{\theta}) \;\propto\; 1/\{F(\vec{\theta})\}^{1/2}.$$

For a problem where we only wish to estimate one parameter, the Fisher information qualifies an intuitive wish to choose the posterior mode of the likelihood maximum. The second derivative of a function measures how quickly the first derivative is changing, or how tight a bend or peak is. The Fisher information is an expected second derivative. If the Fisher information is large, then our uncertainty region is small, and it makes sense to aim for a tight peak. If the Fisher information is small, then it is difficult to justify aiming for the highest point in the likelihood or the posterior without considering that we are looking for a broad peak.

7.2.2 Interpreting the Fisher information in several dimensions

Let us suppose now that we have a parameter estimation problem in several dimensions. If the variables are independent, then so, too, will be the parameters being estimated. This will result in the Fisher information matrix being diagonal, and so its determinant will simply equal the product of the diagonal elements.

On the other hand, it is possible that, instead, the variables are highly correlated, or have a high degree of collinearity. Such high interdependence will intuitively make the uncertainty region large and the Fisher information small, because changing the value of one parameter could still get us to an almost identical point in the likelihood function if we changed the values of some highly correlated variables - sort of like how a flattenned rhombus has a smaller area than a square with the same side length. In case it is a bit confusing to try and picture the meaning of the Fisher information when the Fisher information matrix is not diagonal, the invariance of MML under 1-to-1 re-parameterisation is useful.

If the Fisher information matrix is not diagonal but the determinant is non-zero, then a suitable 1-to-1 transformation will take us to a parameter space (say $\theta' \epsilon \Theta'$) where the Fisher information matrix now is diagonal. We can in principle do the MML estimation in this space and then, by invariance, transform back to the original problem, $\theta \epsilon \Theta$.

7.3 Motivation of MML and other estimation methods

Statistics, econometrics, machine learning, "data mining" and other disciplines are concerned with trying both to model the world around us based on any observations we have made or any data we have observed and to predict the future based on these observations. Many other disciplines are also concerned at least to some degree with those objectives of modelling the world and predicting the future.

7.3.1 Relevance of modelling data

Given how vast the number of areas is in which we wish to model or predict, let us list but a few which people might do professionally at work, recreationally or as part of their private lives:
Economics, financial trading strategies, optimal portfolio balance, trying to predict the housing market in your preferred part of town, trying to predict the value of the Aus$ before your next overseas holiday.
Meteorology, weather, safety of fishing on the bay, safety of an aeroplane taking off if it has to land somewhere overcast in one hour, choosing when to declare a cricket innings in light of incumbent weather.
Astronomy, trying to infer the formation of the solar system.
Physics, collecting data to try to infer various physical laws.
Medicine, psychology, trying to infer causal effects of smoking, trying to infer relevance

of various drugs to treat various conditions, trying to infer good diet, exercise and environmental options for a long and healthy life.
Bushfire prediction and protein structure prediction.
Trying to infer an opponent's poker strategy.
Observing an expert in some discipline who can't or won't tell us what they do, but whose method we hope to eventually learn to some degree.
Clustering and mixture modelling, so as to find clusters within proteins.
Compressing and aligning DNA either to discover genes or to discover ancestry.
Trying to identify the authorship of an ancient art work.
Trying to find a simple but useful model of how chess-players play.
Coming up with a better ranking system for (e.g.) tennis players.

Although modelling the world and predicting the future are related and very similar, we emphasise that these tasks are not identical.

7.3.2 Issues in modelling data

The discipline of fitting models to data and prediction entails more than just running a data-set through some software package. This is true in part because, as we have argued, fitting models (inference) and prediction are not identical. It is also true because statistical modelling in the 1990s is far from unanimously "solved". Many classical point estimation techniques abound which we would contend are both philosophically and empirically flawed — and we will provide examples to advocate this point. One could go as far as saying that before using some statistical, econometric, machine learning or "data mining" software, the data analyst will do well to first determine whether she indeed feels that this software is likely to produce a reliable result.
Sometimes we wish to infer just one model or one set of parameter estimates from the data. For example, some data comes from a Gaussian distribution with mean, μ, and standard deviation (s.d.), σ, such as, e.g. :
$x_1 = 1.2$, $x_2 = 4.0$, $x_3 = 3.7$, $x_4 = 5.6$;
and we wish to estimate μ and σ. Such attempts to summarise the data by the values of a few parameters are known as *point estimates*. The Maximum Likelihood (ML), Minimum Message Length (MML), minimum Expected Kullback-Leibler distance (MEKLD) and posterior mean estimators are all point estimates.

It should be noted, though, that there is a school of Bayesian statisticians who essentially refuse to do point estimation, claiming that their work is essentially finished when they have calculated the functional form of the posterior distribution.
This position seems not altogether unreasonable when considers *loss functions*, as in the following example :

Consider a company which holds stock of some good in a warehouse. The company has a cost in either depreciation or storage space rental for every stock item it holds in warehouse storage, but it loses more in terms of disgruntled customers and lost sales if an

order comes in when the company is out of stock. Rather than use a point estimate (such as Maximum Likelihood, MML or MEKLD) for the expected number of customer orders to come in during a month, the company might do well to have a (posterior) probability distribution on the expected number of customer orders and then to choose an amount optimising the expected company profit.

7.3.3 Desirable features of point estimation methods

Two desirable features for an estimation method are *invariance* under 1-to-1 parameter transformations and *consistency*. One can also argue cases for other desirable properties of estimators — such as performing well on small sample sizes — but, for the time being, we consider just these two.

Invariance
We have discussed invariance in Section 7.1 (and possibly earlier). It basically says that if we transform a problem and then transform it back, we get the same estimator. So, if we are looking at a cube whose side length, l, and volume, V, we wish to estimate, we would like our estimator to return $\hat{V} = (\hat{l^3}) = (\hat{l})^3$.

If point estimation without a loss function is to mean anything at all, then invariance seems like a very reasonable property to require.

We recall that Maximum Likelihood and MML are invariant.

Question :

In the definition of invariance, why do we insist that parameter transformations have to be 1-to-1?

Exercise :

Show that the posterior mean is not invariant by
(i) finding a 1-1 transformation for which it is not invariant
(ii) arguing about the notion of dimension (see Section 7.1)

Exercise (fairly difficult):

Show that the MEKLD estimator is invariant.

Consistency
Informally, we say that an estimator is *consistent* for a certain problem if, as the amount of data grows arbitrarily large, the estimator will converge with probability 1 to the correct answer.

Whenever we use an estimation method, we would like to think that increasing the amount of available data will be expected to bring us closer to the underlying model. Furthermore, if the model generating the data is in the class of models that we are choosing from, we would like to think that increasing the amount of available data will permit our estimator to get arbitrarily close to the correct answer.

For problems where the number of parameters to be estimated is fixed, all the estimation methods discussed so far and many others will be consistent. However, many inference problems involve a very large number of variables which could increase as the amount of data increases. Mixture modelling (see Section 9) and factor analysis are cases in point, and these lead to inconsistencies in both Maximum Likelihood and the related Akaike Information Criterion (AIC) method, both of which are non-Bayesian.
We will later mention the Neyman-Scott problem, for which Maximum Likelihood and AIC are inconsistent but for which MML is consistent.

7.3.4 Invariance and consistency conjecture

Conjecture (Dowe, 1997) :[1][2, p 282]
Any estimation method which is universally both invariant and consistent must use a subjective Bayesian method.

Exercise (open research question, difficult):
Prove the above or find a counter-example.

Further Comments :
Strict MML (Wallace and Boulton, 1975) and MML (Wallace and Boulton, 1968; Wallace and Freeman, 1987) are subjective Bayesian methods shown to be invariant in these papers and also shown to be consistent (Wallace and Freeman, 1987; Barron and Cover, 1991; Wallace, 1996).

Maximum Likelihood and other classical methods are known ot have difficulties with problems where the number of parameters to be estimated increases with the sample size : e.g., the Neyman-Scott problem, factor analysis and fully-parameterised mixture modelling.

8 Applying MML to parameter estimation

Given all the claims that we have made about MML, it is time to apply the formulae from Sections 4.3 and 4.4 initially to the problem of single parameter estimation for a variety of distributions.

When we advance to problems of model selection, such as whether to use a mixture model with one component or two components, or whether to use a cubic, quadratic or constant polynomial, we use the message length as our metric.

MML is concerned with minimising the length of a two-part message, the first part of which states the hypothesis and the second part of which states the data given the hypothesis. Our model utilising more variables will have a more expensive theory than one with less variables, and will only be able to justify this by quantitatively being able to account for a saving of at least as much in the second part of the message.

For the Bernoulli, Gaussian, Poisson and Geometric distributions, we will see below that, for the chosen priors, the Maximum Likelihood (ML) and MML estimators are quite similar. For the multi-state Bernoulli distribution, we also note that the ML and MML estimators are similar to the posterior mean. However, the von Mises distribution is not so friendly, and we see the Maximum Likelihood estimator perform very poorly for small sample sizes.

8.1 Bernoulli distribution

8.1.1 Binomial distribution

Let $p_1 = p$ and $p_2 = 1 - p$ be the respective probabilities of the two outcomes of a binary Bernoulli trial.

If we were not interested in which particular outcomes were from class 1 and which particular outcomes were from class 2 but were only interested in the unordered cumulative total $(x, N - x)$ in both class 1 and class 2, then the likelihood function would be given by

$$f(x|p) \;=\; \binom{N}{x} p^x (1-p)^{N-x}$$

However, given that we are interested in the encoding of the particular individual outcomes, the likelihood function is given by

$$f(x|p) \;=\; p^x (1-p)^{N-x} \tag{49}$$

and

$$L \;=\; -\log f(x|p) \;=\; -x \log p - (N-x)\log(1-p)$$

$$\frac{\partial L}{\partial p} \;=\; -x \times 1/p \;+\; (N-x) \times 1/(1-p) \tag{50}$$

So, the "observed Fisher information", $F(x,p)$, is

$$F(x,p) = \frac{\partial^2 L}{\partial p^2} \;=\; \frac{x}{p^2} + \frac{N-x}{(1-p)^2}$$

So,

$$\begin{aligned} F(p) \;&=\; E_x F(x,p) \;=\; E_x \frac{\partial^2 L}{\partial p^2} \;=\; E_x \left(\frac{x}{p^2} + \frac{N-x}{(1-p)^2} \right) \;=\; \frac{E_x x}{p^2} + \frac{N - E_x x}{(1-p)^2} \\ &=\; \frac{Np}{p^2} + \frac{N-Np}{(1-p)^2} \;=\; \frac{N}{p} + \frac{N}{1-p} \;=\; \frac{N(1-p)+Np}{p(1-p)} \\ &=\; \frac{N}{p(1-p)} \end{aligned} \tag{51}$$

It follows from differentiating the log-likelihood, L, that $\hat{p}_{ML} = \frac{x}{N}$.
Let us now assume a uniform prior $h_p(p) = 1$.
To calculate the posterior mean, we first note in general that

$$\int_0^1 p^\alpha (1-p)^\beta \; dp \;=\; \frac{\alpha!\;\beta!}{(\alpha+\beta+1)!}$$

In calculating the marginal probability $r(x)$ of the unordered cumulative total x, we use the version of the likelihood function preceding equation (49) which[4] uses $\binom{N}{x}$.

$$r(x) = \int_0^1 h(p) f(x|p) \; dp \;=\; \int_0^1 1 \times \binom{N}{x} p^x (1-p)^{N-x} \;=\; \binom{N}{x} \frac{x!(N-x)!}{(N+1)!} \;=\; \frac{1}{N+1}.$$

So,

$$g(p|x) \;=\; \frac{h(p) f(x|p)}{r(x)} \;=\; (N+1) \binom{N}{x} p^x (1-p)^{N-x}$$

The posterior mean is thus

$$\begin{aligned} \int_0^1 p \; g(p|x) \; dp \;&=\; (N+1) \binom{N}{x} \int_0^1 p^{x+1} (1-p)^{N-x} \; dp \\ &=\; (N+1) \binom{N}{x} \frac{(x+1)!(N-x)!}{(N+2)!} \;=\; \frac{x+1}{N+2} \end{aligned}$$

Since $h(p)$ is uniform and

$$\frac{1}{\sqrt{F(p)}} \;=\; \frac{1}{\sqrt{N}} p^{1/2} \; (1-p)^{1/2}$$

[4] although the presence of this term will have no effect in the normalisation shortly to be used to calculate $g(p|x)$.

it should be an easy exercise to show that

$$\hat{p}_{MML} \;=\; \frac{x+1/2}{N+1}$$

Exercise :

Consider the mapping $\rho = m(p) = \frac{p}{1-p}$.

Verify that there is a 1-to-1 mapping from p to ρ.

Show that if p is the probability of a white ball and $1-p$ is the probability of a black ball, then $\frac{p}{1-p}$ is the (long term) ratio of white balls to black balls.

Suppose that x white balls are selected from N drawings. Assume a uniform prior on p.

Derive the posterior mean estimates of p and of ρ.
Is the posterior mean invariant for this problem?

8.1.2 Multinomial Bernoulli distribution

The above results generalise quite nicely, as we show in the exercises below. Let $p_1, p_2, \ldots, p_M$ be the respective probabilities of the M outcomes of a multinomial Bernoulli trial, with

$$p_1 + p_2 + \ldots + p_M \;=\; 1 \quad \text{and} \quad \mathrm{p_i} \geq 0$$

and suppose we observe x_i outcomes in state i.

Exercises:

Show that $(\hat{p}_i)_{ML} \;=\; \frac{x_i}{N}$.

Let $h(\vec{p})$ be the uniform prior $h(\vec{p}) \;=\; \frac{1}{(M-1)!}$ over the $(M-1)$-dimensional simplex.

Show that the posterior mean of p_i is $\frac{x_i+1}{N+M}$.

Show that $(\hat{p}_i)_{MML} \;=\; \frac{x_i+1/2}{N+M/2}$.

This distribution is useful not just for probabilistic prediction, but also for MML mixture modelling, clustering and unsupervised learning (Snob), MML decision trees and MML probabilistic finite state automata (PFSAs).

8.1.3 Kullback-Leibler distance between two Multinomial distributions

$$\sum_{i=1}^{M} p_i \log \frac{p_i}{q_i} \;=\; \sum_{i=1}^{M} p_i (\log p_i - \log q_i) \;=\; \sum_{i=1}^{M} p_i \log p_i - \sum_{j=1}^{M} p_j \log q_j \qquad (52)$$

8.1.4 Kullback-Leibler distance between two Binomial distributions

The Binomial distribution, (?), is a special case of the Multinomial distribution, corresponding to the case when there are two states.

$$\begin{aligned}
\sum_{i=1}^{2} p_i \log \frac{p_i}{q_i} &= \sum_{i=1}^{2} p_i (\log p_i - \log q_i) = \sum_{i=1}^{2} p_i \log p_i - \sum_{j=1}^{2} p_j \log q_j \\
&= p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q} \qquad (53)
\end{aligned}$$

where $p = p_1$, $p_2 = 1 - p_1 = 1 - p$, and $q = q_1$, $q_2 = 1 - q_1 = 1 - q$.
Given data x and a posterior probability distribution $g(p|x)$, what is the Minimum Expected Kullback-Leibler distance (MEKLD) estimator of p ?

8.2 Negative Binomial distribution

Not a lot to say just now :-) .

8.2.1 Kullback-Leibler distance between two Negative Binomial distributions

This problem is defined for $0 \leq p < 1$.

$$\begin{aligned}
&\sum_{i=0}^{\infty} \binom{i+r-1}{r-1} (1-p)^r p^i \log \frac{\binom{i+r-1}{r-1}(1-p)^r p^i}{\binom{i+r-1}{r-1}(1-q)^r q^i} \\
= \;& r \log \frac{1-p}{1-q} + (0 + (1-p)^r p \sum_{i=1}^{\infty} \binom{i+r-1}{r-1} i p^{i-1} \log \frac{p}{q}) \\
= \;& r \log \frac{1-p}{1-q} + (1-p)^r p \sum_{i=1}^{\infty} r \binom{i+r-1}{r} p^{i-1} \log \frac{p}{q} \\
= \;& r \log \frac{1-p}{1-q} + (\frac{rp}{1-p} \log \frac{p}{q}) \; (1-p)^{r+1} \sum_{i=1}^{\infty} \binom{i-1+(r+1)-1}{(r+1)-1} p^{i-1} \\
= \;& r \log \frac{1-p}{1-q} + (\frac{rp}{1-p} \log \frac{p}{q}) \; (1-p)^{r+1} \sum_{i=0}^{\infty} \binom{i+(r+1)-1}{(r+1)-1} p^{i} \\
= \;& r \log \frac{1-p}{1-q} + \frac{rp}{1-p} \log \frac{p}{q} \\
= \;& \frac{r}{1-p} (p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}) \\
= \;& \frac{r}{1-p} \times d_{KL}(Bin(p, 1-p), Bin(q, 1-q)) \qquad (54)
\end{aligned}$$

This problem is defined for $0 \leq p < 1$, and not defined for $p = 1$. For $p = 0$, $1 - p = 1$, and so the distance is r times that for the Binomial distribution.

8.3 Gaussian distribution

The Gaussian, or Normal, distribution, is used often in statistical modelling. It is specified by two parameters. These are μ, the location parameter or *mean*, specifying the middle of the distribution, and σ, the dispersion or *standard deviation*, which specifies the spread of the distribution.
The functional form is
$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((x-\mu)^2)}$,
and this is sometimes written $X \sim N(\mu,\sigma^2)$.
To carry out parameter estimation, we can do Maximum Likelihood quite straightforwardly.
MML estimation requires the use of a Bayesian prior distribution. MML also requires us to take expected second derivatives of the log-likelihood function, thus giving the Fisher information.

8.3.1 The Maximum Likelihood Estimator for the Gaussian distribution

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

$$\begin{aligned} L &= -\log\left\{\prod_{j=1}^{N} \frac{1}{(2\pi)^{1/2}\sigma} e^{-\frac{\frac{1}{2}(x_j-\mu)^2}{\sigma^2}}\right\} \\ &= \frac{N}{2}\log(2\pi) + \frac{N}{2}\log(\sigma^2) + \frac{1}{2\sigma^2}\sum_{j=1}^{N}(x_j-\mu)^2 \end{aligned} \tag{55}$$

Differentiating, we have

$$\frac{\partial L}{\partial \mu} = \frac{1}{2\sigma^2}\frac{\partial}{\partial \mu}\left\{\sum_{j=1}^{N}(x_j-\mu)^2\right\} = \frac{N\mu - (x_1+\ldots+x_N)}{\sigma^2} \tag{56}$$

and

$$\frac{\partial L}{\partial(\sigma^2)} = \frac{N}{2\sigma^2} - \frac{1}{2(\sigma^2)^2}\sum_{j=1}^{N}(x_j-\mu)^2 \tag{57}$$

Setting $\frac{\partial L}{\partial \mu} = 0$, we get

$$(\mu)_{ML} = \frac{x_1+x_2+\ldots+x_N}{N} = \bar{x}$$

and, setting $\frac{\partial L}{\partial(\sigma^2)} = 0$, we also get

$$\begin{aligned} (\sigma^2)_{ML} &= \frac{1}{N}\sum_{j=1}^{N}\{x_j - (\mu)_{ML}\}^2 \\ &= \sum_{j=1}^{N}\frac{(x_j-\bar{x})^2}{N} \\ &= \frac{s^2}{N} \end{aligned} \tag{58}$$

where $s^2 = \sum_{j=1}^{N} (x_j - \bar{x})^2$.

We continue, below, to derive the MML estimator.

8.3.2 The MML Estimator for the Gaussian distribution

Given parameters $\vec{\theta}$, (Bayesian) prior density $h(\vec{\theta})$, likelihood function p, and negative of log-likelihood function, $L = -\log f$ (considered in Section **??**), and expected Fisher information, F, the MML estimate of $\vec{\theta}$ is the value of $\vec{\theta}$ which maximises $h(\vec{\theta})p(x|\vec{\theta})/\{F(\vec{\theta})\}^{1/2}$.

We already have the likelihood function from the previous section, from which we can derive the Fisher information.

We must choose a prior on both μ and σ. It is fairly common practice in Bayesian statistics to assume that the prior in μ and the prior in σ are independent of one another.
We assume a uniform prior, h_μ on μ over some range, $[L_\mu, U_\mu]$; so $h_\mu(\mu) = \frac{1}{U_\mu - L_\mu}$ over this range. We assume a "conjugate" prior, $h_\sigma(\sigma) \propto 1/\sigma$, over some finite range. The effect of the conjugate prior is for it to make no difference whether we measure in centimetres, metres or kilometres.

All that remains before we obtain the MML estimate is for us to calculate the Fisher information, which we now do.

The Fisher Information for the Gaussian distribution
From (56),

$$\frac{\partial^2 L}{\partial\mu\partial(\sigma^2)} = -\frac{N\mu - (x_1 + \ldots + x_N)}{(\sigma^2)^2}$$

where $\bar{x} = \frac{x_1+x_2+\ldots+x_N}{N}$, and so

$$E\left(\frac{\partial^2 L}{\partial\mu\partial(\sigma^2)}\right) = E\left(\frac{\partial^2 L}{\partial(\sigma^2)\partial\mu}\right) = -\frac{N\mu - (N\mu)}{(\sigma^2)^2} = 0 \tag{59}$$

This tells us that the off-diagonal elements in the Fisher information matrix will be zero, thus simplifying later calculations.
Returning to look at the diagonal elements, from (56) and (57),

$$\frac{\partial^2 L}{\partial\mu^2} = \frac{N}{\sigma^2} \tag{60}$$

and

$$\frac{\partial^2 L}{\partial(\sigma^2)^2} = -\frac{N}{2(\sigma^2)^2} + \frac{1}{(\sigma^2)^3}\sum_{j=1}^{N}(x_j - \mu)^2$$

So,

$$E\left\{\frac{\partial^2 L}{\partial(\sigma^2)^2}\right\} = -\frac{N}{2(\sigma^2)^2} + \frac{1}{(\sigma^2)^3}.N\sigma^2 = \frac{N}{2(\sigma^2)^2} \tag{61}$$

Hence, since (59) gives that the Fisher information matrix is diagonal, from (60) and (61),

$$\begin{aligned} F &= \left\{ E\left(\frac{\partial^2 L}{\partial \mu^2}\right)\right\} \ . \ E\left\{\frac{\partial^2 L}{\partial (\sigma^2)^2}\right\} \\ &= \left(\frac{N}{\sigma^2}\right) \ \frac{N}{(2\sigma^2)^2} \\ &= \frac{N^2}{2(\sigma^2)^3} \end{aligned} \tag{62}$$

Minimising the message length for the Gaussian distribution

To within a constant, from (55) and (62),

$$\begin{aligned} MessLen &= -\log h(\vec{\theta}) + L + \frac{1}{2}\log F \ + \ \text{constant} \\ &= -\log h(\vec{\theta}) + \frac{N}{2}\log(2\pi) \ + \ \frac{1}{2}\log(\frac{N^2}{2}) \\ &\quad + \frac{N}{2}\log(\sigma^2) - \frac{3}{2}\log(\sigma^2) \\ &\quad + \frac{1}{2\sigma^2}\sum_{j=1}^{N}(x_j - \mu)^2 \\ &= \frac{N}{2}\log(2\pi) + \frac{1}{2}\log(\frac{N^2}{2}) - \log h(\vec{\theta}) + \frac{N-3}{2}\log(\sigma^2) \\ &\quad + \frac{1}{2\sigma^2}\sum_{j=1}^{N}(x_j - \mu)^2 \end{aligned} \tag{63}$$

$$\frac{\partial MessLen}{\partial \mu} \ = \ -\frac{\partial(\log h)}{\partial \mu} \ + \ \frac{N}{\sigma^2}(\mu - \bar{x})$$

where

$$\bar{x} \ = \ \frac{x_1 + x_2 + \ldots + x_N}{N}$$

and

$$\frac{\partial MessLen}{\partial(\sigma^2)} \ = \ -\frac{\partial(\log h)}{\partial \sigma^2} \ + \ \frac{N-3}{2\sigma^2} - \frac{1}{2(\sigma^2)^2}\sum_{j=1}^{N}(x_j - \mu)^2 \tag{64}$$

With our prior $h(\vec{\theta}) \ \propto \ \frac{1}{\sigma}$, we have that $\frac{\partial h}{\partial \mu} \ = \ 0$ and not surprisingly, that

$$(\mu)_{MML} \ = \ (\mu)_{ML} \ = \ \bar{x} \tag{65}$$

and

$$2(\sigma^2)^2\frac{\partial MessLen}{\partial(\sigma^2)} \ = \ -2(\sigma^2)^2 \ . \ \frac{\partial(\log h)}{\partial(\sigma^2)} \ + \ (N-3)\sigma^2 - \sum_{j=1}^{N}(x_j - \mu)^2 \tag{66}$$

At $(\sigma^2)_{MML}$, the above expression in (66) equals 0.

Substituting $(\mu_i)_{MML} = \bar{x}_i$ from (65) and using the improper prior $h(\sigma^2) = \frac{1}{\sigma^2}$ (with $\log h(\sigma^2) = -\log(\sigma^2)$) gives

$$+2(\sigma^2)^2\frac{1}{\sigma^2} + (N-3)\sigma^2 - \sum_{j=1}^{N}(x_j - \bar{x})^2 = 0$$

and so

$$(\sigma^2)_{MML} = \frac{\sum_{j=1}^{N}(x_j - \bar{x})^2}{N-1} = \frac{s^2}{N-1}. \tag{67}$$

We note from (65), (58) and (67) that $(\mu)_{MML} = (\mu)_{ML} = \bar{x}$, $(\sigma^2)_{ML} = \frac{s^2}{N}$ and $(\sigma^2)_{MML} = \frac{s^2}{N-1}$.
Statisticians often talk of $\frac{s^2}{N-1}$ as being the *unbiased estimator*. We see here a small sample bias in the Maximum Likelihood estimator that has people over-rule its value to return the unbiased estimator, the value which popped out from the MML estimator with the conjugate prior $h_{\mu,\sigma}(\mu,\sigma) \propto \frac{1}{\sigma}$.

This small sample bias of the Maximum Likelihood estimator of the standard deviation, $\hat{\sigma}_{ML}$, will be seen to form the basis for the Neyman-Scott problem (Neyman and Scott, 1948).

The Gaussian distribution is a fairly simple statistical distribution as far as they go. We will see an even worse small sample bias of the Maximum Likelihood estimator for the von Mises circular distribution in Section 8.7.

8.3.3 Choice of Gaussian parameterisation: μ and σ^2

In sections 8.3.1 and 8.3.2, we obtained the Maximum Likelihood and MML estimates with respect to μ and σ^2.
Since both the Maximum Likelihood and MML estimates are invariant under 1-to-1 twice continuously differentiable re-parameterisations, if we re-parameterise our co-ordinate space to μ and σ, we should find that the likelihood function and its maximum remain invariant, and we should likewise find that the message length function and its minimum remain invariant.

Exercise(s):

Re-derive the Maximum Likelihood estimator from section 8.3.1 for the parametisation (μ,σ). sections 8.3.2

Re-derive the MML estimator from section 8.3.2 for the parametisation (μ,σ).

8.3.4 Kullback-Leibler distance between two Gaussian distributions

$$
\begin{aligned}
& \int_{-\infty}^{\infty} dx\ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \times [-\log\sigma - \frac{1}{2\sigma^2}(x-\mu)^2 + \log\hat{\sigma} + \frac{1}{2\hat{\sigma}^2}(x-\hat{\mu})^2] \\
= \quad & \log(\frac{\hat{\sigma}}{\sigma}) + \int_{-\infty}^{\infty} dx\ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \times [-\frac{1}{2\sigma^2}(x-\mu)^2 + \frac{1}{2\hat{\sigma}^2}((x-\mu)+(\mu-\hat{\mu}))^2] \\
= \quad & \log(\frac{\hat{\sigma}}{\sigma}) + \int_{-\infty}^{\infty} dx\ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \\
& \qquad \times [-\frac{1}{2\sigma^2}(x-\mu)^2 + \frac{1}{2\hat{\sigma}^2}\{(x-\mu)^2 + (x-\mu)(\mu-\hat{\mu}) + (\mu-\hat{\mu})^2\}] \\
= \quad & \log(\frac{\hat{\sigma}}{\sigma}) + \frac{1}{2\hat{\sigma}^2}(\mu-\hat{\mu})^2 \quad + \quad (\mu-\hat{\mu}) \int_{-\infty}^{\infty} dx\ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \times (x-\mu) \\
+ \quad & (\frac{1}{2\hat{\sigma}^2} - \frac{1}{2\sigma^2}) \int_{-\infty}^{\infty} dx\ \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \times (x-\mu)^2 \\
= \quad & \log(\frac{\hat{\sigma}}{\sigma}) + \frac{1}{2\hat{\sigma}^2}(\mu-\hat{\mu})^2 + 0 + \frac{1}{2}(\frac{\sigma^2}{\hat{\sigma}^2} - 1) \\
= \quad & \log(\frac{\hat{\sigma}}{\sigma}) + \frac{1}{2\hat{\sigma}^2}(\mu-\hat{\mu})^2 + \frac{1}{2}(\frac{\sigma^2}{\hat{\sigma}^2} - 1)
\end{aligned}
\tag{68}
$$

Let us now differentiate with respect to $\hat{\mu}$ and $\hat{\sigma}$ to find the minimising values of $\hat{\mu}$ and $\hat{\sigma}$.

$$
\frac{\partial}{\partial\hat{\sigma}} \ = \ + \frac{1}{\hat{\sigma}} - \frac{1}{\hat{\sigma}^3}(\mu-\hat{\mu})^2 - \frac{\sigma^2}{\hat{\sigma}^3} \tag{69}
$$

This equals 0 when $\hat{\sigma}^2 \ = \ \sigma^2 + (\mu-\hat{\mu})^2$.

$$
\frac{\partial}{\partial\hat{\mu}} \ = \ - \frac{1}{\hat{\sigma}^2}(\mu-\hat{\mu}) \tag{70}
$$

This equals 0 when $\hat{\mu} \ = \ \mu$.
The joint optimum occurs when $\hat{\mu} \ = \ \mu$ and so $\hat{\sigma} \ = \ \sigma$.

8.4 Poisson distribution

The Poisson distribution is often used to model counts, such as the number of radioactive decays in a certain time period given a certain half-life or the number of traffic accidents occurring along a certain stretch of road in a certain time period. This has also been used to try to identify the authors of 17th century texts, where we might try to model how frequently certain authors use certain words in certain of their works.

Let r be the rate at which the event (radioactive decay, word usage, etc.) occurs, let t_i be the ith time period (or length of document) and let c_i be the ith number of occurrences.

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

$$L \;=\; -\log\left\{\prod_{i=1}^{N}\frac{e^{-rt_i}(rt_i)^{c_i}}{c_i!}\right\} \;=\; \sum_{i=1}^{N} rt_i - c_i\log(r) - c_i\log(t_i) + \log(c_i!)$$

Differentiating, we have

$$\frac{\partial L}{\partial r} \;=\; \sum_{i=1}^{N}(t_i - \frac{c_i}{r}) \tag{71}$$

and

$$\frac{\partial^2 L}{\partial r^2} \;=\; \sum_{i=1}^{N}(\frac{c_i}{r^2}) \;=\; \frac{1}{r^2}\sum_{i=1}^{N} c_i$$

So,

$$F \;=\; E(\frac{1}{r^2}\sum_{i=1}^{N} c_i) \;=\; \frac{1}{r^2}\sum_{i=1}^{N} E(c_i) \;=\; \frac{1}{r^2}\sum_{i=1}^{N} rt_i \;=\; \frac{1}{r}\sum_{i=1}^{N} t_i$$

From some data, $\{(c_i, t_i), i = 1, \ldots, N\}$, we wish to infer an estimate $\hat{r}$ of r.

Clearly, from (71), letting $C \;=\; \sum_{i=1}^{N} c_i$ and $T \;=\; \sum_{i=1}^{N} t_i$, $\quad \hat{r}_{ML} \;=\; \frac{\sum_{i=1}^{N} c_i}{\sum_{i=1}^{N} t_i} \;=\; \frac{C}{T}$.
In order to infer the MML estimate, we first need a prior on r.
For some α, let this prior be $h(r) = \frac{1}{\alpha}e^{-\frac{r}{\alpha}}$.
Then, to within a constant,

$$\begin{aligned} MessLen \quad &= \quad -\log h + L + \frac{1}{2}\log F + \text{constant} \\ &= \quad \log\alpha + \frac{r}{\alpha} + r\sum_{i=1}^{N} t_i - \log r\sum_{i=1}^{N} c_i - \sum_{i=1}^{N} c_i\log(t_i) + \sum_{i=1}^{N}\log(c_i!) - \frac{1}{2}\log r + \frac{1}{2}\log\sum_{i=1}^{N} t_i \end{aligned}$$

$\hat{r}_{MML}$ occurs when $\frac{\partial MessLen}{\partial r} = 0$, i.e., when

$$\frac{1}{\alpha} + \sum_{i=1}^{N} t_i - \frac{1}{r}\sum_{i=1}^{N} c_i - \frac{1}{2r} \;=\; 0$$

Solving this (Wallace and Dowe, 1994, 1997) gives

$$\frac{1}{r}(\sum_{i=1}^{N} c_i + \frac{1}{2}) \;=\; \frac{1}{\alpha} + \sum_{i=1}^{N} t_i$$

and so

$$\hat{r}_{MML} \;=\; (C + 1/2)/(T + 1/\alpha).$$

8.4.1 Kullback-Leibler distance between two Poisson distributions

$$\begin{aligned}
\sum_{i=0}^{\infty} e^{-\lambda_1} \frac{{\lambda_1}^i}{i!} \log \frac{e^{-\lambda_1} \frac{{\lambda_1}^i}{i!}}{e^{-\lambda_2} \frac{{\lambda_2}^i}{i!}} &= e^{-\lambda_1} \sum_{i=0}^{\infty} \frac{{\lambda_1}^i}{i!} \{-\lambda_1 + \lambda_2 + i \log \frac{\lambda_1}{\lambda_2}\} \\
&= -\lambda_1 + \lambda_2 + e^{-\lambda_1} \sum_{i=0}^{\infty} \frac{i{\lambda_1}^i}{i!} \log \frac{\lambda_1}{\lambda_2} \\
&= -\lambda_1 + \lambda_2 + (e^{-\lambda_1} \log \frac{\lambda_1}{\lambda_2}) \lambda_1 \sum_{i=1}^{\infty} \frac{{\lambda_1}^{i-1}}{(i-1)!} \\
&= -\lambda_1 + \lambda_2 + (e^{-\lambda_1} \log \frac{\lambda_1}{\lambda_2}) \lambda_1 e^{\lambda_1} \\
&= -\lambda_1 + \lambda_2 + \lambda_1 (\log \lambda_1 - \log \lambda_2)
\end{aligned} \tag{72}$$

In the special limiting case of $\lambda_1 = 0$, we can obtain either directly from the Taylor series expansion with $\lambda_1 = 0$ or from l'Hôpital's rule in the limit as $\lambda_1 \to 0$ that for $\lambda_1 = 0$, the Kullback-Leibler distance is $\log \frac{1}{e^{-\lambda_2} . 1/1} = \lambda_2$.

$$\frac{\partial}{\partial \lambda_2} = 1 - \frac{\lambda_1}{\lambda_2} \tag{73}$$

The optimum occurs for $\lambda_2 = \lambda_1$.

8.5 Geometric distribution

Consider a coin which is possibly not fair, and then consider the probability distribution of the number of consecutive heads from such a coin.

If the probability of throwing a Head with this possibly biased coin is p, then the probability of getting x consecutive heads (given that $x \geq 1$ is):
$f(x|p) = p^{x-1}(1-p)$.

Now, imagine we sample N different runs of Heads, which have lengths x_i for $i = 1, 2, \ldots, N$. Then, with data $\vec{x} = \{x_1, x_2, \ldots, x_N\}$,

$$f(\vec{x}|p) = \prod_{i=1}^{N} p^{x_i - 1}(1-p) = (1-p)^N \, p^{\sum_{i=1}^{N}(x_i - 1)}$$

Exercise(s):

Calculate the log-likelihood, $L = -\log f(\vec{x}|p)$.
Let $X = \sum_{i=1} x_i$.

Work out the Fisher information, $F(p)$.

$$\begin{aligned} E(x_i) &= (1-p)(1+2p+3p^2+\ldots) \\ &= (1-p)(\frac{d1}{dp}+\frac{dp}{dp}+\frac{dp^2}{dp}+\frac{dp^3}{dp}+\ldots) \\ &= (1-p)(\frac{d}{dp}(1+p+p^2+p^3+\ldots)) \\ &= (1-p)\frac{1}{(1-p)^2} \;=\; \frac{1}{(1-p)} \end{aligned}$$

$$F(p) = N/((1-p)^2 p)$$

Calculate the Maximum Likelihood estimate of p, $\hat{p}_{ML}$.

Assuming a uniform prior on p, $h(p) = 1$, calculate
the posterior mean of p,
the MEKLD estimate of p and
the MML estimate of p, $\hat{p}_{MML}$.

$$\hat{p}_{MML} = (X - N + 1/2)/(X + 3/2)$$

Also, calculate the message length at its minimum, when $p = \hat{p}_{MML}$.

8.5.1 Kullback-Leibler distance between two Geometric distributions

The Geometric distribution, $G(p)$, is a special case of the Negative Binomial distribution, corresponding to the case when $r = 1$. $G(p) \;\sim\; Nb(1,p)$.
In this case, the Kullback-Leibler distance is given by

$$\begin{aligned} &= \frac{1}{1-p}(p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}) \\ &= \frac{1}{1-p} \times d_{KL}(Bin(p,1-p), Bin(q,1-q)) \end{aligned} \tag{74}$$

8.6 Logistic distribution

$$f(y=0|x,\beta,c) \;=\; \frac{1}{1 \;+\; e^{-\beta(x-c)}}$$

$$\begin{aligned} f(y=1|x,\beta,c) &= 1 - f(y=0|x,\beta,c) \;=\; 1 - \frac{1}{1 \;+\; e^{-\beta(x-c)}} \;=\; \frac{(1+e^{-\beta(x-c)})-1}{1 \;+\; e^{-\beta(x-c)}} \\ &= \frac{e^{-\beta(x-c)}}{1 \;+\; e^{-\beta(x-c)}} \;=\; \frac{1}{1 \;+\; e^{+\beta(x-c)}} \end{aligned}$$

We wish to estimate the constant, c, and the "gradient", β.

The logistic distribution is relatively simple and ranges from 0 to 1, so it is good for modelling behaviour.

It is useful in neural networks (as a sigmoid activation function) and for modelling computational or economic "agents".
Let

$$p = \frac{1}{1 + e^{-\beta(x_i - c)}}$$

and so

$$1 - p = \frac{1}{1 + e^{\beta(x_i - c)}}$$

Then $(1-p)^2 - p^2 = ((1-p) - p)((1-p) + p) = 1 - 2p = (1-p) - p$.
Note: Chris Wallace's comment (Mon 10/5/99 - Tue 11/5/99) that this stuff is better done in terms of the state probabilities : p and $1-p$ if binary, and $p_1, \ldots, p_k$ if k-state. This refers to all the calculations in this section, 8.6.

$$\begin{aligned}
L &= -\log \prod_{i=1}^{N} f(y_i|x_i, \beta, c) \\
&= -\log[\prod_{i:y_i=0} f(y_i|x_i, \beta, c) \times \prod_{i:y_i=1} f(y_i|x_i, \beta, c)] \\
&= -\sum_{i:y_i=0} \log[\frac{1}{1 + e^{-\beta(x_i - c)}}] - \sum_{i:y_i=1} \log[\frac{1}{1 + e^{+\beta(x_i - c)}}] \\
&= \sum_{i:y_i=0} \log[1 + e^{-\beta(x_i - c)}] + \sum_{i:y_i=1} \log[1 + e^{+\beta(x_i - c)}]
\end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial \beta} &= -\sum_{i:y_i=0} \frac{(x_i - c)e^{-\beta(x_i - c)}}{1 + e^{-\beta(x_i - c)}} + \sum_{i:y_i=1} \frac{(x_i - c)e^{\beta(x_i - c)}}{1 + e^{\beta(x_i - c)}} \\
&= -\sum_{i:y_i=0} \frac{(x_i - c)}{1 + e^{\beta(x_i - c)}} + \sum_{i:y_i=1} \frac{(x_i - c)}{1 + e^{-\beta(x_i - c)}}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 L}{\partial \beta^2} &= \sum_{i:y_i=0} \frac{(x_i - c)^2 e^{\beta(x_i - c)}}{(1 + e^{\beta(x_i - c)})^2} + \sum_{i:y_i=0} \frac{(x_i - c)^2 e^{-\beta(x_i - c)}}{(1 + e^{-\beta(x_i - c)})^2} \\
&= \sum_{i:y_i=0} \frac{(x_i - c)^2}{4\cosh^2(\frac{\beta(x_i - c)}{2})} + \sum_{i:y_i=1} \frac{(x_i - c)^2}{4\cosh^2(\frac{\beta(x_i - c)}{2})}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{4}\sum_{i=1}^{N}\frac{(x_i-c)^2}{\cosh^2(\frac{\beta(x_i-c)}{2})} \\
&= \frac{1}{4}\sum_{i=1}^{N}(x_i-c)^2\times \operatorname{sech}^2(\frac{\beta(x_i-c)}{2})
\end{aligned}
$$

$$
E(\frac{\partial^2 L}{\partial\beta^2}) = \frac{\partial^2 L}{\partial\beta^2} = \frac{1}{4}\sum_{i=1}^{N}\frac{(x_i-c)^2}{\cosh^2(\frac{\beta(x_i-c)}{2})} = \frac{1}{4}\sum_{i=1}^{N}(x_i-c)^2\times \operatorname{sech}^2(\frac{\beta(x_i-c)}{2})
$$

$$
\begin{aligned}
\frac{\partial L}{\partial c} &= \sum_{i:y_i=0}\frac{\beta e^{-\beta(x_i-c)}}{1+e^{-\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{\beta e^{-\beta(x_i-c)}}{1+e^{-\beta(x_i-c)}} \\
&= \beta[\sum_{i:y_i=0}\frac{e^{-\beta(x_i-c)}}{1+e^{-\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{e^{\beta(x_i-c)}}{1+e^{\beta(x_i-c)}}] \\
&= \beta[\sum_{i:y_i=0}\frac{1}{1+e^{\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{1}{1+e^{-\beta(x_i-c)}}]
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 L}{\partial c^2} &= \beta[\sum_{i:y_i=0}\frac{\beta e^{\beta(x_i-c)}}{(1+e^{\beta(x_i-c))^2}} + \sum_{i:y_i=1}\frac{\beta e^{-\beta(x_i-c)}}{(1+e^{-\beta(x_i-c))^2}}] \\
&= \sum_{i:y_i=0}\frac{\beta^2}{4\cosh^2(\frac{\beta(x_i-c)}{2})} + \sum_{i:y_i=1}\frac{\beta^2}{4\cosh^2(\frac{\beta(x_i-c)}{2})}
\end{aligned}
$$

$$
E(\frac{\partial^2 L}{\partial c^2}) = \frac{\partial^2 L}{\partial c^2} = \frac{1}{4}\sum_{i=1}^{N}\beta^2\times \operatorname{sech}^2(\frac{\beta(x_i-c)}{2})
$$

$$
\frac{\partial^2 L}{\partial c\partial\beta} = \frac{\partial^2 L}{\partial\beta\partial c} = \frac{\partial}{\partial\beta}(\frac{\partial L}{\partial c}) = \frac{\partial}{\partial\beta}(\beta[\sum_{i:y_i=0}\frac{1}{1+e^{\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{1}{1+e^{-\beta(x_i-c)}}])
$$

$$
\begin{aligned}
&= [\sum_{i:y_i=0}\frac{1}{1+e^{\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{1}{1+e^{-\beta(x_i-c)}}] \\
&+ \beta[-\sum_{i:y_i=0}\frac{(x_i-c)e^{\beta(x_i-c)}}{(1+e^{\beta(x_i-c)})^2} - \sum_{i:y_i=1}\frac{(x_i-c)e^{-\beta(x_i-c)}}{(1+e^{-\beta(x_i-c)})^2}] \\
&= [\sum_{i:y_i=0}\frac{1}{1+e^{\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{1}{1+e^{-\beta(x_i-c)}}] \\
&- \beta[\sum_{i:y_i=0}\frac{(x_i-c)\ \operatorname{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4} + \sum_{i:y_i=1}\frac{(x_i-c)\ \operatorname{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4}] \\
&= [\sum_{i:y_i=0}\frac{1}{1+e^{\beta(x_i-c)}} - \sum_{i:y_i=1}\frac{1}{1+e^{-\beta(x_i-c)}}] - \beta\sum_{i=1}^{N}\frac{(x_i-c)\ \operatorname{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4}
\end{aligned}
$$

$$
\begin{aligned}
& E(\frac{\partial^2 L}{\partial\beta\partial c}) \\
= \quad & E([\sum_{i:y_i=0} \frac{1}{1 \ + \ e^{\beta(x_i-c)}} \ - \sum_{i:y_i=1} \frac{1}{1 \ + \ e^{-\beta(x_i-c)}}]) \ - E(\beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4}) \\
= \quad & E([\sum_{i:y_i=0} \frac{1}{1 \ + \ e^{\beta(x_i-c)}} \ - \sum_{i:y_i=1} \frac{1}{1 \ + \ e^{-\beta(x_i-c)}}]) \ - E(\beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4}) \\
= \quad & E(\sum_{i:y_i=0} \frac{1}{1 \ + \ e^{\beta(x_i-c)}}) \ - E(\sum_{i:y_i=1} \frac{1}{1 \ + \ e^{-\beta(x_i-c)}}) \ - \beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4} \\
= \quad & (\sum_{i:y_i=0} \frac{1}{1 \ + \ e^{\beta(x_i-c)}} \times f(y_i=0|x_i,\beta,c)) \ - (\sum_{i:y_i=1} \frac{1}{1 \ + \ e^{-\beta(x_i-c)}} \times f(y_i=1|x_i,\beta,c)) \\
- \quad & \beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4} \\
= \quad & (\sum_{i:y_i=0} \frac{1}{1 \ + \ e^{\beta(x_i-c)}} \times \frac{1}{1 \ + \ e^{-\beta(x_i-c)}}) \ - (\sum_{i:y_i=1} \frac{1}{1 \ + \ e^{-\beta(x_i-c)}} \times \frac{1}{1 \ + \ e^{\beta(x_i-c)}}) \\
- \quad & \beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4} \\
= \quad & -\beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4}
\end{aligned}
$$

$$
\begin{aligned}
F(\beta,c) \quad = \quad & E(\frac{\partial^2 L}{\partial\beta^2}) \ \times \ E(\frac{\partial^2 L}{\partial c^2}) \ - \ E(\frac{\partial^2 L}{\partial\beta\partial c})^2 \ = \ \frac{\partial^2 L}{\partial\beta^2} \ \times \ \frac{\partial^2 L}{\partial c^2} \ - \ E(\frac{\partial^2 L}{\partial\beta\partial c})^2 \\
= \quad & (\sum_{i=1}^{N} (\frac{1}{2}(x_i-c) \times \mathrm{sech}(\frac{\beta(x_i-c)}{2}))^2) \times (\sum_{i=1}^{N} (\frac{1}{2}\beta \times \mathrm{sech}(\frac{\beta(x_i-c)}{2}))^2) \\
- \quad & (\beta \sum_{i=1}^{N} \frac{(x_i-c)\ \mathrm{sech}^2(\beta(\mathrm{x_i}-\mathrm{c})/2)}{4})^2
\end{aligned}
$$

By the Cauchy-Schwar(t)z inequality, $F(\beta,c) \geq 0$. Again, by the Cauchy-Schwar(t)z inequality, $F(\beta,c) = 0$ only in the cases that $\beta = 0$ or all the $x_i - c$ are identical and all the x_i are identical.

8.6.1 Priors

$$
\begin{aligned}
p \quad &= \quad \frac{1}{1+e^{-\beta}} \\
1 \ dp \quad &= \quad h_p(p) \ dp \ = \ h_\beta(\beta) \ d\beta
\end{aligned}
$$

$$
\begin{aligned}
h_\beta(\beta) &= \frac{\partial p}{\partial \beta} = \frac{e^{-\beta}}{(1+e^{-\beta})^2} = \frac{1}{4\cosh^2(\frac{\beta}{2})} = \frac{\operatorname{sech}^2(\frac{\beta}{2})}{4} \\
&= \frac{1}{(1+e^{-\beta})(1+e^{\beta})} = \frac{1}{p(1-p)} \\
h_{\beta,c}(\beta, c) &= h_c(c) \times h_\beta(\beta) \\
|\frac{\partial}{\partial \beta}(\operatorname{sech}(\beta))| &= |\frac{\partial}{\partial \beta}(\frac{1}{\cosh(\beta)})| \\
&= |\frac{\partial}{\partial \beta}(\frac{2}{e^{\beta}+e^{-\beta}})| = |-\frac{2(e^{\beta}-e^{-\beta})}{(e^{\beta}+e^{-\beta})^2}| = \frac{(e^{\beta}-e^{-\beta})/2}{((e^{\beta}+e^{-\beta})/2)^2} = \frac{\sinh\beta}{\cosh^2\beta} = \frac{\tanh\beta}{\cosh\beta} \\
&= \operatorname{sech}\beta \ \tanh\beta
\end{aligned}
$$

8.6.2 Derivatives of Fisher information

Recall that

$$
F(\beta, c) = E(\frac{\partial^2 L}{\partial \beta^2}) \times E(\frac{\partial^2 L}{\partial c^2}) - E(\frac{\partial^2 L}{\partial \beta \partial c})^2 = \frac{\partial^2 L}{\partial \beta^2} \times \frac{\partial^2 L}{\partial c^2} - E(\frac{\partial^2 L}{\partial \beta \partial c})^2
$$

$$
\begin{aligned}
\frac{\partial}{\partial \beta}(E(\frac{\partial L}{\partial \beta^2})) &= \frac{\partial}{\partial \beta}(\frac{1}{4}\sum_{i=1}^{N}(x_i - c)^2 \times \operatorname{sech}^2(\frac{\beta(x_i - c)}{2}) \\
&= \frac{1}{4}\sum_{i=1}^{N}(x_i - c)^3 \operatorname{sech}^2(\frac{\beta(\mathrm{x_i} - \mathrm{c})}{2}) \tanh(\frac{\beta(\mathrm{x_i} - \mathrm{c})}{2})
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial}{\partial \beta}(E(\frac{\partial L}{\partial c^2})) &= \frac{\partial}{\partial \beta}(\frac{1}{4}\sum_{i=1}^{N}\beta^2 \times \operatorname{sech}^2(\frac{\beta(x_i - c)}{2})) \\
&= \frac{1}{2}\sum_{i=1}^{N} 2\beta \times \operatorname{sech}^2(\frac{\beta(x_i - c)}{2}) \\
&+ \frac{1}{4}\sum_{i=1}^{N}\beta^2 \times -\frac{(x_i - c)}{2}\operatorname{sech}(\frac{\beta(\mathrm{x_i} - \mathrm{c})}{2}) \tanh(\frac{\beta(\mathrm{x_i} - \mathrm{c})}{2})
\end{aligned}
$$

$$
\frac{\partial}{\partial \beta}(E(\frac{\partial L}{\partial \beta \partial c}))
$$

$$
\frac{\partial}{\partial c}(E(\frac{\partial L}{\partial \beta^2}))
$$

$$\frac{\partial}{\partial c}(E(\frac{\partial L}{\partial c^2}))$$

$$\frac{\partial}{\partial c}(E(\frac{\partial L}{\partial \beta \partial c}))$$

8.6.3 Derivatives of priors

$$|\frac{\partial h_\beta(\beta)}{\partial \beta}| \;=\; \frac{1}{4}2\ \mathrm{sech}(\frac{\beta}{2}) \times \frac{1}{2}\mathrm{sech}(\frac{\beta}{2})\ \tanh(\frac{\beta}{2}) \;=\; \frac{1}{4}\mathrm{sech}^2(\frac{\beta}{2})\tanh(\frac{\beta}{2})$$

$$|\frac{\partial \log h_\beta(\beta)}{\partial \beta}| \;=\; \frac{\frac{\partial h_\beta(\beta)}{\partial \beta}}{h_\beta(\beta)} \;=\; \frac{\mathrm{sech}^2(\frac{\beta}{2})\tanh(\frac{\beta}{2})/4}{\mathrm{sech}^2(\frac{\beta}{2})/4} \;=\; \tanh(\frac{\beta}{2})$$

8.7 von Mises circular distribution

The von Mises circular distribution is a circular analogue of the Gaussian distribution. It is specified by two parameters. These are μ, the location parameter or *mean*, specifying the middle of the distribution, and κ, the *concentration parameter*, which specifies how tightly the distribution is concentrated. κ can also be thought of as the ratio of magnetic field strength and the temperature. A weak field and high temperature will cause a wobbly compass needle, and a strong field at a low temperature will cause a tight distribution.
The 2-dimensional von Mises density, $M_2(\mu,\kappa)$ or $VM(\mu,\kappa)$, is an analogue of the Gaussian density for angles in the plane. The density of the angular variate θ is given by $f(\theta) \;=\; 1/(2\pi I_0(\kappa)).e^{\kappa\cos(\theta-\mu)}$.
The 2-dimensional von Mises density, $M_2(\mu,\kappa)$ or $VM(\mu,\kappa)$, is an analogue of the Gaussian density for angles in the plane. The density of the angular variate θ is given by $f(\theta) \;=\; 1/(2\pi I_0(\kappa)).e^{\kappa\cos(\theta-\mu)}$, where $I_0(\kappa)$ is a normalisation constant.

Let $I_0(\kappa) \;=\; \frac{1}{2\pi}\int_0^{2\pi} e^{\kappa\cos(\theta)}\ d\theta \;=\; \sum_{r=0}^{\infty}\frac{(\frac{\kappa}{2})^{2r}}{(r!)^2}$ and for $p>0$,
let $I_p(\kappa) \;=\; I_0(\kappa) \times E(\cos(p\theta)) \;=\; I_0(\kappa) \times \frac{1}{2\pi}\int_0^{2\pi}\cos(p\theta)e^{\kappa\cos(\theta)}\ d\theta \;=\; \sum_{r=0}^{\infty}\frac{(\frac{\kappa}{2})^{2r+p}}{(p+r)!\ r!}$.

This functional form

$f(x|\mu,\kappa) \;=\; \frac{1}{2\pi I_0(\kappa)}\ e^{\kappa\cos(x-\mu)}$
is sometimes written $X \sim M_2(\mu,\kappa)$.

So, $I_1(\kappa) \;=\; I_0(\kappa) \times E\left(\cos(\theta)\right) \;=\; \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r+1}}{r!\;(r+1)!} \;=\; \frac{d\;I_0(\kappa)}{d\kappa}$, which we shall soon use.

The functional form of the likelihood is thus
$f(\theta|\mu,\kappa) \;=\; \frac{1}{2\pi I_0(\kappa)}\; e^{\kappa\cos(\theta-\mu)}$,
and this is sometimes written $\theta \sim M_2(\mu,\kappa)$.

It looks something like the empty figure immediately below (for you to draw in by hand):

Just as the Gaussian distribution is a (maximum entropy) distribution on a line, so, too, the von Mises circular distribution is a (maximum entropy) distribution on the circle.

8.7.1 Motivation of the von Mises circular distribution

Richard von Mises, a mathematician and philosopher, was interested (circa 1918) in the distribution of atomic weights modulo unity. Other angular data which this distribution can be used to model include data pertaining to (e.g.)
dihedral angles (ϕ and ψ) in proteins,
arrival times at hospitals around a 24-hour clock,
R. von Mises's original data of atomic weights modulo unity,
magnetic fields,
oceanography,
etc.
(See N.I. Fisher's 1993 book for more examples.)

8.7.2 Parameter estimation for the von Mises circular distribution

$$\text{Let } \mathrm{A}(\kappa) \;=\; \mathrm{E}\left(\cos(\theta-\mu)\right) \;=\; \frac{\mathrm{I}_1(\kappa)}{\mathrm{I}_0(\kappa)} \;=\; \frac{\frac{\partial \mathrm{I}_0(\kappa)}{\partial \kappa}}{\mathrm{I}_0(\kappa)} \;=\; \frac{\mathrm{d}}{\mathrm{d}\kappa}(\log \mathrm{I}_0(\kappa))$$

which we shall need to estimate κ.

For small κ, $A(\kappa) \;=\; \frac{\kappa}{2}(1-\frac{\kappa^2}{8}+\frac{\kappa^4}{48}+O(\kappa^6))$
and for large κ, $A(\kappa) \;=\; 1-\frac{1}{2\kappa}-\frac{1}{8\kappa^2}+O(\kappa^{-3})$.

The log-likelihood, L, is given by :

$$\begin{aligned} L \quad &= \quad -\log\left(\prod_{i=1}^{N}\frac{1}{2\pi I_0(\kappa)}\, e^{\kappa\cos(\theta_i-\mu)}\right) \;=\; -\sum_{i=1}^{N}\log\left(\frac{1}{2\pi I_0(\kappa)}\, e^{\kappa\cos(\theta_i-\mu)}\right) \\ &= \quad N\log(2\pi) \;+\; N\log\left(I_0(\kappa)\right) \;-\; \kappa\sum_{i=1}^{N}\cos(\theta_i-\mu) \end{aligned} \tag{75}$$

For notational convenience, let $x=\sum_{i=1}^{N}\cos\theta_i$ and $y=\sum_{i=1}^{N}\sin\theta_i$, and $R=\sqrt{x^2+y^2}$.

$$\begin{aligned} \frac{\partial L}{d\mu} \;&=\; -\kappa\frac{\partial\sum_{i=1}^{N}\cos(\theta_i-\mu)}{\partial\mu} = -\kappa\frac{\partial\left(\cos\mu\sum_{i=1}^{N}\cos\theta_i+\sin\mu\sum_{i=1}^{N}\sin\theta_i\right)}{\partial\mu} = -\kappa\frac{\partial\left(x\cos\mu+y\sin\mu\right)}{\partial\mu} \\ &=\; -\kappa\sum_{i=1}^{N}\sin(\theta_i-\mu) \qquad = -\kappa(-\sin\mu\sum_{i=1}^{N}\cos\theta_i+\cos\mu\sum_{i=1}^{N}\sin\theta_i) \qquad = -\kappa(-x\sin\mu+y\cos\mu) \end{aligned}$$

So,

$$\hat{\mu}_{ML} \;=\; \tan^{-1}\left(\frac{y}{x}\right) \;=\; \tan^{-1}\left(\frac{\sum_{i=1}^{N}\sin\theta_i}{\sum_{i=1}^{N}\cos\theta_i}\right) \tag{76}$$

It follows immediately from this and our definition of R that

$$\cos(\hat{\mu}_{ML})=\frac{x}{\sqrt{x^2+y^2}}=\frac{x}{R} \quad \text{and} \quad \sin(\hat{\mu}_{\mathrm{ML}})=\frac{\mathrm{y}}{\sqrt{\mathrm{x}^2+\mathrm{y}^2}}=\frac{\mathrm{y}}{\mathrm{R}} \tag{77}$$

Also from equation (75), and using the definition of $A(\kappa)$,

$$\frac{\partial L}{d\kappa} \;=\; N\frac{\partial\log I_0(\kappa)}{d\kappa}-\sum_{i=1}^{N}\cos(\theta_i-\mu) \;=\; NA(\kappa)-(x\cos\mu+y\sin\mu) \tag{78}$$

To determine $\hat{\kappa}_{ML}$, we use equation (77):

$$0 \;=\; \frac{\partial L}{d\kappa} \;=\; NA(\kappa)-(x\frac{x}{R}+y\frac{y}{R}) \;=\; NA(\kappa)-\frac{x^2+y^2}{R} \;=\; NA(\kappa)-R$$

So, letting $\bar{R} = \frac{R}{N}$,

$$\hat{\kappa}_{ML} \;=\; A^{-1}\left(\frac{R}{N}\right) \;=\; A^{-1}(\bar{R}) \tag{79}$$

This gives us the Maximum Likelihood (ML) estimates, $\hat{\mu}_{ML}$ and $\hat{\kappa}_{ML}$, of μ and κ respectively.

The MML estimator needs the Fisher information, F, and a (subjective) Bayesian prior, $h()$, as well as the log-likelihood function, L. Since the Fisher information involves expected second derivatives of the log-likelihood function, it is no detour for us to derive the first derivatives of the log-likelihood function. It is at most a slight detour for us to use these first derivatives to determine the Maximum Likelihood estimates, since it is always at least a good diagnostic check to compare the MML estimator to the Maximum Likelihood estimator.

8.7.3 The Fisher information for the von Mises circular distribution

From equation (78),

$$\frac{\partial^2 L}{\partial\mu\partial\kappa} \;=\; -\sum_{i=1}^{N}\sin(\theta-\mu)$$

Therefore,

$$\begin{aligned}
E\left(\frac{\partial^2 L}{\partial\kappa\partial\mu}\right) \;&=\; E\left(\frac{\partial^2 L}{\partial\mu\partial\kappa}\right) \;=\; E\left(-\sum_{i=1}^{N}\sin(\theta_i-\mu)\right) \;=\; -NE(\sin(\theta-\mu)) \\
&=\; -N\frac{1}{2\pi I_0(\kappa)}\int_0^{2\pi} e^{\kappa\cos(\theta-\mu)}\sin(\theta-\mu)\;d\theta \\
&=\; -\frac{N}{2\pi I_0(\kappa)}\int_0^{2\pi}\frac{\partial}{\partial\theta}\left(e^{\kappa\cos(\theta-\mu)}\right)\;d\theta \\
&=\; -\frac{N}{2\pi I_0(\kappa)}\left[e^{\kappa\cos(\theta-\mu)}\right]_0^{2\pi} \;=\; -\frac{N}{2\pi I_0(\kappa)}\times 0 \\
&=\; 0
\end{aligned} \tag{80}$$

This means that the Fisher information matrix will be diagonal. We recall in passing that this was also the case for the Gaussian distribution in Section 8.3. This is no coincidence. The Gaussian and von Mises distributions are both symmetrical distributions with a (central) location parameter, μ, and a second parameter (σ or κ) measuring dispersion. Such distributions will typically tend to have a diagonal Fisher information matrix.

Knowing that the Fisher information matrix, F, is diagonal, we now calculate its other terms.

From equation (75),

$$\frac{\partial^2 L}{\partial \mu^2} \;=\; \kappa \sum_{i=1}^{N} \cos(\theta_i - \mu)$$

and

$$E\left(\frac{\partial^2 L}{\partial \mu^2}\right) \;=\; \kappa N E\left(\cos(\theta - \mu)\right) \;=\; \kappa N A(\kappa) \tag{81}$$

and, from equation (78),

$$E\left(\frac{\partial^2 L}{\partial \kappa^2}\right) \;=\; \frac{\partial^2 L}{\partial \kappa^2} \;=\; N\,\frac{d\;A(\kappa)}{d\kappa} \;=\; N\left(1 - \frac{A(\kappa)}{\kappa} - (A(\kappa))^2\right)$$

So,

$$\begin{aligned} F \;&=\; E\left(\frac{\partial^2 L}{\partial \mu^2}\right) E\left(\frac{\partial^2 L}{\partial \kappa^2}\right) \;-\; \left(E\left(\frac{\partial^2 L}{\partial \mu \partial \kappa}\right)\right)^2 \;=\; \kappa N A(\kappa) \times N\left(1 - \frac{A(\kappa)}{\kappa} - (A(\kappa))^2\right) - 0 \\ &=\; N^2 \kappa A(\kappa)\left(1 - \frac{A(\kappa)}{\kappa} - (A(\kappa))^2\right) \end{aligned} \tag{82}$$

8.7.4 Choice of prior for the von Mises distribution

Unless we have reason to do otherwise, it makes sense to choose a uniform prior on μ, $h_\mu(\mu) \;=\; \frac{1}{2\pi}$ on the range $[0, 2\pi)$.
Since this is such a "natural", "obvious" prior, let us take this as given and focus (below) on an appropriate prior for κ.
For the prior on κ, we elect (somewhat arbitrarily) to choose
$h_\kappa(\kappa) \;=\; \frac{\kappa}{(1+\kappa^2)^{3/2}}$ on the range $[0, \infty)$. Call this $h_3(\kappa)$.
One might consider also the improper prior $h_1(\kappa) \propto \frac{1}{\kappa}$ or the prior $h_2(\kappa) = \frac{2}{\pi(1+\kappa^2)}$.

On a variety of simulation experiments (Wallace and Dowe, 1993) summarised in the following pages, each of these priors and particularly $h_3(\kappa) \;=\; \frac{\kappa}{(1+\kappa^2)^{3/2}}$ were shown to out-perform Maximum Likelihood (ML) and a variety of other classical estimators.

8.7.5 The message length for the von Mises distribution

The message length is what Minimum Message Length seeks to minimise.

Recall from Section 4.4, (minus a typo or two) that, in general,
the message length is

$$\text{MsgLen} \;=\; -\log\left(\frac{h(\text{parameters})}{{k_n}^{n/2}\sqrt{F(\text{parameters})}}\right) \;+\; (-\log f(\text{data}|\text{parameters})) + \frac{n}{2} \tag{83}$$

$$=\; -\log\left(\frac{h(\text{parameters})\;\text{f}(\text{data}|\text{parameters})}{\sqrt{F(\text{parameters})}}\right) \;+\; \frac{n}{2}\left(1 + \log k_n\right) \tag{84}$$

where n is the number of parameters to be estimated and k_n is a lattice constant[5] which depends on n.

Since we know that wish to estimate nothing more and nothing less than μ and κ, we have that $n = 2$.

From equations (75) and (82) and Section 8.7.4, we can now calculate the message length.

Since $h_\mu(\mu)$ is uniform, note from equation (82) and Section 8.7.4 that μ occurs neither in $h()$ nor in F. So, by equations (84) and (76),

$$\hat{\mu}_{MML} \;=\; \hat{\mu}_{ML} \;=\; \tan^{-1}\left(\frac{y}{x}\right) \;=\; \tan^{-1}\left(\frac{\sum_{i=1}^{N}\sin\theta_i}{\sum_{i=1}^{N}\cos\theta_i}\right) \tag{85}$$

The message length can then be minimised numerically for κ.

If we recall the ML and the MML estimators for the multinomial, Gaussian and Poisson distributions, we see that the ML and MML estimators for these distributions are very similar.

However, the von Mises distribution is slightly "harder" than these distributions, and the small sample bias problems of Maximum Likelihood really start to show.

8.7.6 Kullback-Leibler distance between two von Mises distributions

Observe that

$$\begin{aligned}
& \kappa_1\cos(\theta-\mu_1) - \kappa_2\cos(\theta-\mu_2) \;=\; \kappa_1\cos(\theta-\mu_1) - \kappa_2\cos((\theta-\mu_1)+\mu_1-\mu_2)) \\
= \;& \kappa_1\cos(\theta-\mu_1) - \kappa_2\cos(\mu_1-\mu_2)\cos(\theta-\mu_1) + \kappa_2\sin(\mu_1-\mu_2)\sin(\theta-\mu_1) \\
= \;& (\kappa_1 - \kappa_2\cos(\mu_1-\mu_2))\cos(\theta-\mu_1) + \kappa_2\sin(\mu_1-\mu_2)\sin(\theta-\mu_1)
\end{aligned} \tag{86}$$

So,

$$\begin{aligned}
& \int_0^{2\pi} d\theta\, \frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1\cos(\theta-\mu_1)} \times \left[\log \frac{\frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1\cos(\theta-\mu_1)}}{\frac{1}{2\pi I_0(\kappa_2)} e^{\kappa_2\cos(\theta-\mu_2)}}\right] \\
= \;& \int_0^{2\pi} d\theta\, \frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1\cos(\theta-\mu_1)} \times [\log I_0(\kappa_2) - \log I_0(\kappa_1) + \kappa_1\cos(\theta-\mu_1) - \kappa_2\cos(\theta-\mu_2)] \\
= \;& \log I_0(\kappa_2) - \log I_0(\kappa_1) \\
+ \;& \int_0^{2\pi} d\theta\, \frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1\cos(\theta-\mu_1)} \times (\kappa_1 - \kappa_2\cos(\mu_1-\mu_2))\cos(\theta-\mu_1)
\end{aligned}$$

[5] $k_1 = 1/12 \approx 0.083333$ and $k_2 = 5/(36\sqrt{3}) \approx 0.080188$. See (Conway and Sloane, 1988; pp59-61) if you would like to know more about lattice constants.

$$
\begin{aligned}
&+ \quad \int_0^{2\pi} d\theta \, \frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1 \cos(\theta-\mu_1)} \times \kappa_2 \sin(\mu_1-\mu_2)\sin(\theta-\mu_1) \\
&= \quad \log I_0(\kappa_2) - \log I_0(\kappa_1) \\
&+ \quad \kappa_2 \sin(\mu_1-\mu_2) \int_0^{2\pi} d\theta \, \frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1 \cos(\theta-\mu_1)} \times \sin(\theta-\mu_1) \\
&+ \quad (\kappa_1 - \kappa_2 \cos(\mu_1-\mu_2)) \int_0^{2\pi} d\theta \, \frac{1}{2\pi I_0(\kappa_1)} e^{\kappa_1 \cos(\theta-\mu_1)} \times \cos(\theta-\mu_1) \\
&= \quad \log I_0(\kappa_2) - \log I_0(\kappa_1) \quad + \quad \kappa_2 \sin(\mu_1-\mu_2) \times 0 \quad + \quad \kappa_2 \sin(\mu_1-\mu_2) \times A(\kappa_1) \\
&= \quad \log I_0(\kappa_2) - \log I_0(\kappa_1) \quad + \quad \kappa_2 \sin(\mu_1-\mu_2) \times A(\kappa_1)
\end{aligned}
\tag{87}
$$

($\kappa = \kappa_1$, $\hat{\kappa} = \kappa_2$, $\mu = \mu_1$, $\hat{\mu} = \mu_2$.)

$$
\frac{\partial}{\partial \hat{\kappa}} \; = \; A(\kappa_2) - \cos(\mu - \hat{\mu}) A(\kappa_1) \tag{88}
$$

$$
\frac{\partial}{\partial \hat{\mu}} \; = \; - \hat{\kappa} \sin(\mu_1 - \mu_2) A(\kappa_1) \tag{89}
$$

In the event that $\kappa_1 \neq 0$, this equals 0 when $\hat{\mu} = \mu$. The joint optimum occurs when $\hat{\mu} = \mu$ and so $A(\kappa_2) = A(\kappa_1)$ and $\kappa_2 = \kappa_1$.
We present below some results from simulation runs (Wallace and Dowe, 1993; pages 14 – 19).

8.7.7 Simulation results for the von Mises circular distribution

These results were obtained by pseudo-randomly generating data of various sample sizes from known distributions, and then estimating μ and κ.

Estimation of μ is straighforward, with all the methods considered giving $\hat{\mu} = \hat{\mu}_{ML}$.

The estimators considered for κ include Maximum Likelihood (ML), marginal Maximum Likelihood (R. A. Fisher, 1953; G. Schou, 1978), a method we call "NF" due to N.I. Fisher (1993), and MML with the three priors $h_1(\kappa)$, $h_2(\kappa)$ and $h_3(\kappa)$ from Section 8.7.4.

The results reported pertain to mean bias (mb), mean absolute error (mae), mean squared error (mse) and mean Kullback-Leibler distance (KL).

Values of N of 2, 5, 10, 25, 100 and 500 were chosen.

The number of simulated runs are reported (in brackets) at the top of each column.
Values of κ (true value) of 0.0, 0.25, 0.50. 1.0, 2.0, 5.0 and 10.0 were chosen.

The entries in the table are the mean value (and, in brackets, the standard deviation) of respectively the bias, absolute error, mean squared error and mean Kullback-Leibler distance.

From (Wallace and Dowe, 1993) :

From (Wallace and Dowe, 1993) :

From (Wallace and Dowe, 1993) :

From (Wallace and Dowe, 1993) :

From (Wallace and Dowe, 1993) :

From (Wallace and Dowe, 1993) :

8.8 Wrapped Normal circular distribution

Re-cap on the Normal distribution

Recall that the functional form of the Gaussian - or Normal - distribution is
$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((x-\mu)^2)}$,
and this is sometimes written $X \sim N(\mu,\sigma^2)$.

Wrapped Normal distribution

Swapping from x to θ, for a wrapped Normal distribution,
$f(\theta|\mu,\sigma) = \sum_{j=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((\theta+2j\pi-\mu)^2)} = \frac{1}{\sqrt{2\pi}\sigma} \sum_{j=-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}((\theta+2j\pi-\mu)^2)}$,
and this is sometimes written $\theta \sim WN(\mu,\sigma^2)$.

For several pieces of data $\theta_1, ..., \theta_i, ..., \theta_N$,

$$\begin{aligned} L &= -\log f(\theta|\mu,\sigma) = \sum_{i=1}^{N} -\log\Big(\sum_{j=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)}\Big) \\ &= N\log(\sqrt{2\pi}) + N\log\sigma - \sum_{i=1}^{N} \log\Big(\sum_{j=-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)}\Big). \end{aligned}$$

Notice that for both a von Mises distribution and a wrapped Normal distribution, adding to or subtracting from θ an amount of 2π or any integer multiple of 2π does not change the value of the likelihood function or any of its derivatives.

First and second derivatives and Fisher information for the Wrapped Normal

The only differences between the likelihood function for the Normal distribution and the likelihood function for the wrapped Normal distribution are the appearance of the summation $\sum_{j=-\infty}^{+\infty}$ and the replacement of every occurrence of θ in the Gaussian distribution by $\theta + 2j\pi$.
So, along somewhat similar lines to the Gaussian distribution, the wrapped Normal distribution gives us

$$\begin{aligned} \frac{\partial L}{\partial \mu} &= -\sum_{i=1}^{N} \frac{\partial \log(\sum_{j=-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)})}{\partial\mu} \\ &= -\sum_{i=1}^{N} \frac{\frac{\sum_{j=-\infty}^{+\infty} \partial e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)})}{\partial\mu}}{\sum_{j=-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)})} \\ &= blah \sum_{i=1}^{N} \frac{\sum_{j=-\infty}^{+\infty} ???}{e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)}} \frac{\frac{\partial}{\partial\mu}\left\{\sum_{j=1}^{N}(x_j-\mu)^2\right\}}{blah} \\ &= blah \frac{N\mu - (x_1+\ldots+x_N)}{\sigma^2} blah \end{aligned}$$

and

$$\frac{\partial L}{\partial(\sigma^2)} \;=\; blah\frac{N}{2\sigma^2} \;-\; \frac{1}{2(\sigma^2)^2}\sum_{j=1}^{N}(x_j-\mu)^2 blah \tag{90}$$

8.9 Choice of prior

We have discussed above various methods of point estimation for a variety of distributions – multinomial, Gaussian, Poisson, von Mises circular and geometric. If we wish to use a Bayesian method, we must necessarily use a prior.

MML and MEKLD are but two Bayesian methods of point estimation. These two happen to be invariant and consistent, but there are also other Bayesian methods for point estimation (e.g., posterior mean). In choosing to use a Bayesian method, it is best to spend some time at least early in one's career considering the philosophical and pragmatic issue of choice of priors. Classical (i.e., likelihood-based) statisticians do not opt to use Bayesian priors. At the time of writing, probably less than 20% of statisticians would be willing to describe themselves as Bayesian.

8.9.1 A classical objection to Bayesian priors

The debate between classical (non-Bayesian and "anti"-Bayesian) statisticians is an old and sometimes heated one. Classical statisticians would prefer an "objective" method which neither requires nor takes advantage of prior information, and they might also draw attention to the subjectivity of a prior and one's difficulty in formulating a prior.

8.9.2 The data swamps the prior, anyway

Some will point out that, whatever your data is, you could then choose a prior which would make your inference as close as one liked to some wished conclusion. While this is true, *prior*s are meant to come *before* the data, and the following is true. As long as your prior does not make something a priori impossible, as the amount of data increases, your data will start to "swamp" the prior.

8.9.3 A further Bayesian response to such a classical objection

We have argued earlier (I think) and we argue again now that it is very rare for someone to have *total* ignorance about a data-set. If our prior knowledge gives us additional insight into the underlying model other than what was available to us from the data alone, then it makes pragmatic good sense to quantify this as well as possible and then to

use it. Better to make a decent job of quantifying something useful than not to use it at all.

Furthermore, by the Conjecture in Section 7.3.3 and Section 7.3.4, it would appear to follow that if we want our point estimation technique to both invariant and consistent, then it will have to be Bayesian.

8.9.4 "Mathematically convenient" priors and genuine subjective priors

We have argued earlier that Bayesian inference is necessary if one wants invariant consistent estimators (although this is not (yet) a theorem).

We have also argued the case for the use of a subjective Bayesian prior. However, there are some who support Bayesianism but who prefer to avoid subjective Bayesian priors and instead use what we shall call "mathematically convenient" priors.
We discussed in Section 7.1 that the square root of the Fisher information, $F(\vec{\theta})^{1/2}$, has the same mathematical form as a prior. Indeed, some refer to it as the Jeffreys prior (H. Jeffreys, 1946). It is an example of what we mean by a "mathematically convenient" (or non-subjective) prior.

However, the Jeffreys "prior" is not a *prior* in the sense of coming *before* the data and representing some prior beliefs. Rather, it comes from the likelihood function, and hence from the data. It is not a genuine subjective prior, but rather a "mathematically convenient" prior, undoubtedly motivated by an understandable (and laudable?) wish to try to take subjectivity out of Bayesian inference.

The Fisher information turns out to be some sort of measure of the expected information gain in conducting an experiment. For example, for the binomial distribution, the Fisher information $\propto 1/p(1-p)$, which is largest for extreme values of p (near 0 and 1) and small near $p = 1 - p = 0.5$. When p is not very extreme, we don't expect to learn a great deal from conducting our next experiment (or sampling our next datum), whereas we expect to learn a lot for extreme p. For the Gaussian distribution, $F \propto 1/\sigma^2 \times 1/(\sigma^2)^2 \propto 1/(\sigma^2)^3$, and for the Poisson distribution, $F \propto 1/r$.

The point is made most clearly by considering the von Mises distribution.
From equation (82), $F(\mu) \approx 0$ for small κ, and $F(\kappa) \approx 0$ for large κ; and $F(\mu, \kappa) = F(\mu) \times F(\kappa) \approx 0$ for both small and large κ.

So, the Jeffreys prior says that our prior is to expect the field strength parameter, κ, to be most likely in the region where our needle is most suited (not too weak a κ, lest we notice nothing amidst the noise, and not too strong a κ, lest we notice nothing because the distribution is so tight that the needle doesn't move). So, our Jeffreys prior says that we a priori expect κ to be most likely where our particular chosen compass is most likely to notice a difference. However, if we choose a different measuring implement, either a more sensitive compass or a less sensitive compass, this will see us having a different Jeffreys prior. Since priors are meant to come *before* the data, this example (from Dowe,

Oliver and Wallace, 1996), which suggests that our prior depends upon our measuring instrument, does not auger well for the Jeffreys prior.

Perhaps a better criticism of the Jeffreys prior is (Wallace, private communication, 1997): Imagine a watch-tower in pitch black darkness somewhere on the interior of a circle, but not at the centre of the circle. The guard hears sounds coming from the perimeter, and wishes to estimate the location of their source. A uniform prior on the circle seems most appropriate. However, since the Jeffreys "prior" assigns higher probability to parts of the parameter space that can be measured more accurately, the Jeffreys prior will see estimates biased towards being closer to the watch-tower.

Exercise

If one uses the Jeffreys "prior" to do MML inference, what does one get as the MML estimator?

Is such an estimator subjective Bayesian?
(Hint: The answer to this question immediately above is "No".)
Is it invariant? Is it generally consistent?

8.10 MML, square root of Fisher information and Strict MML

Recall the expression for the Message Length in equation (84) and Section 4.4, (minus a typo or two):

$$\text{MsgLen} \quad = \quad -\log\left(\frac{\text{h(parameters) f(data|parameters)}}{\sqrt{\text{F(parameters)}}}\right) \quad + \quad \frac{\text{n}}{2}\,(1+\log \text{k}_\text{n})$$

Recall (Sections 4.3 and 4.4) that the expression for the MML estimator comes from a quadratic Taylor expansion whose assumptions include the assumption that the prior, $h(.)$, is fairly flat around the estimator and the assumption that the log-likelihood is twice (continuously) differentiable.

It may well happen that only one or even none of these is true.

So, the bad news is that we must make the caveat that MML (as defined by minimising the expression in equation (84)) is not always guaranteed to work in the cases that either of the assumptions fails.

The good news is that there is a modification called Strict MML (Wallace and Boulton, 1975; Wallace and Freeman, 1987) which does not need the above assumptions or make any approximations. Strict MML can be quite intractable in practice, but it is invariant and consistent. For many typical estimation problems, MML is a good, tractable, approximation to Strict MML.

9 Classification, Clustering, Mixture modelling

Mixture modelling is[6] variously known as mixture modelling, clustering, numerical taxonomy, unsupervised learning and intrinsic classification. It has so many names as it is of so much interest to so many different communities.

The problem of mixture modelling is one of partitioning data into a previously unknown number of clusters (or *components*) and then describing each cluster. We must decide upon the number of clusters and their relative abundances. For each cluster, we need to specify distributional parameters such as, for example, means and standard deviations. A somewhat moot point, which we shall return to discuss further, is whether we also need to specify for each data thing which class it is assigned to. In the case where we do specify this, the problem is often known as mixture modelling *with latent class assignment* or *fully-parameterised* mixture modelling.

Mixture modelling is a fundamentally important problem in Artificial Intelligence, where it is variously known as *clustering* or *unsupervised learning*. It is important and much studied in statistics, where it is variously known as mixture modelling or *clustering* and the clusters are generally called *components*. Philosophers talk of *intrinsic classification* and *natural kinds*. Medicine has "symptom clusters". Botanists talk of numerical taxonomy.

The problem of mixture modelling[7] is important in that, given a new body of data, such as a child discovering the world around it or a botanist visiting a new land or a new jungle, we wish to infer a theory of which things are (in some sense) similar so that we can more concisely represent this new world around us.

The MML approach to mixture modelling was first developed by Wallace and Boulton (1968) in their Snob program. Some good expositions are (Wallace, 1986), (Wallace, 1990) and (Wallace and Dowe, 1997).
See http://www.csse.monash.edu.au/~dld/Snob.html .

9.1 MML mixture modelling : constructing a two-part message

In order to do mixture modelling using MML, we want to construct a two-part message conveying the mixture model, where the first part of the message encodes the model (or hypothesis) and the second part of the message encodes the data given the model.

Given the nature of a mixture model, the first part of the message will need to specify the number of classes (components), the relative abundances of each class (using a multinomial distribution) and then, for each class in turn, the parameter estimates. This is most of

[6] ©David L. Dowe 1997-1999
[7] see http://www.csse.monash.edu.au/~dld/cluster.html .

the first part of the message. A moot point arises as to whether first part of the message should also include a statement of which data things[8] are assigned to which components. When we settle upon the form of the hypothesis to be conveyed in the first part of the message, the second part of the message will encode the data in light of this hypothesis.

Related to this apparently moot point of whether the model needs to state which component each thing is assigned to is the issue of whether such an assignment must be (deterministic or) total or whether it can instead be (probabilistic or) partial.

9.1.1 Stating the message – a first draft

Let us imagine our hypothesis, H, conveying:
1 a) No. of classes or components (say k).
1 b) The relative abundance of each component.
1 c) For each component, the distribution parameter estimates to describe the component.
1 d) For each data thing, the component that it is estimated to belong to.

Having sent such an hypothesis, H, we can now transmit the second part of the message – the data, D, given H, by specifying for each thing in turn which class it is most likely to belong to and then encoding the thing given the parameters for that class.

This was the coding mechanism used in the seminal (Wallace and Boulton, 1968) paper.

Let's think about what our priors are and how to send our message.

In 1 a), we need to specify the number of components.
This could (e.g.) be uniform from 1 to (say) 100, with each value assumed equally likely.
Suppose we have k components (or *classes*).
This would have message length $-\log(1/100) \;=\; \log(100)$.
(We should note here that the order in which the classes are transmitted is irrelevant to the model, so we look out for a $k!$ term.)

In 1 b), we specify the relative abundance of each component.
Call these relative abundances $p_1, p_2, \ldots, p_k$.
These must satisfy $p_1 + p_2 + \ldots + p_k = 1$ and for all i $p_i \geq 0$.
These are exactly the conditions of the multinomial distribution in Section 8.1.2 and, as such, the p_i could be stated appropriately.

Doing the exercises in Section 8.1.2
or citing (Wallace and Boulton, 1968; p187 (4), p194 (28)), we get
$\hat{p}_i = (n_i + 1/2)/(N + k/2)$, with a message length of
$(k-1)\log(N/12+1)/2 \;-\; \log((k-1)!) \;-\; \sum_{i=1}^{k}(n_i + 1/2)\log\hat{p}_i$.

[8] the word *thing* or the term *data thing* is chosen because, e.g., "item" is simply the Latin word for "thing".

In 1c), we do something that we have done repeatedly throughout Section **??** earlier.

Whatever the distribution is for a component, we cost the first part of that message, as given in equation (83).

In 1d), we then encode the chosen component (say component i) with code-word of length $-\log_2 \hat{p}_i$.

The above then encodes the hypothesis, H.

To encode the second part of the message, $D|H$, conveying each data thing given its component, we do as with the second part of the message in equation (83).

9.1.2 Stating the message more concisely using partial assignment

The original (Wallace and Boulton, 1968) coding scheme assigns things totally to classes. As such, it is a bit inefficient, for consider the possible savings when two classes overlap substantially.

Furthermore, if such an inefficiency is not corrected, then the coding scheme will result in inconsistent estimates (with the difference between class means over-estimated and the class standard deviations under-estimated).

The trick is to assign things partially to classes (Wallace, 1986).

Parts 1 (a), 1 (b) and 1 (c) of the message can remain unchanged, but we have to reconsider Part 1 (d) and Part 2 of the message.

In the first draft, in Part 1 (d), for each datum x and each component $j = 1, \ldots, k$, we considered $p(j, x) = p_j f(x|\text{class j})$ and assigned x to the class with $\max_j p(j, x)$.

To encode more concisely, let $P(x) = \sum_{j=1}^{k} p(j, x)$,

and probabilistically assign x to class j with probability $p(j, x)/P(x)$.

This gives a more efficient coding, and is consistent.

9.1.3 Some comments about parameter estimation in Snob

The original (Wallace and Boulton, 1968) Snob dealt with multinomial and Gaussian distributions, but did not use partial assignment. (Wallace, 1986) and (Wallace, 1990) used partial assignment.

This work was extended (Wallace and Dowe, 1994, 1997) to allow Poisson and von Mises circular distributions.

Although Maximum Likelihood gives similar values to MML for the multinomial, Gaussian and Poisson distributions, it does not do so for the von Mises distribution. But, Maximum Likelihood has another problem regarding mixture modelling. While it is easy to assign a likelihood to one component, how does one choose the number of components to use? A popular method for penalising Maximum Likelihood for having too many components, called the Akaike Information Criterion (AIC), is inconsistent[9] for mixture modelling.

A problem for many methods in dealing with problems like mixture modelling (such as estimation of polynomials, factor analysis and the Neyman-Scott problem) is that the number of parameters to be estimated can increase with the amount of data. Most methods have little or no trouble for a fixed number of parameters, but some methods will stumble or even become inconsistent when the number of parameters to be estimated can grow with the data.

9.1.4 Underlying assumptions in Snob

We are assuming that the various attributes are independent in each component (although work is being done on using MML to model correlation within mixture components). We also assume that the data is not serially correlated (although work is also being done on using MML to model correlation in sequential data).

[9] it is invariant but not Bayesian – see conjecture in Section 7.3.4.

Perhaps most subtly, we have also assumed that the parameters from one component do not interact with the parameters from any other components, i.e., that the relevant second cross-derivatives in the Fisher information matrix will be zero. This assumption is reasonable when components are well-separated, but there is some slight inefficiency here when components overlap.

9.1.5 Snob and Missing data

Given the above assumptions of independence between attributes, missing data does not pose a problem in Snob's message length framework. The missing data can be assumed to have a fixed, constant cost which will not impact on the minimisation of the message length.

9.1.6 Applications of Snob

Wallace and Boulton (1968) applied Snob to some data from the British museum on seal skulls.

Other studies include
sportsperson/surfer personality profile (J. Patrick?, 19??),
using the Poisson and Gaussian distributions to look at word counts to identify authorship style (unpublished),
clustering of data on grieving families (Kissane et al., 1996).

More recently, Dowe, Allison, Dix, Hunter, Wallace and Edgoose (1996) used the von Mises clustering in Snob to look at protein dihedral angles.

Proteins consist of a chain Nitrogen-(alpha-Carbon)-(beta-Carbon)- ... with an amino acid attached to the α-Carbon. Around the α-Carbon, proteins have two dihedral angles ϕ and ψ, which almost totally determine their 3-dimensional structure at that point.

Snob was the first program (and we are not yet aware at the time of writing of another such program outside Monash) to be able to deal with clustering of angular data.

Earlier attempts to cluster protein data in search of structure involved trying to use the Euclidean co-ordinates and then model these as coming from a Gaussian distribution. We were able to reduce $3 \times 3 = 9$ non-rotationally-invariant Euclidean co-ordinates into 2 invariant angles, ϕ and ψ, which we were then able to use the von Mises distribution in Snob to cluster.

Why?
Recall the uses of the von Mises distribution.
People might wish to analyse hospital arrival times around a 24-hour clock to look for clusters.
In proteins, people often talk of Extended, Helix and Other; or Extended, Helix, Turn

and Coil.
Given that others have worked in this area but had not been able to apply angular data, this was a natural problem for us to address, especially as we had the only angular clustering software and we knew from simulation runs that MML out-performed rival estimators when there was only one component and we know that MML has no difficulty in making a transition from one component to many components.

So, what did we find?
See (Dowe, Allison, Dix, Hunter, Wallace and Edgoose, 1996) and (Edgoose, Allison and Dowe, 1998).

From (Dowe, Allison, Dix, Hunter, Wallace and Edgoose, 1996) :

From (Dowe, Allison, Dix, Hunter, Wallace and Edgoose, 1996) :

From (Edgoose, Allison and Dowe, 1998) :

From (Edgoose, Allison and Dowe, 1998) :

From (Edgoose, Allison and Dowe, 1998) :

From (Edgoose, Allison and Dowe, 1998) :

9.2 Inconsistency from total assignment in mixture modelling

We recall from Section 9.1.2 that the method of Section 9.1.1 of using total assignment to estimate class membership in mixture modelling gave rise to a simple inconsistency.

It is worth examining this inconsistency a bit more closely. We are estimating p_1 and $1 - p_1$, the relative abundances of two overlapping 2-dimensional Gaussian distributions, and their respective means and standard deviations, μ_{11}, μ_{12}, μ_{21}, μ_{22} and σ_{11}, σ_{12}, σ_{21} and σ_{22}. The inconsistency will remain even if we assume that $p_1 = 1 - p_1 = 1/2$ and $\sigma_{11} = \sigma_{12} = \sigma_{21} = \sigma_{22} = \sigma$ and thus reduce the problem to one of estimating μ_{11}, μ_{12}, μ_{21}, μ_{22} and σ.

The cause of the inconsistency is the total assignment (in Section 9.1.1), which is due in turn to the rather questionable use of Maximum Likelihood to do the estimation of the q_i $(i = 1 \ldots N)$, the probability that thing i is in Class 1.
Let thing i be in class $C(i)$, and let N_1 and N_2 respectively be the number of things in class 1 and class 2, with $N_1 + N_2 = N$.

Consider the likelihood function

$$f(\vec{x}|\mu_{11}, \mu_{12}, \mu_{21}, \mu_{22}, \sigma, q_1, \ldots, q_i, \ldots, q_N) \tag{91}$$

as the likelihood function for data, $\vec{x}$, in terms of such a 2-component mixture specified by μ_{11}, μ_{12}, μ_{21}, μ_{22}, and σ and membership probabilities $q_1, \ldots, q_N$ for Class 1.

Let us estimate μ_{11}, μ_{12}, μ_{21}, μ_{22}, and σ by whatever means and then consider the estimation of $q_1, \ldots, q_N$.
Looking at the likelihood function above shows that the Maximum Likelihood estimator of each q_i will be either 0 or 1.

For the discrete binomial distribution, we recall from Section 8.1.1 that $\hat{p}_{ML} = \frac{x}{N}$, giving rise to small sample bias of Maximum Likelihood estimation, particularly so for $N = 1$.

It is this small sample bias of Maximum Likelihood for the binomial distribution that insists upon total assignment and in turn gives rise to the relevant inconsistency in estimating μ_{11}, μ_{12}, μ_{21}, μ_{22}, and σ.

Note that Maximum Likelihood has difficulty in estimating all the parameters, μ_{11}, μ_{12}, μ_{21}, μ_{22}, σ, $q_1, \ldots, q_N$ simultaneously. As the reader might have anticipated, MML is consistent for this exercise. Whereas Maximum Likelihood brings about its downfall by stating estimates with greater certainty than is warranted, MML obtains consistency by using the Fisher information to acknowledge an appropriate degree of uncertainty in the parameter estimates.

10 The Neyman-Scott problem – another inconsistency in ML

The example above shows an inconsistency in Maximum Likelihood estimation due to Maximum Likelihood's over-fitting in estimating the parameter(s) of a discrete distribution – namely the binomial distribution. Neyman and Scott gave an example (Neyman and Scott, 1948) of Maximum Likelihood being inconsistent in estimating a parameter from a continuous distribution, namely the variance, σ^2. As in the mixture modelling inconsistency above, Maximum Likelihood falls victim to stating estimates with greater certainty than is warranted. Once again, MML obtains consistency by using the Fisher information to acknowledge an appropriate degree of uncertainty in the parameter estimates.

The Neyman-Scott problem concerns M Gaussian distributions with unknown means $\mu_1, ..., \mu_i, ..., \mu_M$ respectively and identical but unknown standard deviation, σ. Two data, x_{i1} and x_{i2}, are sampled from each distribution, $N(\mu_i, \sigma^2)$. We then let M tend to infinity.

10.0.1 The likelihood function for the Neyman-Scott problem

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

$$\begin{aligned} L &= -\log\left\{\prod_{i=1}^{M}\prod_{j=1}^{2}\frac{1}{(2\pi)^{1/2}\sigma}e^{-\frac{\frac{1}{2}(x_{ij}-\mu_i)^2}{\sigma^2}}\right\} \\ &= M\log(2\pi) + M\log(\sigma^2) + \frac{1}{2\sigma^2}\sum_{i=1}^{M}\sum_{j=1}^{2}(x_{ij}-\mu_i)^2 \end{aligned}$$

Differentiating, we have

$$\frac{\partial L}{\partial \mu_k} = \frac{1}{2\sigma^2}\frac{\partial}{\partial \mu_k}\left\{\sum_{j=1}^{2}(x_{kj}-\mu_k)^2\right\} = \frac{2\mu_k - x_{k1} - x_{k2}}{2\sigma^2} \tag{92}$$

and

$$\frac{\partial L}{\partial(\sigma^2)} = \frac{M}{\sigma^2} - \frac{1}{2(\sigma^2)^2}\sum_{i=1}^{M}\sum_{j=1}^{2}(x_{ij}-\mu_i)^2 \tag{93}$$

As observed by Neyman and Scott (1948), this gives

$$(\mu_i)_{ML} = \frac{x_{i1}+x_{i2}}{2} = \bar{x}_i$$

and

$$(\sigma^2)_{ML} = \frac{1}{2M}\sum_{i=1}^{M}\sum_{j=1}^{2}\{x_{ij}-(\mu_i)_{ML}\}^2 \tag{94}$$

$$= \frac{\sum_{i=1}^{M}\left(\frac{x_{i1}-x_{i2}}{2^{1/2}}\right)^2}{2M}$$

$$\to \frac{\sigma^2}{2} \tag{95}$$

as $M \to \infty$. This shows the inconsistency in Maximum Likelihood.

We now outline the derivation (Dowe and Wallace, 1997) of the MML estimate, showing its (asymptotic unbiasedness and) consistency.

10.0.2 The Fisher Information for the Neyman-Scott problem

From (92),

$$E\left(\frac{\partial^2 L}{\partial\mu_k\partial\mu_l}\right) = \frac{\partial^2 L}{\partial\mu_k\partial\mu_l} = 0, \qquad k \neq l \tag{96}$$

and

$$\frac{\partial^2 L}{\partial\mu_k\partial(\sigma^2)} = -\frac{1}{(\sigma^2)^2}(2\mu_k - x_{k1} - x_{k2}) = -\frac{2}{(\sigma^2)^2}(\mu_k - \bar{x}_k),$$

where $\bar{x}_k = \frac{x_{k1}+x_{k2}}{2}$, and so

$$E\left(\frac{\partial^2 L}{\partial \mu_k \partial(\sigma^2)}\right) = 0 \tag{97}$$

This tells us that the off-diagonal elements in the Fisher information matrix will be zero, thus simplifying later calculations.
Returning to look at the diagonal elements, from (92) and (93),

$$\frac{\partial^2 L}{\partial \mu_k^2} = \frac{2}{\sigma^2} \tag{98}$$

and

$$\frac{\partial^2 L}{\partial(\sigma^2)^2} = -\frac{M}{(\sigma^2)^2} + \frac{1}{(\sigma^2)^3}\sum_{i=1}^{M}\sum_{j=1}^{2}(x_{ij}-\mu_i)^2$$

So,

$$E\left\{\frac{\partial^2 L}{\partial(\sigma^2)^2}\right\} = -\frac{M}{(\sigma^2)^2} + \frac{1}{(\sigma^2)^3}.M.2\sigma^2 = \frac{M}{(\sigma^2)^2} \tag{99}$$

Hence, since (96) and (97) give that the Fisher information matrix is diagonal, from the definition of F, (98) and (99),

$$\begin{aligned} F &= \left\{\prod_{k=1}^{M} E\left(\frac{\partial^2 L}{\partial \mu_k^2}\right)\right\} . E\left\{\frac{\partial^2 L}{\partial(\sigma^2)^2}\right\} \\ &= \left(\frac{2}{\sigma^2}\right)^M \frac{M}{(\sigma^2)^2} \\ &= \frac{2^M M}{(\sigma^2)^{M+2}} \end{aligned}$$

Since the Fisher information is independent of μ_i, choosing a uniform prior on the μ_i will give that the only dependence of the message length on μ_i will be via the (log-)likelihood, giving that

$$(\mu_i)_{MML} = (\mu_i)_{ML} = \frac{x_{i1}+x_{i2}}{2} = \bar{x}_i$$

Any choice of prior on σ (except a prior which has a value of 0 over some interval and thereby making some estimates impossible) leads (Dowe and Wallace, 1997) to a consistent estimate.

The "conjugate" prior, $h_\sigma(\sigma) \propto 1/\sigma$, gives

$$(\sigma^2)_{MML} = \frac{2\sum_{i=1}^{M}\left(\frac{x_{i1}-x_{i2}}{2}\right)^2}{M} = \frac{\sum_{i=1}^{M}\left(\frac{x_{i1}-x_{i2}}{2^{1/2}}\right)^2}{M} \asymp N(\sigma^2, \frac{2}{M}) \tag{100}$$

So, $(\sigma^2)_{MML} \rightarrow \frac{M\sigma^2}{M} = \sigma^2$ as $M \rightarrow \infty$, and $\text{plim}((\sigma^2)_{MML}) = \sigma^2$.

The inconsistency of Maximum Likelihood in doing mixture modelling was due to Maximum Likelihood's inability to acknowledge an appropriate degree of uncertainty in the parameter estimates of a discrete distribution.
The inconsistency of Maximum Likelihood in the Neyman-Scott problem is due to Maximum Likelihood's inability to acknowledge an appropriate degree of uncertainty in the parameter estimates of a discrete distribution – namely, its not acknowledging the uncertainty in estimating the μ_i of the Gaussian distribution.

The Neyman-Scott problem is one of estimating a number of parameters that increases as the data increases. Mixture modelling has an element of this problem to it in that we do not know in advance how many components to fit to our data.

A classical method known as Akaike's Information Criterion (AIC) which uses Maximum Likelihood for estimation but penalises the number of parameters used in an attempt to avoid over-fitting is inconsistent for both the Neyman-Scott problem and for mixture modelling.

MML was and is consistent for both problems, as it[10] always is.

Exercise

If you know what Akaike's Information Criterion (AIC) is, show that it is inconsistent in estimating σ for the Neyman-Scott problem.

We note in passing and might show later that Maximum Likelihood has similar problems with single factor analysis and multiple factor analysis, both of which have been done by MML (Wallace and Freeman, 1992; Wallace, to appear) with rather impressive results.

[10] or an improved approximation to Strict MML (Wallace and Boulton, 1975; Wallace and Freeman, 1987)

11 Decision trees, decision graphs and applications

A binary tree can[5] often be defined recursively in terms of its root, its left sub-tree and its right sub-tree. Any tree with a root can similarly be recursively defined.

11.1 Decision trees

A decision tree (or "classification tree") can be defined to be a tree with a root such that every interior (non-leaf)[6] node has a test conducted at it and every leaf has a class (or probability vector of the classes) in it.

For example (please draw your second-favourite decision tree) :

[5] ©David L. Dowe 1997-1998

[6] interior nodes are also known as non-leaf nodes. For a decision tree, these are also known as "split" nodes.

Now, please draw your favourite decision tree :

Decision tree induction is an example of the *regression* problem in that, given the values of several *explanatory* variables, we wish to have a model of another variable, a *goal* variable. Linear regression, polynomial regression and decision tree induction are all examples of the regression problem.

For decision tree induction and other regression problems, we typically assume that the values of all the explanatory variables are known to both sender and receiver and that it remains to transmit as concisely as possible a two-part message conveying a (decision tree) hypothesis, H, and then the data (D) given H.

The encoding of a decision tree (hypothesis) includes an encoding of the structure of the tree and an encoding of the leaf probabilities. The encoding of the data given the hypothesis includes, for each datum[7] $x \epsilon D$, encoding the datum given the leaf probability.

Whereas mixture modelling is often referred to as unsupervised learning because we do not know in advance how many components there are or what things go into each component, decision tree induction is often referred to as supervised learning because we give examples of what goes in which class and then try to learn the theory.

Forest example.

Decision tree inference has been studied by many, perhaps first by Quinlan (1986), also by Quinlan and Rivest (1989) using Minimum Description Length (MDL) and Wallace and Patrick (1993) using MML.
J. R. Quinlan has a more recent (1992) popular program called C4.5, which has recently been updated.

[7] having followed it through the decision tree to the relevant leaf node

11.1.1 Coding of binary decision trees with two leaf classes

We recall that we assume that both sender and receiver have the values of the regressor variable, and that it remains for the sender to send the receiver a two-part message conveying the decision tree theory, H, and then for each datum, the value of the desired attribute.

The encoding of data from the two leaf classes is as for a binomial distribution[8].
The encoding of the structure of the tree is as in (Quinlan and Rivest, 1989) :

$$< \text{root node} >< \text{left sub} - \text{tree} >< \text{right sub} - \text{tree} >,$$

where each interior[9] node is encoded by a "1" and each leaf node is encoded by a "0".

Encoding of the structure of the tree

A finite sequence of 0s and 1s which has
(i) at least as many 1s as 0s until its last symbol,
(ii) a "0" as its last symbol, and
(iii) which has 1 more occurrence of 0 than of 1
can be used to encode the structure of a decision tree.

Example

$< 0 >$ encodes the tree which has one node, the root. Call it T_0.
$< 100 >$ encodes the tree which has a leaf as each child of the root. Call it T_{100}.
$< 10100 > \quad = \quad < 1(0)(100) >$ encodes the tree with T_0 as its left sub-tree and T_{100} as its right sub-tree. Call it T_{10100}.

Exercise

Draw the structures of these three binary trees, T_0, T_{100} and T_{10100}.

[8] and, indeed, the encoding of data for M classes is as for an M-state multinomial distribution.
[9] or non-leaf or "split"

Recall your second-favourite and favourite binary tree structures from Section 11.1.

Exercise

Using our prefix code for binary tree structures, for both of these binary trees in turn, state the binary string encoding the structure.

Random walks, prefix codes and codes of binary tree structure

Recall the requirements above for a finite binary sequence to encode the structure of a binary tree.

Now, consider an arbitrarily long random sequence with 0s and 1s chosen equi-probably. For any such sequence, with probability 1, there will be a time when the symbol 0 has occurred exactly once more than the symbol 1 and, furthermore, there will be a first such time. Let us now consider the first time that the tally of 0s exceeds the tally of 1s, and store in our code-book the binary string which stops just as the count of 0s over-takes the count of 1s.

Fact 1 :

All such initial sub-sequence binary strings end with a 0.

Fact 2 :

The set of all such initial sub-sequence binary strings forms a prefix code.
(Why?)

Exercise

Generate a random string of 0s and 1s from the toss of a fair coin or a fair (pseudo-)random number generator or some such.

Using our prefix code for binary tree structures, draw out the tree structure corresponding to this binary string.

Hint : Use the fact that the tree structure is encoded recursively.

Exercise :

Use Facts 1 and 2 immediately above to explain that

$$\sum_{\text{all binary tree structures}} 2^{\text{length of binary code of tree structure}} = 1 \qquad (101)$$

Encoding of the split attributes and leaf probabilities

The code for the decision tree hypothesis includes
1 (i) the code for the structure of the tree (as above),
1 (ii) for each split node, a code for which attribute is being split on, and
1 (iii) for each leaf node, a code for the class probabilities.

We have dealt with 1 (i) above.

The encoding for 1 (ii) depends upon the number of regressor attributes.
If there are K of these, then we could encode the choice of attribute with a message of length $\log_2(K)$ bits. However, if we look above the current node to the root node and see that (say) k splits have already been done above us (where $0 \leq k < K$), then only $K - k$ attributes are available to be split on, and we encode the choice of split attribute with a message of length $\log_2(K - k)$ bits.

The encoding for 1 (iii) is identically the same as for a binomial distribution (as in Section 8.1.1).

This completes the encoding of the decision tree hypothesis, H.

Encoding the data given the decision tree

To encode the second part of the message, the data, D, given H, follow each datum, $x \epsilon D$, through to its relevant leaf node. Each leaf node can be treated as a separate binomial distribution.
The cost of the binomial probability coefficients from part 1 (iii) of the message and the cost of the second part of the message are given by equation (83) applied to the binomial distribution.

11.1.2 Coding of binary decision trees with M leaf classes

This section is identical to Section 11.1.2 above except that we now change from the binomial distribution[10] to the multinomial distribution.

Analogously as with Section 11.1.2, the encoding for 1 (iii) and for the second part of the message is identically the same as for a multinomial distribution (as in Section 8.1.2).

Note that here in Section 11.1.2 (and above in Section 11.1.1), by using a random walk to encode binary trees, we are using the Quinlan and Rivest (1989) code of

$$P_{\text{split}} \;=\; P_{\text{leaf}} \;=\; \frac{1}{2} \;\;, \tag{102}$$

encoding each leaf as a "0" and each internal node as a "1".

11.1.3 Coding of ternary decision trees

Above, in dealing with binary trees in Sections 11.1.1 and 11.1.2, we used the Quinlan and Rivest (1989) code of equation (102), with

$$P_{\text{split}} \;=\; P_{\text{leaf}} \;=\; \frac{1}{2}$$

which assumed that split and leaf nodes are equally likely.
This relied on the fact pointed out in Section 11.1.1 (Facts 1 and 2 and equation(101)) that
any random sequence of 0s and 1s has a unique prefix[11] corresponding to a unique binary tree.

This is *not* the case for a simple encoding of ternary tree structures, as the following "exercise" is intended to help demonstrate.

[10] which is a special case of the multinomial distribution with $M = 2$
[11] namely, that prefix where the tally of 0s first exceeds the tally of 1s by unity

"Exercise"

Attempt, where possible, to draw a ternary tree structure given by the binary sequence 0.
Attempt, where possible, to draw a ternary tree structure given by the binary sequence 100.
Attempt, where possible, to draw a ternary tree structure given by the binary sequence 10100 = 1(0)(100).

Attempt, where possible, to draw a ternary tree structure given by the binary sequence 1000.
Attempt, where possible, to draw a ternary tree structure given by the binary sequence 1(0)(1000)(0).
Attempt, where possible, to draw a ternary tree structure given by the binary sequence 1(1000)(1000)(0).

The problem is this: for binary tree structures, we have one more 0 than 1,
and random walk theory guarantees that this will happen with a fair sequence.

However, for a ternary tree structure,

$$\text{the number of 0s} \quad = \quad 2 \times (\text{the number of 1s}) \; + \; 1$$

There is no guarantee in random walk theory that any random walk must necessarily have such a sub-sequence as a prefix. Basically, they have too many 0s.
Hence, this encoding of ternary trees and general n-ary trees by Quinlan and Rivest (1989) is inefficient, in that

$$\sum_{\text{all ternary tree structures}} 2^{\text{length of Quinlan}-\text{Rivest binary code of tree structure}} \quad < \quad 1$$

The correct way of dealing with ternary trees so as to avoid this inefficiency is (Wallace and Patrick, 1993):

$$P_{\text{split}} \; = \; \frac{1}{3} \qquad \text{and} \qquad \text{P}_{\text{leaf}} \; = \; \frac{2}{3}$$

We generalise this in the next section, Section 11.1.4.

11.1.4 General coding of n-ary decision trees

The "*arity*" of an attribute is the number of values it takes.
So, a binary attribute is two-valued, and has arity of 2.
A ternary attribute is three-valued, and has arity of 3.
An n-ary attribute is n-valued, and has arity of n.

For any attribute, we certainly need to have

$$P_{\text{split}} \ + \ P_{\text{leaf}} \ \ = \ \ 1 \tag{103}$$

We also wish to have

$$P_{\text{split}} \times (\text{arity of parent node}) \ + \ \text{P}_{\text{leaf}} \times 0 \ \ = \ \ 1 \tag{104}$$

because we can argue that we want the expected number of nodes at each level of the tree to remain constant.
These two simultaneous equation lead us to the encoding probabilities:

$$P_{\text{split}} \ = \ \frac{1}{\text{arity of parent node}} \qquad \text{and} \qquad \text{P}_{\text{leaf}} \ = \ \frac{(\text{arity of parent node}) - 1}{\text{arity of parent node}} \tag{105}$$

An alternative justification for this goes along the lines of what we showed for ternary trees in Section 11.1.3 – namely, for an n-ary tree,

$$\text{the number of leaves} \qquad = \qquad (\text{n} - 1) \times (\text{the number of splits}) \ \ + \ \ 1$$

We note that this generalises equation (102) in Section 11.1.2 and the claim in Section 11.1.3.

Remark about MML and consistent decision tree inference

Spare some thought for Maximum Likelihood when looking at decision tree inference. For many data-sets, sufficient splitting will lead us to some or all leaf nodes being very pure or even totally pure in one class. Thus, Maximum Likelihood will enjoy splitting and over-fitting unless it has some way of being penalised.

Recall also from Sections 8.1.1 and 8.1.2 that Maximum Likelihood will be a bit biased for small samples or where the leaf is very pure in one class.

The message length keeps us honest and discourages us from over-fitting.

(continued)

11.1.5 Continuous-valued attributes in decision trees

We begin by observing that

$$\sum_{i=1}^{\infty} \frac{2^{i-1}}{2^{2i-1}} = \frac{2^0}{2^1} + \frac{2^1}{2^3} + \frac{2^2}{2^5} + \frac{2^3}{2^7} + \ldots = \frac{1}{2} + \frac{2}{8} + \frac{4}{32} + \frac{8}{128} + \ldots = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \ldots = 1$$

and that this can be used to generate a code-book (as follows) :

0		1
100		011
101		010
11000		00111
11001		00110
11010		00101
11011		00100
1110000	or	0001111
1110001		0001110
1110010		0001101
1110011		0001100
1110100		0001011
1110101		0001010
1110110		0001001

1110111 0001000
111100000 000011111
:

Suppose we want to split on some continuous-valued attribute, say attribute j.
Sender and receiver know that it has (say) N_j values.
Then, we can use this code above to encode the various split percentiles[12].

Furthermore, we can do this by using the right-hand version of the code and putting a decimal point at the end and then reading it back the front as binary "decimals" to indicate the percentile at which the split occurs.

In other words :
the code-word 1 (corresponding to probability $\frac{1}{2}$) is used to designate a split at position 0.1 ($\frac{1}{2}$ of the way through the data);

the code-word 011 (corresponding to probability $\frac{1}{8}$) is used to designate a split at position 0.110 ($\frac{3}{4}$ of the way through the data);

the code-word 010 (corresponding to probability $\frac{1}{8}$) is used to designate a split at position 0.010 ($\frac{1}{4}$ of the way through the data);

the code-word 00111 (corresponding to probability $\frac{1}{32}$) is used to designate a split at position 0.11100 ($\frac{7}{8}$ of the way through the data);

the code-word 00110 (corresponding to probability $\frac{1}{32}$) is used to designate a split at position 0.01100 ($\frac{3}{8}$ of the way through the data);
etc.

Since attribute j has N_j different values, there are $N_j - 1$ different possible "*cut-points*" between these values. (If $N_j = 1$, then there is no need to split on this attribute.)

Two alternative coding systems to that detailed above are as below:
(i) (Quinlan and Rivest, 1989) uses the fact that there are $N_j - 1$ different possible cut-points and then encodes them all as being equally likely. These will have code-words of length $-\log_2\left(\frac{1}{N_j - 1}\right) = \log_2(N_j - 1)$

(ii) (Quinlan and Rivest, 1989) proposes (but neither implements nor uses) as an alternative idea using the (Fisher information) uncertainty region (Wallace and Boulton, 1968) to encode the cut-points. This is not elaborated upon.

With any luck, the above gives a thorough account of the information-theoretic encoding

[12]The first author was supported by Australian Research Council (ARC) Large Grants Nos. A49602504 and A49330656.

of decision trees.

MML gives a good account of how inference might (and should?) be done in terms of providing an objective function to optimise. However, knowing which function to optimise is sometimes not much more than a good start in finding that optimum. We shortly return to discuss this search problem, but we first set an exercise and then express an afterthought re the choice of priors for the leaf distributions.

Exercise

1. Decide upon a small number of discrete (binary) attributes, and one continuous attribute with 5 values.
1 (i) Draw (below) a fairly large decision tree into whose interior nodes it is possible to split on various values of the above attributes.
1 (ii) Into each interior node, place a split on an attribute, making sure that this split is legal given all the splits that have already taken place earlier[13] in the decision tree message. For a continuous-value attribute, we need to specify where we are splitting.
1 (iii) Think about some probabilities for each leaf class (but do not do anything more yet than just think).
2 Place some data in the leaves according to a multinomial (binomial) distribution.

[13] we typically code recursively from left to right

Exercise

Assuming a coding scheme for the above,
calculate a code-length for parts 1 (i) and 1 (ii) of the message.

Go further, and try and actually construct a code for parts 1 (i) and 1 (ii) of the message.

Part 1 (iii) of the message consists of the Bernoulli co-efficients of each leaf in turn. Part 2 of the message consists of the data given these Bernoulli probability estimates.
If you can, cost parts 1 (iii) and part 2 separately.

If you can not do that, at least give the total cost of parts 1 (iii) and 2.

What is the total length of your message conveying your decision tree and the data?

11.1.6 Afterthought re choice of priors for leaf distributions

Recall from Section 8.1.1 on the binomial distribution that, for $\alpha > -1$ and $\beta > -1$,

$$\int_0^1 p^\alpha (1-p)^\beta \, dp \;=\; \frac{\alpha!\;\beta!}{(\alpha+\beta+1)!} \tag{106}$$

If we were to use a prior on the categories in the leaves of

$$h(p) \quad = \quad \frac{(2\alpha+1)!}{(\alpha!)^2}\, p^\alpha (1-p)^\alpha$$

of course, as Maximum Likelihood does not use priors, we still get $(\hat{p})_{ML} \;=\; \frac{x}{N}$.

We leave it as an exercise to show that we also would get

$$(\hat{p})_{\text{posterior mean}} \;=\; \frac{x+\alpha+1}{N+2\alpha+2} \qquad \text{and} \qquad (\hat{\mathrm{p}})_{\mathrm{MML}} \;=\; \frac{\mathrm{x}+\alpha+1/2}{\mathrm{N}+2\alpha+1}$$

If there is more than one leaf, then it is possible to regard α as a parameter to be estimated.

Wallace and Patrick (1993) discusses estimating α (with Maximum Likelihood) for the leaves of decision trees.

For $\alpha = 0$, this corresponds to a uniform prior.

Exercise(s)

Draw this prior for $-1 < \alpha < 0$.
Draw this prior for $\alpha > 0$.

11.1.7 The search problem for (MML) decision trees

Re-cap on the search problem for mixture modelling:
the "Expectation Maximisation" (EM) algorithm for re-iterating
parameter estimation followed by partial (re-)assignment according to this;
and then re-estimating, etc. until convergence.

The search in mixture modelling for (e.g.) the number of components is messier.

As with the search problem for mixture modelling and in general, knowing the desired objective function (in our case, the message length) is a good start to finding an optimum, but it does not of itself give us an optimum. What it does give us is a way of comparing two theories.

For most choices of prior, for the multinomial, Gaussian and Poisson distributions, we can simply write down the MML estimator analytically.

For the von Mises distribution, the MML estimator and even the Maximum Likelihood estimator can be specified by equations, but they still have to be solved for numerically.

For mixture models and decision trees, one is trying to select a model class (e.g. number of mixture components or number and location of interior split nodes) as well as estimating parameter values. This makes the search space discrete (or "gritty") in places and then continuous in other places. In short, it can and does in general lead to some messy search problems.

In practice, we search for the optimum guided by the best heuristics we can muster.

As for decision trees, there are many ways in which could rummage amongst possible decision trees in search of the MML tree. One way might be genetic algorithms. A better way way would probably be simulated annealing (which is a greedy algorithm modified to sometimes take random steps "for the worse", something which happens less and less often as the "temperature" decreases).

The Wallace and Patrick (1993) decision tree program (available from Monash Computer Science cs.monash.edu.au at ~csw/dtree/) uses a greedy algorithm combined with a lookahead.

Lookahead

Before playing a move, a chess player will typically look ahead a number of moves to that player's "search horizon", and then choose the current move subject to what was seen at the various options available at the search horizon.

Imagine now a decision tree which is being grown from the root down. What we could do with the current draft of the decision tree is to look at each node in turn which has not yet been split upon, and consider for each of these in turn all the legal[14] splits that might be performed at them. In addition to all these split options, we also include as an option declaring every unsplit node to be a leaf node and to complete the growing of the tree.

Having considered these various options, we could then take that option which gives rise to the shortest message length.
This way of choosing a "move" in growing a decision tree is called *lookahead* 1, in that it looks at all possible options 1 level ahead and then chooses the best one. We iterate the lookahead 1 process until the tree is completely grown.

Lookahead n looks ahead n steps and then greedily chooses that single step which had the shortest message length on the horizon of looking ahead n steps.

Looking ahead further has the advantage of being more likely to find a shorter message length. However, as with chess, although we generally expect that looking ahead further is preferable to not looking ahead very far, there is no guarantee that looking ahead further will necessarily always lead to a better choice than a shallower search will. Furthermore, looking ahead a long way generally slows the search down substantially.

[14] at any given node, we can only split on discrete attributes that have not been split on at an ancestor node earlier in the tree or on continuous attributes

11.2 Decision graphs

Recall equations (103) and (104) :

$$P_{\text{split}} \; + \; P_{\text{leaf}} \quad = \quad 1$$

and

$$P_{\text{split}} \times (\text{arity of parent node}) \; + \; \text{P}_{\text{leaf}} \times 0 \quad = \quad 1$$

in Section 11.1.4 expressing[15] the fact that, in a decision tree, all nodes must be either a split node or a leaf node[16] and also the fact that the expected number of children is the arity times the probability of a split node (plus 0 times the probability of a leaf node).

Decision graphs[17] are a generalisation of decision trees. Whereas all non-leaf interior nodes in a decision tree are split nodes (corresponding to "and"), decision graphs permit interior nodes to be join nodes (corresponding to "or") as well as split nodes.

The upside of generalising something is that the language becomes more expressive and some things can now be expressed more concisely than previously. For example, a decision tree which has two or more identical or almost identical sub-trees will probably gain by having the relevant nodes joined so that all the duplicated sub-trees can become one.

Exercise

Draw a decision tree with two (or more) identically replicated sub-trees, and then draw a decision graph taking advantage of the fact that joining the roots of the replicated sub-trees will make the encoding more efficient.

The downside of having a more expressive language is that the search for the optimum typically takes longer.

In encoding a decision graph, the fact that we now have join nodes modifies decision tree equation (103) to be

[15] apologies if the equation labels here might be a little bit out

[16] since, for a decision tree, all interior non-leaf nodes are split nodes

[17] or "classification graphs"

$$P_{\text{split}} \; + \; P_{\text{Join}} \; + \; P_{\text{leaf}} \;\; = \;\; 1 \tag{107}$$

with a decision tree being identical to a decision graph with $P_{\text{Join}} \; = \; 0$.

$$P_{\text{split}} \times (\text{arity of parent node}) \; + \; \frac{\text{P}_{\text{Join}}}{2} \; + \; \text{P}_{\text{leaf}} \times 0 \;\; = \;\; 1 \tag{108}$$

The theory of (MML) decision graphs was first expounded in Oliver and Wallace (1991)[18], and subsequently re-iterated in Oliver, Dowe and Wallace (1992) and Oliver (1993).
If one solves the two equations (107) and (108) above for the three unknowns, P_{split}, P_{Join} and P_{leaf}, one arrives at

$$P_{\text{split}} \;\; = \;\; \frac{1 - \frac{P_{\text{Join}}}{2}}{\text{arity of parent node}} \tag{109}$$

and

$$P_{\text{leaf}} \;\; = \;\; 1 - P_{\text{Join}} - P_{\text{split}} \;\; = \;\; 1 - P_{\text{Join}} - \frac{1 - \frac{P_{\text{Join}}}{2}}{\text{arity of parent node}} \tag{110}$$

Since two equations in three unknowns leave one variable undetermined, in encoding a decision graph, it is necessary to (infer and) state a value for P_{Join}.

11.2.1 Communicating a graph's topology

If we recall from Section 11.1.1 the encoding of a two-part message entailing a decision tree, we will note that part 1 (i) of the message now changes so that we encode (P_{Join} followed by) the topology of the graph.

The encoding of the topology of the graph goes along the following lines, but gets a bit messy for reasons we shall shortly come to.

All split nodes and leaf nodes are quite easily coded as with decision trees.

The difficulty comes in the encoding of join nodes. Join nodes come in pairs. Imagine growing a decision graph and coming to a join node whose partner has not yet been grown. This seems like a problem, but we deal with it by perservering and growing the decision graph in "*generations*" as follows:

[18] which was subsequent to the Wallace and Patrick (1993) decision tree journal article

The first generation of a decision graph's topology

Initially, grow the decision graph from the root as far as is possible until all nodes are either leaf nodes or join nodes. Now, go through labelling the nodes which have a partner they can join with. There will necessarily be an even number (say E) of such nodes. Now, encode the pairing(s), of which there will be

$$\frac{\binom{E}{2} \times \binom{E-2}{2} \times \ldots \times \binom{2}{2}}{(E/2)!} = \frac{1}{(E/2)!} \times \frac{E(E-1)}{2!} \times \frac{(E-2)(E-3)}{2!} \times \ldots \times \frac{2 \times 1}{2!}$$
$$= \frac{E!}{(E/2)!\ (2!)^{E/2}}$$

Subsequent generations of a decision graph's topology

Any nodes which could have joined but did not are labelled as "*Old*". For the next generation, we continue from having made all the possible joins above, until we are again at a situation where it is not possible to do any further splitting without first doing one or more joins (or possibly because we have already completely grown the tree). We now have "*New*" nodes which have been created this generation and *Old* left-over nodes from the last generation (or earlier). The only joins which can be done are between two *New* nodes or between an *Old* node and a *New* node, since we assume that two *Old* nodes would have been joined in an earlier generation at the first available opportunity.

The combinatorics gets messier, but there is a *recurrence relation* of the number of ways in which the joins can be done.

We iterate through generation after generation of the graph until its topology has finally been communicated to the receiver.

11.2.2 Communicating the decision graph

Having communicated the decision graph's topology as in Section 11.2.1,
as in Section 11.1.1 it now remains only to transmit
1 (ii) for each split node, a code for which attribute is being split on, and
1 (iii) for each leaf node, a code for the class probabilities; and
2. The data, D, given the decision graph.

11.3 Application of decision trees to bush-fire prediction

The data here[19] is from the Mallee region of North-West Victoria, Australia, for the 3833 days from 1 September 1979 until 28 February 1990.

The *dependent* (or *goal*) variable to be (probabilistically) predicted is the binary-valued variable of whether or not a bush-fire occurred.

The data-set also included 10 explanatory variables.

Although it is now perhaps trivial that MML lends itself fairly natural to probabilistic prediction, perhaps the first papers (other than those going back to 1952 concerning probabilistic prediction and football) in which the connection between MML and probabilistic prediction are made explicitly clear are (arguably) the Wallace and Patrick (1993) paper on MML decision trees and the Dowe and Krusel (1993, 1994) applications to bush-fire prediction.

Include Dowe and Krusel (1993) results.

[19] ©David L. Dowe 1997-1998

11.4 Application of decision graphs to protein folding

We might recall from Section 9.1.6 that proteins consist of a repeating Nitrogen-(α-Carbon)-(β-Carbon)- ... chain, with an amino acid attached to each α-Carbon[20].

We might also recall that proteins are often considered as having 3 secondary structures, Extended, Helix and Other: E, H and O, which are used to describe the local shape (or *secondary structure*) at a point. The reason that this is done is that the goal of predicting the tertiary (or 3-dimensional) structure of a general protein is from its *primary* amino acid sequence is currently considered intractable, and it is thought that the study of secondary structure will provide some insight to assist the study of tertiary structure.

There are 20 naturally occuring amino acids (AAs), with their names, 3-letter abbreviations and 1-letter abbreviations (possibly) being given below or on the next page (as taken from Schulz and Schirmer, 1983?):

Recall that Section 11.3 considered the application of decision trees to bushfire data. Since all the explanatory variables were continuous for this problem, we had to use the theory

[20]The first author was supported by Australian Research Council (ARC) Large Grants Nos. A49602504 and A49330656.

from Section 11.1.5 on splitting on continuous-valued attributes in decision trees.

For this problem of protein secondary structure prediction, the explanatory variables here are the Amino Acids (AAs). As such, we use the theory of decision graphs (from Section 11.2) accompanied by the fact that all our attributes are discrete (see Section 11.1.4).

Assuming that both sender and receiver already have the amino acid values in sequence, the idea is to try and infer a theory of which secondary structure conformation (E, H or O) will appear at a certain site as a function of the surrounding amino acids :

An inefficiency in this model is the fact that it fails to take account of the serial correlation in the secondary structure sequence. Correcting this inefficiency will shorten the message length and almost certainly improve predictive accuracy. There is certainly room for further research here.

The data used in (Dowe, Oliver, Allison, Dix and Wallace, 1993) is listed below.
It comprised 75 GOR (Garnier-Osguthorpe-Robson) proteins, totalling
$2515 + 3790 + 6452 = 12757 \approx 13000$ amino acids.

We split these into 8 non-homologous groups so as to make the prediction problem fairer.

Further below are the results obtained from this study.

From (Dowe, Oliver, Allison, Dix and Wallace, 1993) :

12 Probabilistic Finite State Automata (PFSAs)

Finite State Automata (FSAs), or finite state machines, are determined by
a fixed, finite number of states,
an alphabet of characters,
and a mapping $m : \text{States} \times \text{Alphabet} \rightarrow \text{States}$,

which tells the machine in the current state that if it encounters a certain symbol from the alphabet then it should head to a new state as determined by the mapping.

FSAs can also be regarded as *grammars*.

Question and observation :

FSAs are not necessarily as computationally powerful as (Universal) Turing Machines. True or False?

If not, why not?

Of course, just as we have FSAs, we also have probabilistic finite state automata (PFSAs), for which the mapping m above is probabilistic.

And, of course, given a body of data which we suspect as having come from some (probabilistic) grammar or *language*, the inference question arises of how best to infer such a grammar.

The inference of PFSAs using MML was first studied in Wallace and Georgeff (1983) and Georgeff and Wallace (1984).

From wherever it is attributed to being from :

From wherever it is attributed to being from :

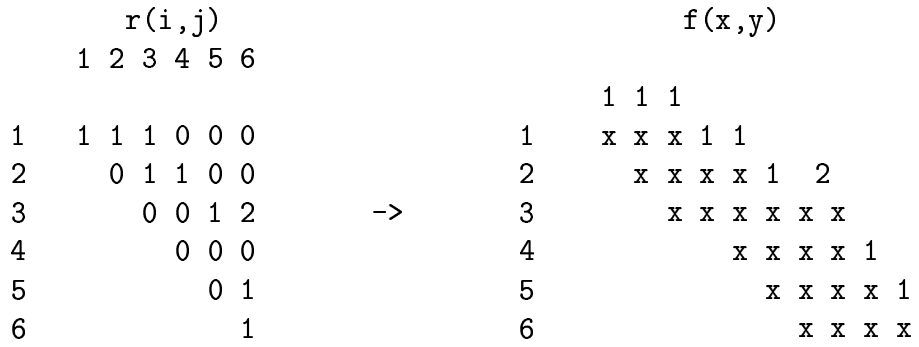
From wherever it is attributed to being from :

From wherever it is attributed to being from :

From Wallace and Georgeff (1984) :

13 DNA alignment

Consider an attempt to align some DNA fragments, as in the Figure[21] following.

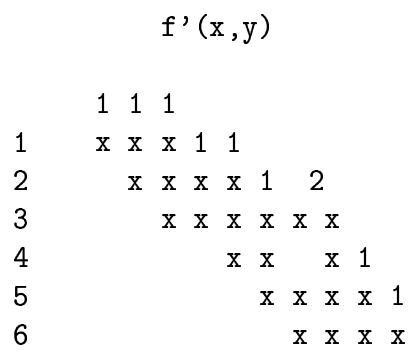


Entry $r(i,j)$ gives the number of fragment stacks beginning in position i and finishing in position j.

The fragment match on the right is an example of something like the raw data that a biologist might get from doing DNA restriction site mapping.

There is a 1-to-1 mapping between the stack on the left and the fragment match on the right.

The problem arises if there is an error in the fragment match on the right, and we have (e.g.) a hole :



The inference problem which arises is, given that we have an error in our data, what model was it most likely to have come from.

13.1 A message for DNA restriction mapping alignment

On way of encoding the fragment match with a hole in it, f', would be to encode f followed by a message saying that there is one hole and that the hole is in position $(x,y) = (4,3)$,

[21] which was supplied by Daniel Loo, who we acknowledge and thank.

i.e., the 4th row and 3 positions in from the left.

Then, to encode f', we need to encode f and the error(s).

Since there is a 1-to-1 mapping between r and f, we can therefore encode f' instead by just encoding r and the error(s).

13.1.1 Encoding the stack, r

For those familiar with the problem or who study the stack matrix, $r(i,j)$, long enough, the shape of r is determined by beginning at the top-left hand corner $(1,1)$ and making successive moves of right, down, or down-and-right. Other than following these numbers along, every other entry is a 0. And, we can encode the right, down, or down-and-right as coming from a multinomial distribution (see Section 8.1.2).

Now, these numbers are all non-negative integers. As such, they could be encoded by (e.g.) a Poisson distribution or a geometric distribution.

13.1.2 Possible encoding of the fragment match with errors

Encode the error-free stack, r, by encoding a trinomial distribution of the path of the non-zero numbers.

Encode these non-negative numbers as coming from (e.g.) a Poisson distribution or (e.g.) a geometric distribution.

The receiver now has the error-free fragment match, f.

Now encode the error(s), if any, to give f'.

14 Linear regression and Causal nets

14.1 Linear regression

(Baxter and Dowe, 1994, 1996; Wallace, 1996; Viswanathan and Wallace, 1999).

Recall the Gaussian distribution $N(\mu, \sigma^2)$ from Section 8.3.

Letting $a_0 = \mu$, Gaussian regression can be regarded as finding a horizontal line,
$y = a_0 + N(0, \sigma^2)$
of best fit

For linear regression, letting
a_0 be the constant term,
a_1 be the gradient and (as usual)
σ^2 be the variance,
we wish to find a line

$y = a_0 + a_1 x + N(0, \sigma^2)$

of best fit.

How do we do this?

We calculate the likelihood function,
$f(y|a_0, a_1, \sigma^2)$.
We calculate the second derivatives of the likelihood function, and their expectations. The determinant of the matrix of these gives the Fisher information.

We also need a Bayesian prior on the parameters, a_0, a_1 and σ^2.
Recall equations (83) and (84), where k_n is a lattice constant that can safely be assumed to be $\frac{1}{12}$.

We choose the model which minimises the message length.

The simulation results in (Baxter and Dowe, 1994, 1996) show MML to be at least as good as all the available classical rivals considered.

14.2 Quadratic regression

For quadratic regression, letting
a_0 be the constant term,
a_1 be the coefficient of x,
a_2 be the coefficient of x^2 and (as usual)
σ^2 be the variance,
we wish to find a curve

$y = a_0 + a_1 x + a_2 x^2 + N(0, \sigma^2)$

of best fit.

How do we do this?

By the methods above.

14.3 Polynomial regression

(Wallace, 1997+; Viswanathan and Wallace, 1999).

$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_d x^d + N(0, \sigma^2)$

We have to first encode the degree, d, of the polynomial.
We end up with the first part of the message encoding $d, a_0, a_1, \ldots, a_d, \sigma^2$.

The slight tricks to bear in mind here are
(i) we must encode d so that the rest of the message makes sense
(ii) d comes from a discrete distribution (of range $0, 1, 2, \ldots$) whereas the a_i are from a continuous distribution.

(Wallace, 1997+; Viswanathan and Wallace, 1999) shows the MML approach to be superior to the recently developed V-C (Vapnik-Chervonenkis) dimension method, and vastly superior to better known classical rivals.

14.4 Causal nets

(Wallace, Korb and Dai, 1996.)

A causal network is a directed acyclic graph (DAG) of (purported) (linear) causal relations between variables.

Where several variables are deemed to be causally affecting another, we do a linear regression (see Section 14.1) to model a variable as a function of those supposed to be causally affecting it.

15 Factor analysis

A statistical *factor* is something which captures an underlying piece of behaviour in two or more variables.

Example 1

For example, taller animals tend to be broader (front to back) and wider (left to right). This is not always true, but there is a trend for this to happen. Ditto amongst humans.

We could propose a factor called *size*. The *factor loads* would then consist of how much each of height, breadth and width contributed to size. Each animal or human would have a *factor score*, which would be a measure of its/her/his size.

Example 2

For the petrol heads, we could consider a variety of petrols and how well they perform on a variety of engine. We could call such a factor *octane rating*. The *factor loads* would then consist of how much each engine contributes to octane rating. The *factor scores* would then be the octane ratings of the petrols, measuring how well (or badly) each petrol does on the variety of engines.

Example 3

The following proposed factor can be emotionally charged. Imagine a variety of aptitude tests in linguistic, mathematical, etc. abilities. One could propose a factor called I.Q. The factor loads in such a model would be the contribution of the various tests to I.Q., and the factor scores would be the various I.Q.s of the various people.

When factor analysis is done, it is assumed (for simplicity) that the variables combine linearly to form the factor. This probably makes sense when we combine height, breadth and width to form the size factor, but it might make less sense if we also included weight.

Also, as well as assuming that variables combine linearly, it is also traditional (for simplicity) to assume that the variables are Gaussian.

So, assume – as everyone else does – linear Gaussian factors.

15.1 Single factor analysis

(Wallace and Freeman, 1992.)

The simulation results comparing alternative methods are quite impressive.

15.2 Multiple factor analysis

(Wallace, 1995+, 1999.)

The simulation results comparing alternative methods are very impressive.

16 Regression with the spherical von Mises-Fisher distribution

(Dowe, Oliver and Wallace, 1996.)

17 MDL and MML: "one-part" messages and two-part messages

This section briefly compares and contrasts Minimum Description Length (MDL) (J. J. Rissanen, 1978, etc.) with MML (C. S. Wallace et al., 1968, 1969, 1970, 1973, 1973, 1975, 1975, etc.).

Recall Section 3.3 (General Inference scenario), pp14-15 and following sections.

Let $\vec{x} = D$ be some data.

Define

$$I_0(\vec{x}) = -\log_2(r(\vec{x})) = -\log_2\left(\int h(\vec{\theta}) f(\vec{x}|\vec{\theta})\ d\vec{\theta}\right)$$

and let $I_1(\vec{x})$ be the length of the shortest two-part message conveying an hypothesis[22], H, and the data, $\vec{x}$.

$$\begin{aligned} I_1(\vec{x}) &= \min_H\ (-\log_2(h(H)) - \log_2(f(\vec{x}|H))) = \min_H\ (-\log_2(h(H) f(\vec{x}|H))) \\ &= \min_H\ (-\log_2(p(\vec{x}, H))) \end{aligned}$$

$I_0(\vec{x})$ is the length of the shortest one-part message for conveying the data, $\vec{x}$.

$I_1(\vec{x})$ is the length of the shortest two-part message for conveying an hypothesis followed by the data given the hypothesis.

$$\begin{aligned} I_1(\vec{x}) - I_0(\vec{x}) &= \min_H\ (-\log_2(p(\vec{x}, H)) + \log_2 r(\vec{x})) = \min_H\ \left(-\log_2(\frac{p(\vec{x}, H)}{r(\vec{x})})\right) \\ &= \min_H\ (-\log_2(g(H|\vec{x}))) \ \geq\ 0 \end{aligned}$$

[22] the MML hypothesis

Rissanen's motivation in his 1978 paper is partly the notion of Kolmogorov Complexity. As we (should) see shortly, the information content in some data can be thought of as the length of the shortest binary string needed by a Universal Turing Machine (UTM) to construct the data.

The following example highlights a crucial difference between MDL and MML. Consider the transmission of a two-part message in which the first part of the message is sent as usual but a subtle change is made to the second part of the message. Since both sender and receiver know the first part of the message after it has been sent, one could devise a code in which the second part of the message was sent assuming that the first part of the message contained the optimal inference.

(Note that this is *not* the way in which MML encodes the second part of the message. The MML encoding of $D|H$ assumes nothing about whether H is a particularly good nor particularly lousy hypothesis for D.)

The original MDL criterion (Rissanen, 1978) minimised the log-likelihood function plus the logarithm of the number of parameters, a principle usually referred to as "Bayes's Information Criterion" (BIC). For some problems (e.g. the Neyman-Scott problem of Section 10), this MDL (1978) or BIC criterion is inconsistent.

The 1987 version of MDL (Rissanen, 1978) had coding blocks spiralling (anti-clockwise) outward from the "*natural*" origin. This is not invariant under re-parameterisation, partly because it begs the issue of a "*natural* origin", and partly because the notion of anti-clockwise or some such order of the spiralling depends upon putting some sort of "natural order" on the attributes. This will not be invariant if we inter-change two axes.

Exercise

Draw two axes with outwardly spiralling coding blocks in the available space.

Interchange the two axes, and see what happens to the direction of the spiralling of the coding blocks.

A more recent version of MDL (Rissanen, 1996) uses the Jeffreys "prior" from Section 7.1. This gives a presumably invariant but presumably inconsistent estimation method.

Rissanen's work shows his laudable desire to find an estimation method which is invariant and universally consistent but most definitely not Bayesian. While this approach is laudable and perhaps a holy grail of inductive inference, we conjecture(d) in Section 7.3.3 that it is not possible.

For a longer discussion of the relative pros and cons of MDL and MML, see (Rissanen, pp223-239, 1987), (Wallace and Freeman, pp240-252, 1987) and the ensuing discussion (pp253+, 1987).

18 Turing Machines and Universal TMs as priors

Much of the material in this section is from Dowe and Wallace (SMML and Kolmogorov complexity, 1997+), Section 1 and Section 2.2.2, and need not be absorbed in great detail.

18.1 Kolmogorov complexity

We define[23] the Kolmogorov complexity (Kolmogorov, 1965)[24] of a string x with respect to some universal Turing Machine, U, to be the length, $|q|$, of the shortest input string, q, such that when q is input to U, U reads all of q, then outputs all of x, and then either stops or tries to read more input.
Given some Universal Turing Machine (UTM), U,

$$K_U(x) = \min_q \{|q| : U(q) = x \text{ and U halts or reads input}\} \quad (111)$$

where $U(q)$ denotes the output from machine, U, after it has been fed input, q.
Following on for each output string, x, we can **define** the marginal probability, ($r_U(x)$ or) $P_U(x)$, of x, as follows:
Definition:

$$P_U(x) = \sum_{q:U(q)=x \text{ and then halts or reads}} 2^{(-|q|)} \quad (112)$$

$$= \Pr(\text{U generates x from random string}) \quad (113)$$

since we insist that U then halts or reads. The halting or reading of U in this definition ensures that the set of our acceptable code-words, q, gives rise to a prefix code. It thus

[23] the notes here in Section 18.1 on MML and Kolmogorov Complexity are being revised from those below to form and article by Wallace and Dowe to appear in the Computer Journal, 1999.
[24] defined independently, and apparently earlier, by R.J. Solomonoff (1964).

follows (Kraft, 1949) that

$$\sum_x 2^{-K_U(x)} \leq \sum_x \sum_{U(q)=x} 2^{(-|q|)} \leq 1 \quad (114)$$

where U outputs x and then halts or waits to read. This natural and fundamentally important concept of Kolmogorov complexity, $K_U(.)$, has been well-studied - see, e.g., (Li and Vitànyi, 1997).

18.2 One-part and Two-part messages

A *one-part* message is, as above, an encoding, q, of a datum, x.
We say that q gives a *two-part* message for x with respect to U if there exist strings q_0 and q_1 such that $q = q_0q_1$ is their concatenation, $U(q_0) = \{\}$, $U(q) = x$, and, for all strings q_2,

$$\begin{aligned} U(qq_2) &= U(q_0q_1q_2) = U(q_0q_1)U(q_0q_2) && (115)\\ &= xU(q_0q_2) = \{\}(U(q_0))(q_1).(U(q_0))(q_2) && (116)\\ &= U(q_0).(U(q_0))(q_1).(U(q_0))(q_2) && (117)\end{aligned}$$

We take the first line as the **definition**, and the subsequent equations are consequences of this definition. In a two-part encoding, the first part of the message (here, q_0) contains the hypothesis, and the second part contains the data encoded given the hypothesis. We note that any data, x_2 (encoded using q_2), is encoded without reference to q_1 - the only reference to the encoding of x is to the first part of the message, q_0, the encoding of the hypothesis. Paraphrasing, the joint probability $Pr(q_0q_1q_2)$ can be written (adaptively) as $Pr(q_0q_1q_2) = Pr(q_0)Pr(q_1|q_0)Pr(q_2|q_0q_1)$.
Our definition above gives that $Pr(q_0q_1q_2) = Pr(q_0)Pr(q_1|q_0)Pr(q_2|q_0)$
In the sense that we thus have that, for all q_1 for all q_2, $Pr(q_2|q_0q_1) = Pr(q_2|q_0)$, it follows that q_0 encodes an hypothesis.

18.3 SMML inference re-visited

Recall the discussion on Strict MML in Section 4.5, pp25-26.

A related notion, MMLA (or Fairly Strict MML, FSMML), can also be defined.

Re-printed from C. S. Wallace (1994), presuming kind permission :

Re-printed from C. S. Wallace (1994), presuming kind permission :

18.4 SMML and Two-part code-books

Statistical inference typically requires likelihood functions. Bayesian inference further requires prior distributions on parameter values. In order to be able to make inductive inferences, we further wish to restrict ourselves to distributions known to be totally computable, i.e., distributions which return computable values (in the sense of Section 1) given any finite input string.
Consider a totally computable Bayesian prior, $h(.)$, over our parameter space and a totally computable likelihood function, $f(.|.)$, of the data given the parameters.
For each datum, x, define the marginal probability, $r(x)$, of datum, x, as being given by

$$r(x) = \int_{\theta} h(\theta) f(x|\theta) d\theta \tag{118}$$

or

$$r(x) = \sum_{H} P(H) f(x|H) = \sum_{\theta} h(\theta) f(x|\theta) \tag{119}$$

with the former being the discrete case and the latter being the continuous case. This also appears to be the logarithm of Rissanen's stochastic complexity, $SC(x)$ (Rissanen, 1989; Rissanen, 1996; Rissanen, 1978).
Since in finite time, we can only generate finitely many bit strings, in finite time we can only generate finitely many x (and $r(x)$). In practice, $r(x)$ might be non-analytical or numerical, so we might be willing to tolerate some agreed upon (and very small) tolerance error in $r(x)$.
Consider a machine, M, which takes an arbitrary $h(.)$ and $f(.|.)$ and generates for both sender and receiver a code-book of two-part messages, $< \hat{\theta} > \quad < x|\hat{\theta} >$, conveying $\hat{\theta}$ and then x given $\hat{\theta}$. How might we choose M? We consider some alternatives below.

18.4.1 A universal two-part code-book

Whether or not we know a functional form for $h(.)$, we can choose an arbitrary universal distribution, d_u, for the encoding of θ, since for all universal d_u, for some constant, C_u, for all θ $h(\theta) \leq C_u d_u(\theta)$, and so
for all $\theta \quad -\log_2 d_u(\theta) \leq -\log_2 h(\theta) + \log_2(C_u)$.
So, simply choose any old Universal Turing Machine (Vitányi and Li, 1996).
The scheme in this (universal) option tells us that if we ignore any available prior information, we can choose a universal distribution which will enable us to have a code-book whose expected length is within some fixed constant of the minimum possible expected length.
Our objection to this option (Vitányi and Li, 1996) is that the choice of universal distribution, d_u, is arbitrary, and so leads to an arbitrary constant, $\log_2(C_u)$, which is possibly larger than the size of the data-set under consideration.

18.5 Strict MML and its code-book

18.5.1 Strict Minimum Message Length

In SMML, we wish to form data into groups via a many-to-one mapping, m, from the data-set, X, to the estimator set[25]. We choose this mapping to minimise the expected length of a two-part message. The prior probability of an estimator (coding) block, $\hat{\theta}$, is the sum of the marginal probabilities, $r(x)$, of all the data, x, for which $m(x) = \hat{\theta}$. So,

$$\text{(Prior probability of } \hat{\theta}) \quad = \quad \text{p}(\hat{\theta}) \quad = \quad \sum_{\text{x:m(x)}=\hat{\theta}} \text{r(x)}$$

By Bayes's Theorem,

$$Pr(H|D) \quad = \quad (1/Pr(D)) \; . \; Pr(H\&D) \quad = \quad (1/Pr(D)) \; . \; Pr(H) \; Pr(D|H) \qquad (120)$$

and so the posterior probability, $Pr(H|D)$, of H given D, is proportional to $Pr(H)Pr(D|H)$. Appealing to Shannon's information theory, an event of probability p can be encoded (e.g. by a Huffman code) by a binary string of length $-\log_2 p$, ignoring issues of possible round-up to the next integer. (See (Wallace, 1968; Wallace, 1987), even (Wallace and Dowe, 1994), where this is appealed to in an MML context.)

18.5.2 The SMML code-book

The SMML machine, M_{SMML}, takes a computable probability distribution and designs a code-book which is optimal for that code-book in terms of minimising expected code-length. By Kraft's inequality (Kraft, 1949), the length of the code-words generated will be such that (within round-off error)
$2^{-\text{ (length of codeword for } \hat{\theta})} =$ prior probability of region encoded by $\hat{\theta}$.
The desirability of minimising the expected code-length can be thought of in either entropy terms or minimum expected Kullback-Leibler distance terms.
In talking of SMML code-books (Wallace, 1975; Wallace, 1987), the point made explicitly clear in this paper is that, in the Kolmogorov Complexity and UTM paradigm, we can[26] use a machine to be the code-book generator. We now describe how this can (in principle) be done.
Choose a machine, M, for the two-part joint encoding of $\hat{\theta}$ and $x|\hat{\theta}$ so as to minimise the expected length of this transmission. The reason for choosing our code-book so as to minimise the expected length is that if we have events, ev_i, of probability, p_i and code-word length, l_i, minimising $\sum_i p_i l_i$ results in $l_i \approx -\log_2 p_i$.
We call this machine M_{SMML}, since it behaves according to the Strict Minimum Message Length (SMML) criterion.
M_{SMML} takes as input an encoding of the prior, $h(.)$, and the likelihood, $f(.|.)$. In the event that either $h(.)$ or $f(.|.)$ is not total, then M_{SMML} might in turn not complete its calculations (and produce a code-book); and hence we insist that $h(.)$ and $f(.|.)$ be total.

[25] which will be a subset of the parameter space
[26] subject to some assumptions of total computability

Given its input $h(.)$ and $f(.|.)$, M_{SMML} then writes a code-book of minimum expected message length. Both sender and receiver have this machine, M_{SMML}, and they both also have the prior, $h(.)$, and the likelihood, $f(.|.)$. Armed with these, they can both generate the code-book.
M_{SMML} will (Wallace, 1975; Wallace, 1987) consider the marginal probabilities, $r(x)$, of all the (finite) possible data, x, and then search through the (possibly intractably many[27]) partitions of the data into groups so as to minimise the expected message length[28].
When the sender inspects the data, x, the sender can then send a two-part code for x from this code-book. The code which will be selected will be that $\hat{\theta}$ such that the two-part message transmitting $<\hat{\theta}><x|\hat{\theta}>$ will be of the minimum possible length. Armed with the code-book, the receiver will be able to decode $\hat{\theta}$, and, then (using $\hat{\theta}$) x.

18.6 Computational complexity of SMML in inference

(Farr and Wallace, 1997.)

18.7 "Efficient" Markets

Optimal inference entails finding the shortest program for some data. In general, by the halting problem, this is not decidable. Hence, it is silly to say that the markets are necessarily efficient.

Furthermore, optimal prediction is even harder than optimal inference, since it entails combining several models, not just the optimal. (Dowe and Korb, 1996).

SMML inference is difficult (Section 18.6), but combining predictions which use Universal Turing Machines is undecidable.

19 MML, inductive learning and the Turing Test

The Turing Test is a finite behavioural test. It can be argued from MML principles that inductive learning = compression. As such, the requirement of compression can be added to the test as a non-behavioural enhancement (Dowe and Hájek, 1997, 1998).

Question (Dowe and Hájek, 1997):

Given two programs H_1 and H_2 respectively of lengths l_1 and l_2, $l_1 < l_2$, if H_1 and H_2 perform equally well on a Turing Test (or if $Pr(\text{Data}|\text{H}_1) = \Pr(\text{Data}|\text{H}_2)$),
which, if either, should be predictively preferred for the future
for "right"/"wrong" prediction?

[27] It is the intractability of this search that leads us to say that the construction of an SMML code-book can be done *in principle*.

[28] These probabilities could be (slightly) rounded off, such as to a factor of, e.g., 2^{-30}

for probabilistic prediction?

Answer (probably):
Clearly, the theory which is more likely a priori will be more likely a posteriori. This will *almost certainly* be the better of the two theories as a predictor.

20 (Further) Applications

20.1 (Further) Snob Applications

(Papp, Dowe and Cox, 1993), LandSat data.
Kissane et al. (1996a, 1996b), Melbourne family grief study; Prior et al. (1998), autism data.

21 Discussion, Summary and Conclusion

If one wants to make sense of data, one can accept a pedestrian method will little question or one can take trouble to try and get things right. From all the available information available at the time of writing, it looks like the methods described here are the way of the future.

If having mathematical or other difficulty, be encouraged that the trouble is worth the effort. Difficult exercises are likely to be rewarded with due acknowledgment.

21.1 Quotable quotations

(about probabilistic prediction)

"*Human beings are perhaps never more frightening than when they are convinced beyond doubt that they are right.*" Laurens van der Post.
(from desk calendar, 2 October 1997.)

cf. a quotation similar to that above from Jacob Bronowski in *The ascent of man*.

"The Master said, Yu, shall I teach you what knowledge is? When you know a thing, to recognise that you know it, and when you do not know a thing, to recognise that you do not know it. That is knowledge."
Analects of Confucius (transl. by Arthur Waley), Book II, No. 17, circa 500B.C.

See also the Revision section.

22 Revision

Anyone using this material to lecture is recommended and encouraged to leave some time at the end of the course for revision.

Good luck.

23 Examples sprinkled throughout

23.1 What now?

Any crucial typos?
Revision.
My many slides from many MML talks.

23.1.1 What now?

If getting stuck at this point, make a cursory discussion of the material below and return to either earlier written glanced-over material or some of my published papers not yet mentioned.

Chess stuff, logistic problem with it

CSC423 LEARNING AND PREDICTION

Assignment 2

DUE: 12:00 noon, Friday 26 September 1997,
at the Computer Science General Office

This assignment is worth 15% of the total assessment for this subject.
Please read carefully the submission requirements on page 4.

Introduction

Somewhere on planet Earth, north of the tropic of Capricorn, there is a site measuring several hundred square kilometres with a significantly high concentration of a radioactive isotope[1]. Some would like to mine this site, others are concerned about the possibility of contaminating the local river system and one entrepreneurial tour operator suggests that a cool, refreshing drink of the tailings water will give an inner glow.

In a supposed attempt to bring some objectivity into the debate, it is desired to measure the concentration of this isotope. The site has a divide running through it, with sub-sites G and D to the left and right of the divide respectively. The concentration of the isotope is (initially) assumed to be uniform around G and uniform around D, but by no means necessarily the same for G and D.

An experiment is conducted as follows:
On Day 1, a number, N_G, of Geiger counters are placed at random sites around G for a duration of $t_G = 18$ hours, and counts $c_{G1}, \ldots, c_{GN_G}$ are observed on these Geiger counters.
The Geiger counters are believed to have behaved with little or no fault on Day 1 but, after their protracted exposure to extreme weather, it is believed that there might be some problems - such as the following - with some of these Geiger counters for future use:
• Some of the Geiger counters (type "IQ") could be "improperly quenched", meaning that[2] they are likely to return spurious readings roughly according to a geometric distribution (whose probability parameter is not known).
• Some of the Geiger counters (type "SO") switch off and fail to read if they have not registered a count in the last one hour period. Working out the probability of a count occurring in a one hour period, these can be assumed to return readings according to a geometric distribution (whose probability parameter can be related to the relevant Poisson rate, r).
• There is a suspicion that a small portion (type "F") of the results might have been fabricated[3] using a geometric distribution with the same mean as the Poisson distribution,

[1] Rather than phrase this problem in terms of radioactive decays, we could have alternatively phrased it in terms of counting arrivals of μ-mesons from cosmic showers.

[2] in short

[3] by cost-cutters presumed to be less than scrupulous

$\mathrm{Pn}(\mathrm{r_G})$, and it is also desired to carry out a routine check on this suspicion.

Since types "IQ", "SO" and "F" are variations of the geometric distribution, we shall assume for simplicity initially that only one of them (IQ) actually occurs.

At this point, miners and non-miners alike call for the "data miners".

Question 1 (1 mark)

Assuming a Poisson distribution $\mathrm{Pn}(\mathrm{r_G t_G})$ of radioactive decays measured for each of the N_G counters in the site G on Day 1, derive
1 (a) the mean of the distribution and
1 (b) the likelihood function for the observed counts $c_{G1}, \ldots, c_{GN_G}$.

1 (c) Derive the Maximum Likelihood estimate, $(\hat{r_G})_{ML}$, of r_G.

1 (d) From 1(b), derive the Fisher information for this distribution.

1 (e) Give a brief justification for your choice of prior, $h(r_G)$, on r_G.
Use this prior, 1(b) and 1(d) to derive the Minimum Message Length (MML) estimate, $(\hat{r_G})_{MML}$, of r_G, and state the value of the message length at its minimum.

Question 2 (3 marks)

Consider a geometric distribution, Geom(q), with probability parameter, q :
$f(x|q) = q^x(1-q)$, for non-negative integers, x.

Assuming n_x runs of length x, $x \geq 0$, and a total number of runs given by $N = \sum_x n_x$,

2 (a) derive the likelihood function and

2 (b) state the Maximum Likelihood estimate, $(\hat{q})_{ML}$, of q.

2 (c) Derive the Fisher information for this distribution.

2 (d) Choosing the uniform prior, $h(q) = 1$, or any other prior you wish to briefly justify, derive the Minimum Message Length (MML) estimate, $(\hat{q})_{MML}$, of q, and

2 (e) state the value of the message length at its minimum.

Leading in to Question 3, N_D of these counters are now placed around D on Day 2 for a duration of $t_D = 18$ hours. Counts $c_{D1}, \ldots, c_{DN_D}$ are observed.

Question 3 (5 marks)

Let p_1 be the proportion of correctly operating counters operating when sub-site D is inspected, and assume that the remaining $p_2 = p_{IQ} = 1 - p_1$ of the counters are improperly quenched. Assume further that the Poisson rate for correct counters is r_D and that IQ counters return a geometric distribution, Geom($\mathrm{q_D}$) with parameter, q_D.

Given the observed counts $c_{D1}, \ldots, c_{DN_D}$,

3 (a) carefully state the likelihood as a function of p_1, r_D and q_D.

3 (b) If data things were to be assigned totally (rather than partially) to classes, discuss in general terms the effect that this might have on the estimates of p_1, r_D and q_D.

3 (c) Stating a prior on the relevant parameters, state the length of a two-part message length which conveys the values of the parameters and then the data given these parameter values. Make explicit and clear any assumptions that might be made (such as, e.g., off-diagonal terms in the Fisher information matrix). Try to make as few assumptions as possible, and try to justify any that have to be made.

Leading in to Question 4, we now wish to collect some data and test some hypotheses, one comparing the G and D isotope concentrations and the other concerning the Geiger counters used at D :
Measuring r_G and r_D in decays per hour, the two hypotheses that we would like to test (as in Question 4 below) are
(a) whether $r_G = r_D$, and (b) whether $q_D = 1 - e^{-r_D}$.

Question 4 (6 marks)

Do *either* Question 4 (a) *or* Question 4 (b), but *not both.*

Question 4 (a) (If attempting this, do not do 4(b).)

Generate data from N_G.Pn($\mathrm{r_G t_G}$) and from N_D.(0.5 Pn($\mathrm{r_D t_D}$) + 0.5 Geom($\mathrm{q_D}$)) with
$N_G = 2N_D$, $q_D = 0.7$, $r_D = 1.25$ and $r_G = 3.0$;
and N_D varying through the range 3, 10, 20, 100, 1000.
Also generate such data with $N_G = N_D$, $q_D = 0.7$, $r_D = 1.8$ and $r_G = 2.4$;
and N_D varying through the range 3, 10, 20, 100, 1000.
(See note on page 4 about random number generation.)

For each of these cases, from the test data generated,
4 (a) test whether $r_G = r_D$.

Hint (for Question 4 (a)):
For 4 (a), compare the cost of sending two two-part messages (one involving r_G but not r_D and the other involving r_D but not r_G) with the cost of sending one two-part message (somehow involving both rates).

Question 4 (b) (If attempting this, do not do 4(a).)

Generate data from N_D.(0.5 $\mathrm{Pn}(\mathrm{r_D t_D})$ + 0.5 $\mathrm{Geom}(\mathrm{q_D})$) with
$N_G = 2N_D$, $q_D = 0.7$ and $r_D = 1.25$;
and N_D varying through the range 3, 10, 100, 1000, 2000.
Also generate such data with $N_G = N_D$, $q_D = 0.7$, $r_D = 1.8$;
and N_D varying through the range 3, 10, 100, 1000, 2000.
(See note on page 4 about random number generation.)

For each of these cases, from the test data generated,
4 (b) test whether $q_D = 1 - e^{-r_D}$.

Hint (for Question 4 (b)):
For 4 (b), in stating the two-part message to convey the data at sub-site D, try and cost the message both (i) not using the hypothesised functional dependence between q_D and $1 - e^{-r_D}$ and; (ii) also, as accurately as possible, assuming the dependence.

Note (about random number generation) :
For the duration of the assignment, software will be available to generate multinomial, Poisson and other distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ .

Submission requirements
Create the requested data-sets of various sizes as in Question 4, using or not using (as you prefer) the available software for pseudo-random number generation.
Submit any source code written along with your assignment solutions and answers.
Make your data-sets readable from the time of submission until the assignment is returned, and include the path and file names of the data-sets in your printed submission.

CSC423 LEARNING AND PREDICTION

Assignment 2, 2nd Semester, 1998 - *and* **worth 30%**.

DUE: 12:00 noon, Friday 23 October 1998,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 30% of the total assessment for this subject.
Please read *carefully* the submission requirements on page 3.

Introduction

Machine learning and "data mining" have tended to be more interested in boolean and discrete objects like decision trees, whereas statistics has traditionally tended to focus more on continuous distributions, such as the Gaussian (or Normal) distribution. Decision tree programs (Quinlan, 1986; Quinlan and Rivest, 1989; Wallace and Patrick, 1993; etc.) have usually had binomial or multinomial distributions in the leaves. In the 1980s and 1990s, we have seen such programs as CART (Classification And Regression Trees), MARS and[1] J. R. Quinlan's M5, which endeavour to do Gaussian or linear regressions in the leaf nodes.
In this assignment, we try our own hand at using Minimum Message Length (MML) to grow decision trees with Gaussian distributions in the leaf nodes.

About the questions

Let x_1, x_2, x_3, x_4, x_5 and x_6 be binary boolean attributes, and
let x_7 be a ternary (3-valued) attribute.
Let x_8 and x_9 be Gaussian attributes.

Assume that (pseudo-)random number generator generates each binary attribute as being equally likely to be 0 (false) or 1 (true).
Similarly assume that (pseudo-)random number generator generates each ternary attribute as being equally likely to be 0, 1 or 2.

[1] For what it's worth, J.H. Friedman, who is involved with at least one of CART or MARS, is expected to be visiting Australia and Melbourne in November 1998.

Question 1 (worth 19 marks)

Using the (pseudo-)random number generators, generate data of sample size N for varying values of N : $N = 10, 100, 300, 1000, 3000$.

Include the binary attributes x_1, x_2, x_3 and x_4, and the Gaussian attribute x_8.

The tree will have four leaves :
x_1;
not(x_1) and x_2;
not(x_1) and not(x_2) and x_3;
not(x_1) and not(x_2) and not(x_3).

In each the four leaves, there will be one Gaussian distribution, $N(\mu, \sigma^2)$. This will be
$x_1 : x_8 \sim N(-5, 15^2)$
not(x_1) and $x_2 : x_8 \sim N(+5, 15^2)$
not(x_1) and not(x_2) and $x_3 : x_8 \sim N(-5, 5^2)$
not(x_1) and not(x_2) and not(x_3) : $x_8 \sim N(+5, 5^2)$.

For the 5 given values of N above, now run the experiment of (pseudo-)randomly generating a data-set of size N with the distribution specifications above.
Repeat this data generation process two (2) times, so that for each value of N, you have two (2) data-sets.

Use MML inference with your appropriate choice of priors on μ and σ to infer the MML decision trees and the appropriate leaf regressions for each of your $2 \times 5 = 10$ data-sets.

Question 2 (worth 6 marks)

For the two trees and leaf distributions you inferred when $N = 300$, use the Kullback-Leibler distance to estimate the distance from the true distribution (as specified in Question 1 and pseudo-randomly generated by you) to the distribution inferred by your tree structure.

Question 3 (worth 5 marks)

Modify Question 1 to include all of the variables x_1, x_2, x_3, x_4 and x_8 from Question 1 and also at least either the ternary attribute x_7 or the second Gaussian attribute x_9.
If using x_9, make sure that both x_8 and x_9 occur in leaf nodes.
If using x_3, expand the tree so that the former leaf node x_1 now becomes three leaf nodes: x_1 and $(x_3 = 0)$; x_1 and $(x_3 = 1)$ and x_1 and $(x_3 = 2)$.

Generate $2 \times 5 = 10$ data-sets with $N = 10, 100, 300, 1000, 3000$ and some choice of Gaussian distributions in the leaves.

Use your MML inference program from Question 1 to try and infer what the structure of the tree was. Only a short comment need be made here about goodness of fit.

Note (about random number generation) :
For the duration of the assignment, software will be available to generate multinomial, Gaussian, Poisson and other distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
although you should feel free to use any decent (pseudo-)random number generator that you like.

Submission requirements - *please read carefully*
Submit any source code written along with your assignment solutions and answers.
Make your data-sets readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

CSC423 LEARNING AND PREDICTION

Assignment 2, 2nd Semester, 1999 - *and* **worth 30%**.

DUE: 12:00 noon midday on Friday 8th October 1999,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 30% of the total assessment for this subject.
Please read *carefully* the submission requirements on page 4.

Introduction

Machine learning and "data mining" have tended to be more interested in boolean and discrete objects like decision trees, whereas statistics has traditionally tended to focus more on continuous distributions, such as the Gaussian (or Normal) distribution.

Some linguists and some others interested in grammar or syntax will sometimes be interested in inferring a (probabilistic) grammar from some data. In the style of Dr Doolittle, the grammar being inferred could be that of animal communication. One model of a (probabilistic) grammar is a probabilistic finite state automaton, or PFSA.
In this assignment, we try our own hand at using Minimum Message Length (MML) to cost PFSAs. We then use various search heuristics to search a rather large search space, and try to find the MML PFSA. The detailed assignment will follow below.

About the questions

Consider the following two Probabilistic Finite State Automata (PFSAs):

‘	*a*	*b*	*c*	*d*	(PFSA I)
1	3 (0.60)	2 (0.40)	0 (0.00)	0 (0.00)	
2	3 (0.45)	3 (0.25)	1 (0.30)	0 (0.00)	
3	4 (0.55)	5 (0.45)	0 (0.00)	0 (0.00)	
4	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	
5	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	

and

‘	*a*	*b*	*c*	*d*	(PFSA II)
1	3 (0.52)	4 (0.48)	0 (0.00)	0 (0.00)	
2	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	
3	5 (0.59)	2 (0.41)	0 (0.00)	0 (0.00)	
4	3 (0.35)	3 (0.30)	1 (0.349)	4 (0.001)	
5	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	

These machines, PFSA-I and PFSA-II, correspond to the true data generation processes of two speakers (I and II), be they human, other animal, mechanical, extra-terrestrial or etc. It is probably worthwhile drawing both these PFSAs.

The transition table entry in row j and column s is the (output) state that the machine will go to if it sees the symbol s when in (input) state j. The number in brackets is the probability of the machine seeing the symbol s when it is in state j. If it is impossible (probability 0) for the machine to see a certain symbol from a certain state, then we denote the output state as being 0.

In both Questions 1 and 2, when we generate the data, we use the two PFSAs PFSA-I and PFSA-II, *but* when we come to do inference, we do not know what the true PFSAs are.

Question 1 (worth 24 marks)

Using some (pseudo-)random number generator (see the note at the end of the assignment), generate data of sample size N for varying values of N : $N = 40, 100, 250, 500, 1000$. The data-set(s) should be generated as follows.

Your data-set should have two attributes, the first of which is an index (or time) value, i, ranging from 1 to N. Select a (pseudo-)random number, r_1, which is an equally likely, 50%-50%, choice out of the two numbers 1 and 2. Select a second (pseudo-)random number, r_2, from a distribution uniform between 1 and N.

If $r_1 = 1$, then speaker 1 speaks first using PFSA-I, followed by speaker 2 using PFSA-II. Otherwise, if $r_1 = 2$, then speaker 2 speaks first using PFSA-II, followed by speaker 1 using PFSA-I. So, r_1 tells us who speaks first.

And r_2 tells us how long the first speaker speaks for, speaking from time from 1 up to and including r_2, and then the second speaker speaks from time $r_2 + 1$ up to and including N.

You should record the values of r_1 and r_2, and then (pseudo-)randomly generate symbol data ('a', 'b', 'c', 'd') from PFSA-I and PFSA-II for each time point ranging from 1 to N. Each PFSA will initially commence in State 1 and then generate data according to its probabilistic grammar, which is described on the previous page before the question. Between times r_2 and $r_2 + 1$, the speakers will change. The data generated by the PFSAs is the second attribute in the data-set.

For the 5 given values of N above, now run the experiment of (pseudo-)randomly generating a data-set of size N with the distribution specifications above.
Repeat this data generation process two (2) times, so that for each of your five (5) values

of N, you have two (2) data-sets; thus making a total of $2 \times 5 = 10$ data-sets.

Use MML inference with your appropriate choice of priors and coding schemes for the PFSAs to infer whether it was most likely that there was one speaker ($r_2 = N$) or two speakers ($1 \leq r_2 < N$). If one speaker, use MML to infer the structure of the PFSA. If two speakers, then use MML to infer the value of r_1 (who spoke first), the value of r_2 (how long this person spoke for) *and* the structures and details of the two inferred PFSAs.

Do this for each of your $2 \times 5 = 10$ data-sets.

Question 2 involves a model which is a generalisation of that in Question 1.

Question 2 (worth 6 marks)

This time the data-set has one additional attribute. Again, the first attribute is an index value, i, ranging from 1 to N. The second attribute, T, is temperature, since we have reason to believe that at some crucial, threshold, temperature, it might be possible that there will be toggle changes from PFSA-I to PFSA-II[1]. The second attribute, the temperature, T, will be distributed (randomly and) uniformly between 10.00 and 30.00 degrees Celsius.

Select a (pseudo-)random number, r_0, which is an equally likely, 50%-50%, choice out of the two numbers 1 and 2.

If $r_0 = 1$, then the data-sets will be generated as in Question 1, except for the fact that the second attribute will be the temperature, T. The value of the symbol ('a', 'b', 'c' or 'd') will be the third attribute and will be generated as in Question 1, being independent of the temperature attribute, and depending only upon the values of r_1 and r_2.

However, if $r_0 = 2$, then select r_1 randomly and uniformly from the range such that $10.00 \leq r_1 \leq 30.00$. Select a (pseudo-)random number, r_2, which is an equally likely, 50%-50%, choice out of the two numbers 1 and 2.

If $r_2 = 1$, then the speaker uses PFSA-I for $10.00 \leq T \leq r_1$ and uses PFSA-II for $r_1 < T \leq 30.00$. Otherwise, if $r_2 = 2$, then the speaker uses PFSA-II for $10.00 \leq T \leq r_1$ and uses PFSA-I for $r_1 < T \leq 30.00$. It can be assumed that each PFSA will initially commence in State 1 and it can also be assumed[2] that, when the speaker toggles to, from

[1] Perhaps there are two speakers who play tag-team every time the temperature passes through this threshold, or perhaps there is one speaker who changes his/her style of speaking depending upon whether it is hot or cold.

[2] if you like and you also state this.

and between PFSA-I and PFSA-II, that in returning to PFSA-j from PFSA-$(3-j)$, it is as though the intermittent time spent in PFSA-$(3-j)$ and the symbols generated there are (temporarily) forgotten.

You should record the values of r_0, r_1 and r_2. Generate the second attribute, the temperature, T, (randomly and) uniformly between 10.00 and 30.00. Depending upon the value of r_0, then (pseudo-)randomly generate symbol data ('a', 'b', 'c', 'd') accordingly as the third attribute for each time point ranging from 1 to N.

For the 5 given values of N above, now run the experiment of (pseudo-)randomly generating a data-set of size N with the distribution specifications above.
Repeat this data generation process two (2) times, so that for each of your five (5) values of N, you have two (2) data-sets; thus making a total of $2 \times 5 = 10$ data-sets.

Use MML inference with your appropriate choice of priors and coding schemes for the PFSAs to infer whether it was most likely that there was:
one speaker ($r_0 = 1$, $r_2 = N$),
two speakers with a single change-over ($r_0 = 1$, $1 \leq r_2 < N$)
or toggling speech patterns ($r_0 = 2$) toggling at temperature r_1.
Also use MML to infer the structure(s) and details of the PFSA(s).

Do this for each of your $2 \times 5 = 10$ data-sets.

Note (about random number generation) :
For the duration of the assignment, software will be available to generate multinomial, Gaussian, Poisson and other distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ , although you should feel free to use any decent (pseudo-)random number generator that you like.

Submission requirements - *please read carefully*
Submit any source code written along with your assignment solutions and answers.
Make your data-sets readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

Very crude outline of marking scheme
Generating the data-sets should be relatively easy. Costing the message length will be far more important than the search heuristics. You might be called upon to demonstrate your working code. Possible bonus marks for any sensible comments about Kullback-Leibler distance(s) (from true to inferred).

(End of Assignment.)

CSC423 LEARNING AND PREDICTION

Assignment 2, 2nd Semester, 2000 - *and* **worth 30%**.

DUE: 12:00 noon, Wednesday 11 October 2000,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 30% of the total assessment for this subject.
Please read *carefully* the submission requirements on page 3.

Introduction

Machine learning and "data mining" have tended to be more interested in boolean and discrete objects like decision trees, whereas statistics has traditionally tended to focus more on continuous distributions, such as the Gaussian (or Normal) distribution. Decision tree programs (Quinlan, 1986; Quinlan and Rivest, 1989; Wallace and Patrick, 1993; etc.) have usually had binomial or multinomial distributions in the leaves. In the 1980s and 1990s, we have seen such programs as CART (Classification And Regression Trees), MARS and J. R. Quinlan's M5, which endeavour to do Gaussian or linear regressions in the leaf nodes.

In this assignment, we try our own hand at using Minimum Message Length (MML) to grow decision trees with Gaussian distributions in the leaf nodes.

About the questions

Let x_1, x_2, x_3 and x_4 be binary boolean attributes, and let x_5 be a Gaussian attribute.

Assume that (pseudo-)random number generator generates each binary attribute as being equally likely to be 0 (false) or 1 (true).

Question 1 (worth 25 marks)

Using a (pseudo-)random number generator (see page 3)[1], generate data of sample size N for varying values of N : $N = 40, 200, 500, 1000, 2000$.

Include the binary attributes x_1, x_2, x_3 and x_4, and the Gaussian attribute x_5.

The tree describing the data will have four leaves :
x_1;
not(x_1) and x_2;
not(x_1) and not(x_2) and x_3;
not(x_1) and not(x_2) and not(x_3).

In each of the four leaves, there will be one Gaussian distribution, $N(\mu, \sigma^2)$. This will be
$x_1 : x_5 \sim N(-6, 15^2)$
not(x_1) and $x_2 : x_5 \sim N(+4, 15^2)$
not(x_1) and not(x_2) and $x_3 : x_5 \sim N(+8, 5^2)$
not(x_1) and not(x_2) and not(x_3) : $x_5 \sim N(+16, 5^2)$.

For the 5 given values of N above, now run the experiment of (pseudo-)randomly generating a data-set of size N with the distribution specifications above.
Repeat this data generation process two (2) times, so that for each value of N, you have two (2) data-sets.

Use MML inference with your appropriate choice of priors on μ and σ to infer the MML decision trees and the appropriate leaf regressions for each of your $2 \times 5 = 10$ data-sets.

[1] see page 3

Question 2 (worth 5 marks)

For the two trees and leaf distributions that you inferred from the data in Question 1 when $N = 200$, use the Kullback-Leibler distance to estimate the distance from the true distribution (as specified in Question 1 and pseudo-randomly generated by you) to the distribution inferred by your tree structure.

Note (about random number generation for Question 1) :
For the duration of the assignment, software will be available to generate multinomial, Gaussian, Poisson and other distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
http://www.csse.monash.edu.au/~dld/Hons/2000/dldprojects (under "(13)")
and http://random.mat.sbg.ac.at/links/rando.html ,
although you should feel free to use any decent (pseudo-)random number generator that you like.

Comment about **search heuristics** for Question 1 :
We will discuss search heuristics, ideally in the lecture slot.

Submission requirements - *please read carefully*
Your program should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java.

Submit any source code written (both in hard copy and in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission. The soft copy of your source code should be sent with Subject line: "CSC423 Assignment 2" to dld@cs.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of Assignment 2.

CSE423 LEARNING AND PREDICTION

Assignment 3, 1st Semester, 2001 - *and* **worth 10%**.

DUE: 12:00 noon, ?day ? ? 2001 (to be decided in class - Mon. 7th May 2001), **at the Computer Science and Software Eng. (Clayton) General Office**

This assignment is worth 10% of the total assessment for this subject.
Please read *carefully* the submission requirements on page 3.

Introduction

This assignment asks student to use the Snob program (C. S. Wallace and D. L. Dowe, MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions, *Statistics and Computing*, Vol. 10, No. 1, Jan. 2000, pp73-83) to analyse dog-bite data, DNA micro-array data and some "*secret*" but not unfriendly data.

Snob software

The Snob software is available from
http://www.csse.monash.edu.au/~dld/Snob.html and also from
http://www.csse.monash.edu.au/research/mdmc/software .
It is also installed on the machines in the CSSE Clayton Bldg. 26 Hons lab, and on the CSSE indy's.

It is capable of analysing at least the statistical distributions described in the title of the abovementioned (Wallace and Dowe, 2000) paper.

Data: sd1.raw, dog-bites and DNA micro-arrays

The sd1.raw data is available from http://www.csse.monash.edu.au/~dld/Snob.html ,

the dog-bite data is available from
http://www.csse.monash.edu.au/~lloyd/tilde/CSC4/CSC423/Local/dog

and the DNA micro-array data is available from
http://www.csse.monash.edu.au/~lloyd/tilde/CSC4/CSC423/Local/Micro-Arrays/ .

Question 1 (worth 2 marks)

Analyse the data-set sd1.raw . State the number of components (classes), relative abundances (mixing proportions) and component distributional parameters for the theory you find with the shortest message length.

State this shortest message length in both nits and bits.

Question 2 (worth 4 marks)

Use Snob to analyse the dog-bite data.

A lunar month is approximately 29 days 12 hours 44 minutes, or 29.53 days.

Hint: The Poisson distribution *could* be used to model (counts and) rates. Regarding phases of the moon as angles, the von Mises distribution *could* be used to model these.

Does your analysis suggest anything at all about whether or not the phase of the moon is relevant to the probability of dog-bite occurrence?

If so, what does it suggest?

Question 3 (worth 4 marks)

Use Snob to analyse the DNA micro-array data.

Read the paper and the README file.

You should experiment with some scaling and data transformations because
(i) Snob may not accept the raw data,
(ii) there are correlations between the experiments/attributes,
(iii) low levels have been set to zero, and
(iv) biologists are interested in the *ratios* of expression levels.

What do you find?

Note (about random number generation) :

For the duration of the assignment, software - should you need it (you might need it for Ass't 2, but you might well not need it for this assignment, Ass't 3) - will be available to generate multinomial, Gaussian, Poisson and von Mises distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
http://www.csse.monash.edu.au/~dld/datalinks.html ,
http://www.csse.monash.edu.au/~dld/Hons/2001/dldprojects (under "(16)")
and http://random.mat.sbg.ac.at/links/rando.html ,
although you should feel free to use any decent (pseudo-)random number generator that you like.

Comment about **search heuristics** using Snob :

See snob.doc and sample files typically called *.control for guidelines on search heuristics. We could discuss search heuristics further in the lecture slot.

Submission requirements - *please read carefully*

Any programs should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java.

Submit any source code written (both in hard copy and in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission and be submitted as on page 1 of this assignment. The soft copy of your source code should be sent with Subject line: "CSE423 Assignment 3" to dld@cs.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of Assignment 3, 2001.

CSE423 LEARNING AND PREDICTION

Assignment 4, 1st Semester, 2001 - *and* **worth 20%.**

DUE: 12:00 noon, Monday 28th May 2001,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 20% of the total assessment for this subject.
Please read *carefully* the submission requirements above and on page 4.

Introduction

Students have previously seen an MML analysis of the Poisson distribution and the von Mises circular distribution, and have also (in Assignment 3) used the Snob program, which contains these distributions. Students have also seen an MML analysis of decision trees with multinomial distributions in the leaves.

This assignment asks students to develop a decision tree program with Poisson and von Mises regressions in the leaves. Students are then asked to test their program(s).

The assignment then asks students to re-analyse the dog-bite data from CSE423 Semester 1, 2001, Assignment 3.

DTreeProg software

The Wallace and Patrick (Coding Decision Trees, *Machine Learning Journal*, Vol. 11, pp7-22, 1993) DTreeProg software (and documentation, dtree.doc) is available from http://www.csse.monash.edu.au/research/mdmc/software .

Snob software

The Snob software is available as in CSE423 Semester 1, 2001, Assignment 3.

Data: dog-bites

As in CSE423 Semester 1, 2001, Assignment 3, the dog-bite data is available from http://www.csse.monash.edu.au/~lloyd/tilde/CSC4/CSC423/Local/dog .

Question 1 (worth 2 + 4 + 4 = 10 marks)

Write a decision tree program that will cost the length of a two-part message of a decision tree program with a multinomial leaf attribute. (2 marks)

Extend your decision tree program so that it does two of the following three options: 1A, 1B and 1C (worth 4 marks each). If one of your choices is option 1C, then your program should able to do simultaneously both 1C and your other choice.

1A: Modify your decision tree program so that it can have, in each leaf, a univariate Poisson distribution.

1B: Modify your decision tree program so that it can have, in each leaf, a von Mises circular distribution.

For option 1C, recall discussions on continuous-valued cut-points in decision trees.

1C: Modify your decision tree program so that it can have internal split nodes on angular-valued cut-points.

Question 2 (worth 5 marks)

Test your program from Question 1 by first specifying two decision tree functions that you think are appropriate. Make sure that your two decision tree test functions include the two options that you chose in Question 1.

Draw in machine-readable form or in clear and legible hand-writing your two decision tree functions. Clearly describe:
• the arity of all multinomial attributes,
• which attributes are split on (and, for continous- or angular-valued attributes, what the cut-point is),
• the type of distribution (multinomial, Poisson or von Mises) in each leaf node, and
• the distributional parameters (probability for multinomial, rate for Poisson, μ and κ for von Mises) in each leaf node.
Put as much of this description in machine-readable form as possible.

For each of the two decision functions in turn, (pseudo-)randomly generate five data-sets of size $N = 50,\ 100,\ 250,\ 500$ and 1000 respectively.

For each such data-set, draw and describe the inferred decision function, and describe how different or similar it seems to the underlying function used to generate the data.

Question 3 (worth 4 marks)

Recall your analysis of the dog-bite data from CSE423 Semester 1, 2001, Assignment 3, Question 2.

Now, use your program from Question 1 above to re-analyse the dog-bite data.

Does your new analysis suggest anything at all about whether or not the phase of the moon is relevant to the probability of dog-bite occurrence?

If so, what does it suggest?

Question 4 (worth 1 mark)

Include your earlier analysis of the dog-bite data from CSE423 Semester 1, 2001, Assignment 3, Question 2. This should be identical to what you submitted then.

Compare and contrast your analysis from Question 3 immediately above with your earlier submitted analysis from CSE423 Semester 1, 2001, Assignment 3, Question 2.

Which analysis do you prefer, if either, and why?

Note (about random number generation) :

For the duration of the assignment, software - should you need it - will be available to generate multinomial, Gaussian, Poisson and von Mises distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
http://www.csse.monash.edu.au/~dld/datalinks.html ,
http://www.csse.monash.edu.au/~dld/Hons/2001/dldprojects (under "(16)"),
http://random.mat.sbg.ac.at/links/rando.html ,
and http://www.csse.monash.edu.au/research/mdmc/software/random/index.shtml ,
although you should feel free to use any decent (pseudo-)random number generator that you like.

A comment about **search heuristics** for decision trees :

See
• the section regarding *Lookahead* in the Wallace and Patrick (1993) paper,
• from dtree.doc (which comes with the DTreeProg software), section "Running the program", sub-section "<0-9>", or
• material on decision trees distributed in lectures (or recall any discussions that may have taken place in the lecture slot).

Submission requirements - *please read carefully*

Any programs should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java.

Submit any source code written (both in hard copy and in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission and be submitted as on page 1 of this assignment. The soft copy of your source code should be sent with Subject line: "CSE423 Assignment 4" to dld@cs.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets (such as those in Question 2) readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of Assignment 4, 2001.

CSE455 LEARNING AND PREDICTION II: MML "Data Mining"

Assignment 1, 1st Semester, 2002 - *and* **worth 20%.**

DUE: 12:00 noon, Wednesday 22nd May 2002,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 20% of the total assessment for this subject.
Please read *carefully* the submission requirements above and on page 5.
Total marks: $2+3+2+2+0+3+4+4 \ = \ 20$.

Introduction

Recall the Snob program (C. S. Wallace and D. L. Dowe, MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions, *Statistics and Computing*, Vol. 10, No. 1, Jan. 2000, pp73-83 ; http://www.csse.monash.edu.au/∼dld/Snob.html), which was used to analyse dog-bite data in CSE454 Prac' 1 2002
(http://www.csse.monash.edu.au/∼lloyd/tilde/CSC4/CSE454/Local/2002/prac1.html).
Snob uses the von Mises circular distribution, $M_2(\mu, \kappa)$, to analyse angular data. Another possible model for angular data is the wrapped Normal distribution, $WN(\mu, \sigma^2)$. Of course, there are other distributions for angular data.

Snob software
The Snob software is available from
http://www.csse.monash.edu.au/∼dld/Snob.html and also from
http://www.csse.monash.edu.au/research/mdmc/software .
It should be installed on the machines in the CSSE Clayton Bldg. 26 Hons lab, and on the CSSE indy's. It is capable of analysing at least the statistical distributions described in the title of the abovementioned (Wallace and Dowe, 2000) paper.

von Mises circular distribution

The 2-dimensional von Mises density, $M_2(\mu, \kappa)$ or $VM(\mu, \kappa)$, is an analogue of the Gaussian density for angles in the plane.

Let $I_0(\kappa) \ = \ \frac{1}{2\pi} \int_0^{2\pi} e^{\kappa \cos(\theta)} \ d\theta \ = \ \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r}}{(r!)^2}$ and for $p > 0$,
let $I_p(\kappa) \ = \ I_0(\kappa) \times E(\cos(p\theta)) \ = \ I_0(\kappa) \times \frac{1}{2\pi} \int_0^{2\pi} \cos(p\theta) e^{\kappa \cos(\theta)} \ d\theta \ = \ \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r+p}}{(p+r)! \ r!}$.

So, $I_1(\kappa) \ = \ I_0(\kappa) \times E\left(\cos(\theta)\right) \ = \ \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r+1}}{r! \ (r+1)!} \ = \ \frac{d \ I_0(\kappa)}{d\kappa}$.

The density of the angular variate θ is given by $f(\theta) \ = \ 1/(2\pi I_0(\kappa)).e^{\kappa \cos(\theta - \mu)}$, where $I_0(\kappa)$ is a normalisation constant. The functional form of the likelihood is

$f(\theta|\mu,\kappa) \;=\; \frac{1}{2\pi I_0(\kappa)}\; e^{\kappa\cos(x-\mu)}$, and is sometimes written $\theta \sim M_2(\mu,\kappa)$.

Wrapped Normal circular distribution

The wrapped Normal is interesting, but not directly relevant to this assignment.

Re-cap on the Normal distribution
Recall that the functional form of the Gaussian - or Normal - distribution is
$f(x|\mu,\sigma) \;=\; \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((x-\mu)^2)}$, and this is sometimes written $X \;\sim\; N(\mu,\sigma^2)$.

Wrapped Normal distribution
Swapping from x to θ, for a wrapped Normal distribution,
$f(\theta|\mu,\sigma) \;=\; \sum_{j=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((\theta+2j\pi-\mu)^2)} \;=\; \frac{1}{\sqrt{2\pi}\sigma} \sum_{j=-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}((\theta+2j\pi-\mu)^2)}$,
and this is sometimes written $\theta \;\sim\; WN(\mu,\sigma^2)$.

For several pieces of data $\theta_1, ..., \theta_i, ..., \theta_N$,

$$\begin{aligned} L \quad &= \quad -\log f(\theta|\mu,\sigma) \;=\; \sum_{i=1}^{N} -\log\Big(\sum_{j=-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)}\Big) \\ &= \quad N\log(\sqrt{2\pi}) + N\log\sigma - \sum_{i=1}^{N} \log\Big(\sum_{j=-\infty}^{+\infty} e^{-\frac{1}{2\sigma^2}((\theta_i+2j\pi-\mu)^2)}\Big). \end{aligned}$$

The wrapped Normal is interesting, but not directly relevant to this assignment.

sin^2 and cos^2 circular distributions

Let $f(\theta|\mu) \;=\; \frac{1}{\pi}\; \cos^2(\frac{n(\theta-\mu)}{2}) \;=\; \frac{1}{\pi}\; \cos^2(\frac{n}{2}(\theta-\mu))$ for some positive integer, n.

Notice that for a von Mises distribution, for a wrapped Normal distribution and for the distribution above, adding to or subtracting from θ an amount of 2π or any integer multiple of 2π does not change the value of the likelihood function or any of its derivatives.

$$L \quad = \quad -\log \pi_{i=1}^{N} f(\theta_i|\mu) \;=\; N\log(\pi) - \sum_{i=1}^{N} 2\log\cos(\frac{n}{2}(\theta-\mu))$$

$$\frac{\partial L}{\partial \mu} \quad = \quad -\sum_{i=1}^{N} \frac{2\;\frac{n}{2}\sin(\frac{n}{2}(\theta-\mu))}{\cos(\frac{n}{2}(\theta_i-\mu))} \;=\; n\sum_{i=1}^{N}\tan(\frac{n}{2}(\theta_i-\mu)) \;=\; n\sum_{i=1}^{N}\tan(\frac{n}{2}(\mu-\theta_i))$$

$$\frac{\partial^2 L}{\partial \mu^2} \quad = \quad n\sum_{i=1}^{N}\frac{n}{2}\sec^2(\frac{\mathrm{n}}{2}(\mu-\theta_{\mathrm{i}})) \;=\; \frac{\mathrm{n}^2}{2}\sum_{\mathrm{i}=1}^{\mathrm{N}}\sec^2(\frac{\mathrm{n}}{2}(\mu-\theta_{\mathrm{i}})) \;=\; \frac{\mathrm{n}^2}{2}\sum_{\mathrm{i}=1}^{\mathrm{N}}\frac{1}{\cos^2(\frac{\mathrm{n}}{2}(\mu-\theta_{\mathrm{i}}))}$$

$$F \quad = \quad E(\frac{\partial^2 L}{\partial \mu^2}) \; = \; \frac{Nn^2}{2} \; E(\frac{1}{\cos^2(\frac{n}{2}(\mu - \theta_i))}) \; = \; \frac{Nn^2}{2} \times \frac{1}{\pi} \times 2\pi \; = \; n^2 N.$$

I will be amongst the first to admit that there maybe a missing factor of 2 or $\frac{1}{2}$ or a floating minus sign, $-$, or some such in the above. So, please feel highly invited to check the above mathematics and correct any possible such mistake.

Question 1 (worth 2 marks)
For this cos^2 model, obtain a (possibly implicit) formula for the maximum likelihood estimator of μ, $\hat{\mu}_{ML}$, given data $\vec{\theta} = \{\theta_1, \ldots, \theta_i, \ldots \theta_N\}$. Do this for general n.

Question 2 (worth 3 marks)
Assuming a uniform prior on μ, use the above to derive a message length for the sin^2 model and given data $\vec{\theta} = \{\theta_1, \ldots, \theta_i, \ldots \theta_N\}$. Do this for general n.

Question 3 (worth 2 marks)
Minimise this message length expression to obtain the minimum message length (MML) estimator, $\hat{\mu}_{MML}$, of μ.

Question 4 (worth 2 marks)
What can you say, if anything, about the relationship between the maximum likelihood estimator, $\hat{\mu}_{ML}$ and the minimum message length (MML) estimator, $\hat{\mu}_{MML}$?

Your name and student id, etc. should be attached to the green sheet at the front of your assignment. The second-last digit of your student id will be in the range from 0 to 9. Add 2 to it so that it is now in the range from 2 to 11. Let this be your own personal n, n_{me}.

Question 5 (worth 0 marks)
What is your value of n_{me}?

Write a program which, given input n and data $\vec{\theta} = \{\theta_1, \ldots, \theta_i, \ldots \theta_N\}$ will be able to give a message length for a specified $\hat{\mu}$ and which will be able to infer the MML estimator and give its message length - both in bits and in nits.

For both $n = 1$ and $n = n_{me}$, generate one data-set each of size $N = 50$ with $\mu = 0$.
(See guide later in the assignment about random number generation.)

Question 6 (worth 1+2=3 marks)
For the data set generated with $n = 1$ and $N = 50$,
Question 6a what is the message length and the MML estimate assuming $n = 1$?
Question 6b what is the message length and the MML estimate assuming $n = n_{me}$?

Question 7 (worth 2+2=4 marks)
For the data set generated with $n = n_{me}$ and $N = 50$,
Question 7a what is the message length and the MML estimate assuming $n = 1$?
Question 7b what is the message length and the MML estimate assuming $n = n_{me}$?

Question 8 (worth 4 marks)
For the **dog bite** data set from CSE454 Prac' 1 at
http://www.csse.monash.edu.au/~lloyd/tilde/CSC4/CSE454/Local/dog , write a short (absolute maximum 2 page) report on your analysis of just the *angles* from the dog-bite data.

Hint for **Question 8**: How much would it cost to encode the phase of the moon data using the *uniform* distribution around the circle?

Data: Dog bites

A lunar month is approximately 29 days 12 hours 44 minutes, or 29.53 days. The phases of the moon can be regarded as angles.

Note (about random number generation) :

For the duration of the assignment, software - should you need it - will be available to generate multinomial, Gaussian, Poisson and von Mises distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
http://www.csse.monash.edu.au/~dld/datalinks.html ,
http://www.csse.monash.edu.au/~dld/Hons/2001/dldprojects (under "(16)"),
http://random.mat.sbg.ac.at/links/rando.html ,
and http://www.csse.monash.edu.au/research/mdmc/software/random/index.shtml ,
although you should feel free to use any decent (pseudo-)random number generator that you like.

Submission requirements - *please read carefully*

Any programs should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java.

Submit any source code written (both in hard copy and in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission and be submitted as on page 1 of this assignment. The soft copy of your source code should be sent as plain ASCII text with Subject line: "CSE455 Assignment 1" to dld@cs.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets (such as those in Questions 6 and 7) readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of CSE455 Assignment 1, 2002.

CSE455 LEARNING AND PREDICTION II: MML "Data Mining"

Assignment 2, 1st Semester, 2002 - *and* **worth 30%.**

DUE: 12:00 noon, Wednesday 12th June 2002,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 30% of the total assessment for this subject.
Please read *carefully* the submission requirements above and on page 6.
Note also that only one of the four parts of Question 6 is to be attempted.
Total marks: $7 + 0 + 1 + 3 + 12 + 7 \ = \ 30$ marks.

Introduction

C5.0

Recall the decision tree program, C5.0 (due to J. Ross Quinlan, www.rulequest.com), a copy of which is installed on nexus via /usr/local/lib/c5/bin/c5.0 . There are also some accompanying data-sets at /usr/local/lib/c5/Data/ .

C5 was used in CSE454 Prac' 2, and some notes on C5 are given at
http://www.rulequest.com and at
http://www.csse.monash.edu.au/~lloyd/tildeMML/Other/C5 .

Tan-Dowe multi-way join decision graph

At ~ptan/M1/ver1.2 and ~ptan/M1/ver1.2/sunos (these are directories, not WWW URLs), you will find the Tan-Dowe multi-way join decision graph program, written by Peter J. Tan. It has the same input format as C5.

Question 1 (worth 7 marks)

Use this above-mentioned multi-way join decision graph program to analyse one of the data-sets in the /usr/local/lib/c5/Data/ directory from CSE454 Ass't 1. You are assigned as follows.
anneal - Doug
breast-cancer - Sarah
credit - Yvonne, Ryan
genetics - Brian, Di Wu
letter - Andris, Andrew, + anyone not listed
sonar - Susie

Whereas C5 can deal with continous-valued attributes, the slightly bad news is that the current version of the above-mentioned multi-way join decision graph program currently can not yet.

The data-sets *breast-cancer*, *letter* and *sonar* appear to have only continuous-valued attributes. The data-set *genetics* has only non-continuous, discrete-valued attributes. Both the others, *anneal* and *credit*, have both continuous- and discrete-valued attributes.

Recall that in an MML decision tree and decision graph framework, it is customary to assume that *both sender and receiver know the non-target attributes in advance* and that is the the job of the sender to trasmit the target attribute (to the receiver) as concisely as possible using the known non-target attributes and some decision tree/graph hypothesis.

For each continuous-valued non-target attribute in your data-set, it will be necessary to make some sort of discretisation in order that it can be used in a model by the current version of the multi-way join decision graph program. For each continuous-valued non-target attribute in your data-set, discretise the attribute as appropriately as you can. Attempt to justify your approach.

In 1-page, draw the graph (nicely!), or the top levels if the whole is too big. If you had no continuous-valued attributes in your data-set, then draw the graph in a little extra detail. Describe what your inferred and drawn graph means for the data set.

Question 2 (worth 0 marks)
What, if anything, can you say about your model from Question 1 above as compared to your model using C5 from CSE454 Prac' 2, Question 1? (This question is worth 0 marks on this assignment.)

Question 3 (worth 1 mark)
Given a true underlying multinomial model for generating the data, $\vec{p}$, and an inferred multinomial model from a relevant generated data-set, $\hat{\vec{p}}$, define the Kullback-Leibler distance from $\vec{p}$ to $\hat{\vec{p}}$.

Regarding Questions 4 and 5, recall the XD6 generation process from CSE454 Prac' 2 and http://www.csse.monash.edu.au/~lloyd/tildeMML/Other/XD6 . Also recall that any XD6 data-set will have only discrete attributes.

Question 4 (worth 3 marks)
Given a true underlying tree/graph model for generating the data (such as that of XD6) and an inferred tree/graph model from a relevant data-set (such as one obtained from C5 or from the multi-way join decision graph), (generalise your answer to Question 3 to) describe how to calculate the Kullback-Leibler distance from the true model to the inferred model.

Question 5 (worth 12 marks)
Recall CSE454 Prac' 2 Question 4 concerning XD6 and C5. This question will be similar (and verbatim in parts), but for multi-way join decision graphs (instead of C5).

Investigate the ability of the multi-way join decision graph to learn a good model, or the true model, for XD6 as you vary
i. the size of the training data and
ii. the noise level (you will need to modify the generator).

Recalling C5's performance at least in part, write a short report on the multi-way join decision graph's performance. You might like to include
i. Optionally, results on a reduced XD' data set.
ii. Some example trees.
iii. Tables of results.
iv. Right/wrong scores on test data.
v. The "closeness" of true and inferred trees (recall your answer to Question 4).
vi. Other ?

Question 6 (worth 7 marks)
Attempt *exactly one* and no more than one of Questions 6A, 6B, 6C and 6D. (If more than one of these is attempted, then it will be at the discretion of the marker as to which one is marked.)

Question 6A
Recall CSE455 Ass't 1, Questions 3, 6 and 7 and likewise recall your models of the $\frac{1}{\pi}\ \cos^2(\frac{n}{2}(\theta - \mu))$ distribution with $n = 1$ and $n = n_{me}$.

Consider a 2-component mixture model of the form
$p[\frac{1}{\pi}\ \cos^2(\frac{1}{2}(\theta - \mu_1))] + (1-p)[\frac{1}{\pi}\ \cos^2(\frac{n}{2}(\theta - \mu_2))]$.

6A (i) What are the parameters of this mixture model?

For the next part of the question, you might possibly want to follow Wallace and Dowe (2000): "MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions", Statistics and Computing, Vol. 10, No. 1, Jan. 2000, pp73-83
(http://www.csse.monash.edu.au/~dld/Snob.html).

6A (ii) Write out the parts of the MML message and outline how you would cost - in message length terms - each of them.

6A (iii) Write a program to do such 2-component mixture modelling.

6A (iv) Re-analyse the dog-bite data (from CSE455 Ass't 1, Qu 8 and CSE454 Prac' 1 at

www.csse.monash.edu.au/~lloyd/tilde/CSC4/CSE454/Local/dog) using this program.

Question 6B (random coding)
Consider a form of MML inference (often reserved for intractable problems) called *random coding*. Both sender and receiver have a prior, $h(\theta)$, a (negative) log-likelihood function, $L(\vec{x}|\theta) = -\log f(\vec{x}|\theta)$ *and* the *same* (pseudo-)random number generator with the *same* seed.

Parameters $\theta_1, \theta_2, ..., \theta_i, ...$ are sampled from the prior. Using a code for the integers, such as $\log^*$, θ_i can be encoded using, e.g., the code for $\log^*(i)$.

The length of the second-part of the message would be $-\log f(\vec{x}|\theta_i)$.

Using random coding, re-visit CSE455 Ass't 1 Questions 3, 6 and 7.

Question 6C
Choose i a "random" integer between 1 and 20,000 by taking (the last 5 digits of) your student_id modulo 20,000.

6C (i) What is your value of i? (0 marks)

Consider the base 10 digits of π (listed in large part at
http://newton.ex.ac.uk/research/semiconductors/theory/collabs/pi/)
and the 10,000 base 10 digits of π from position i to position $i+9999$. In a manner about to be described, each of these 10,000 base 10 digits is to be mutated with probability 0.6. Seed your (pseudo-)random number generator with the last 5 digits of your student_id. For each digit, d_j, in turn, leave it as is with probability 0.4 and, with probability 0.6, perform the following operation: use your (pseudo-)random number generator to choose a digit, d', from 0 to 9 and change the digit, d_j (from π), to become d'.

6C (ii) What is your resultant list (or data-set) of 10,000 base 10 digits?

Assume that there exists a program of length 100 bytes (or 800 bits) which, given inputs m and n, can output the base 10 digits of π from positions m to n. (Feel invited to comment on this assumption.)

At this point, you might possibly want to examine Wallace and Dowe (1999), "Minimum Message Length and Kolmogorov complexity", Comp. J., Vol 42, No. 4 (1999), pp270-283 { CSE455/DLD/2002/6 }.

6C (iii) Give an MML inference and a message length for your data-set. The message length should include the cost of the first part (and any sub-parts) and the second part.

Question 6D

Quotation(s):

"(So-called) data mining is the supposed 'art' of making pretty graphics of over-fitting and spurious correlations." - Anonymous.

"If you can't develop anything new, give something a new name." - Anonymous.

6D (i) Graph the prior distribution on a single parameter, p, $h_0(p) = \frac{1!}{0!0!}p^0(1-p)^0 = 1$ for $0 \leq p \leq 1$.

6D (ii) Graph the prior distribution, $h_4(p) = \frac{9!}{4!4!}p^4(1-p)^4 = 630p^4(1-p)^4$ for $0 \leq p \leq 1$.

6D (iii) Using prior h_0, generate (and record) a parameter, p.
Using p, generate 15 binary data points, and record this data set.
Using the prior, $h_0(p)$, and the generated data set, infer a value $\hat{p}$ for p.

6D (iv) Repeat 6D (iii) but replacing the prior $h_0(p)$ by $h_4(p)$ throughout:
Using prior h_4, generate (and record) a parameter, p.
Using p, generate 15 binary data points, and record this data set.
Using the prior, $h_4(p)$, and the generated data set, infer a value $\hat{p}$ for p.

6D (v) Using the value of p generated from prior h_0 in 6D (iii), generate and record 100 data-sets each of size 15.
For which of these data-sets would you infer the largest/smallest value of $\hat{p}$?
What is the largest/smallest value of $\hat{p}$ that you would infer?

6D (vi) Using prior $h_4(p)$, generate and record 100 different values of p.
What are the largest and smallest values of p generated?
For each value of p, generate and record 100 binary data-sets of size 15.
For which of these data sets would you infer the largest/smallest value of $\hat{p}$?
What is the largest/smallest value of $\hat{p}$ that you would infer?

6D (vii) Feel invited to comment on your results from the four parts immediately above, namely 6D (iii) to 6D (vi).

Make sure that you have attempted *exactly one* and no more than one of Questions 6A, 6B, 6C and 6D. If more than one of these is attempted, then it will be at the discretion of the marker as to which one is marked.

Submission requirements - *please read carefully*

Any programs should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java.

Submit any source code written (both in hard copy and in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission and be submitted as on page 1 of this assignment. The soft copy of your source code should be sent as plain ASCII text with Subject line: "CSE455 Assignment 2" to dld@cs.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets (such as those in Questions 5 and 6) readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of CSE455 Assignment 2, 2002.

CSE455 LEARNING AND PREDICTION II: MML "Data Mining"

Assignment 1, 2nd Semester, 2003 - *and* **worth 20%**.

DUE: 12:00 noon, Monday 1st September 2003,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 20% of the total assessment for this subject.
Please read *carefully* the submission requirements above and on page 6.
Total marks: $4 + 4 + 4 + 1 + 1 + 1 + 0 + 5 + 0 \ = \ 20$.

Introduction
Recall the probabilistic football prediction competition at
http://www.csse.monash.edu.au/~footy/ , noting both the *probabilistic* competition and the *Gaussian* competition. We use this as a guide to comparing inference (to the one single best explanation) with prediction (doing a weighted average of several explanations). The data from this will be used for the first part of this assignment.

Note also the (Tan and Dowe, 2002) multi-way join decision graph program from:
Tan, P.J. and D.L. Dowe (2002). MML Inference of Decision Graphs with Multi-Way Joins. Proc. 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia, 2-6 Dec. 2002, Lecture Notes in Artificial Intelligence (LNAI) 2557, Springer-Verlag, pp131-142.
This paper also gives comparisons with J.R. Quinlan's decision tree programs, C4.5 and (from www.rulequest.com) C5.0. The (Tan and Dowe, 2002) executable code is currently available - solely for the purposes of this assignment - at www.csse.monash.edu.au/~ptan/ or www.csse.monash.edu.au/~ptan/dgraph.zip . A licensed version of C5.0 is available from on (a machine called) nexus in /local/lib/c5/bin .
Decision tree/graph analysis will be used with some DNA micro-array data pertaining to oncological (or cancer) data for the second (and last) part of the assignment. Unless we find another data-set, the data-set I have in mind is the "van't Veer" breast cancer data-set, which is downloadable from http://www.rii.com/publications/2002/vantveer.htm . If you are having trouble because of your InterNet browser, then this "van't Veer" data-set should also be obtainable from http://www.csse.monash.edu.au/~ptan/ or
http://www.csse.monash.edu.au/~ptan/ArrayData_less_than_5yr.zip . If you would rather analyse a colon, rectal or colorectal cancer data-set and are also capable of finding and obtaining such a data-set, please confer with me.
The "van't Veer" data-set apparently has 25000 continuous input attributes, 1 binary output attribute and 78 data things.

First part of the assignment - footy-tipping questions

Consider the data from the 2003 footy-tipping competition.

Question 1 (worth 4 marks)

Consider probabilistic tippers which, having seen the data from the first i rounds, use the following approach to do their tips in round $(i+1)$:
(i) the best tipper (so far)
(ii) the average (weighted equally) of all tippers (so far)
(iii) the weighted average of all tippers (so far)

Explicitly state and briefly explain or justify your choice(s) of prior probabilities.

For each of these three tippers, (i), (ii) and (iii), give the score of each game in each round. Also, for each round up until the submission of your assignment, give the score from that round and the cumulative score after that round.

Any comments or conclusions?

Leading into the Gaussian competition, note that if N Gaussian distributions $N(\mu_i, \sigma_i^2)$, $i = 1, \ldots, N$ are weighted $w_i, i = 1, \ldots, N$ where $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$, then the mean of the weighted sum is $\mu_w = \sum_{i=1}^{N} w_i \mu_i$ and the variance of the weighted sum is $\sigma_w^2 = \sum_{i=1}^{N} w_i(\sigma_i^2 + (\mu_w - \mu_i)^2) = (\sum_{i=1}^{N} w_i \sigma_i^2) + (\sum_{i=1}^{N} w_i(\mu_w - \mu_i)^2)$.

Question 2 (worth 4 marks)

Consider Gaussian tippers which, having seen the data from the first i rounds, use the following approach to do their tips in round $(i+1)$:
(i) the best tipper (so far)
(ii) the average (weighted equally) of all tippers (so far)
(iii) the weighted average of all tippers (so far)

All tippers in Question 2 must be Gaussian.
For each of these three tippers, (i), (ii) and (iii), give the score of each game in each round. Also, for each round up until the submission of your assignment, give the score from that round and the cumulative score after that round.

Question 3 (worth 4 marks)

Re-visit Question 2 above where now tippers (ii) and (iii) are permitted to be mixture models (with mixing proportions or relative abundances given by the weights) of Gaussian

distributions. To (attempt to) avoid confusion, refer to 3 (ii) as (iv) and refer to 3 (iii) as (v).

Any comments or conclusions?

Second part of the assignment: dgraph, angular data and onco' data-sets

***sin*^{2} and *cos*^{2} circular distributions**

Let $f(\theta|\mu) = \frac{1}{\pi}\cos^2(\frac{n(\theta-\mu)}{2}) = \frac{1}{\pi}\cos^2(\frac{n}{2}(\theta-\mu)) = \frac{1}{\pi}\cos^2(\frac{n}{2}(\theta-\mu)) = \frac{1}{\pi}\sin^2(\frac{n}{2}(\theta-\mu)-\frac{\pi}{2})$ for some n, presumably a positive integer; $0 \leq \theta \leq 2\pi$.

Notice that for a von Mises distribution, for a wrapped Normal distribution and for the $\cos^2$ distribution above, if n is a positive integer then adding to or subtracting from θ an amount of 2π or any integer multiple of 2π does not change the value of the likelihood function or any of its derivatives.

$$L \quad = \quad -\log \pi_{i=1}^{N} f(\theta_i|\mu) \;=\; N\log(\pi) - \sum_{i=1}^{N} \log\cos^2(\frac{n}{2}(\theta-\mu))$$

$$\frac{\partial L}{\partial \mu} \quad = \quad -\sum_{i=1}^{N} \frac{2\ \cos(\frac{n}{2}(\theta_i-\mu))\frac{n}{2}\sin(\frac{n}{2}(\theta-\mu))}{\cos^2(\frac{n}{2}(\theta_i-\mu))} \;=\; -n\sum_{i=1}^{N}\tan(\frac{n}{2}(\theta_i-\mu)) \;=\; n\sum_{i=1}^{N}\tan(\frac{n}{2}(\mu-\theta_i))$$

$$\frac{\partial^2 L}{\partial \mu^2} \quad = \quad n\sum_{i=1}^{N}\frac{n}{2}\sec^2(\frac{\mathrm{n}}{2}(\mu-\theta_{\mathrm{i}})) \;=\; \frac{\mathrm{n}^2}{2}\sum_{\mathrm{i}=1}^{\mathrm{N}}\sec^2(\frac{\mathrm{n}}{2}(\mu-\theta_{\mathrm{i}})) \;=\; \frac{\mathrm{n}^2}{2}\sum_{\mathrm{i}=1}^{\mathrm{N}}\frac{1}{\cos^2(\frac{\mathrm{n}}{2}(\mu-\theta_{\mathrm{i}}))}$$

$$F(\mu) \quad = \quad E(\frac{\partial^2 L}{\partial \mu^2}) \;=\; \frac{Nn^2}{2}\ E(\frac{1}{\cos^2(\frac{n}{2}(\mu-\theta_i))})$$

$$E(\frac{1}{\cos^2(\frac{n}{2}(\mu-\theta_i))}) \;=\; \int_0^{2\pi}\frac{1}{\pi}\cos^2(\frac{n}{2}(\mu-\theta_i))\frac{1}{\cos^2(\frac{n}{2}(\mu-\theta_i))}d\theta \;=\; \frac{1}{\pi}\times 2\pi = 2$$

(This result above also follows by symmetry.)

So, $F(\mu) = \frac{Nn^2}{2}\times\frac{1}{\pi}\times 2\pi = \frac{Nn^2}{2}\times 2 \;=\; n^2N.$ For $n=2$, $F(\mu) = 4N$.

I will be amongst the first to admit that there maybe a missing factor of 2 or $\frac{1}{2}$ or a floating minus sign, $-$, or some such in the above. So, please feel highly invited to check the above mathematics and correct any possible such mistake.

The sender and receiver can agree by convention to scale the explanatory (or input) attributes to range from 0 to π or to range from 0 to 2π or to be some (possibly asymmetric) subset thereof. Having discussed hte $\sin^2$ and $\cos^2$ distributions, we now lead into a discussion of the von Mises circular distribution.

Snob uses the von Mises circular distribution, $M_2(\mu, \kappa)$, to analyse angular data. Another possible model for angular data is the wrapped Normal distribution, $WN(\mu, \sigma^2)$. Of course, there are other distributions for angular data.

Snob software
The Snob software is available from
http://www.csse.monash.edu.au/~dld/Snob.html (and also from
http://www.csse.monash.edu.au/research/mdmc/software).
It should be installed on the machines in the CSSE Clayton Bldg. 26 Hons lab, and on the CSSE indy's. It is capable of analysing at least the statistical distributions described in the title of the (Wallace and Dowe, 2000) paper:
Wallace, C.S. and D.L. Dowe (2000). MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions, Statistics and Computing, Vol. 10, No. 1, Jan. 2000, pp73-83.

von Mises circular distribution
The 2-dimensional von Mises density, $M_2(\mu, \kappa)$ or $VM(\mu, \kappa)$, is an analogue of the Gaussian density for angles in the plane.

Let $I_0(\kappa) = \frac{1}{2\pi} \int_0^{2\pi} e^{\kappa \cos(\theta)} \, d\theta = \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r}}{(r!)^2}$ and for $p > 0$,
let $I_p(\kappa) = I_0(\kappa) \times E(\cos(p\theta)) = I_0(\kappa) \times \frac{1}{2\pi} \int_0^{2\pi} \cos(p\theta) e^{\kappa \cos(\theta)} \, d\theta = \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r+p}}{(p+r)! \; r!}$.

So, $I_1(\kappa) = I_0(\kappa) \times E\left(\cos(\theta)\right) = \sum_{r=0}^{\infty} \frac{(\frac{\kappa}{2})^{2r+1}}{r! \; (r+1)!} = \frac{d \; I_0(\kappa)}{d\kappa}$.

The density of the angular variate θ is given by $f(\theta) = 1/(2\pi I_0(\kappa)).e^{\kappa \cos(\theta - \mu)}$, where $I_0(\kappa)$ is a normalisation constant. The functional form of the likelihood is

$f(\theta|\mu, \kappa) = \frac{1}{2\pi I_0(\kappa)} \, e^{\kappa \cos(x-\mu)}$, and is sometimes written $\theta \sim M_2(\mu, \kappa)$.

Note (about random number generation) :

For the duration of the assignment, software - should you need it - will be available to generate multinomial, Gaussian, Poisson and von Mises distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
http://www.csse.monash.edu.au/~dld/datalinks.html ,
(maybe) www.csse.monash.edu.au/~dld/Hons/2001/dldprojects (under "(16)") (maybe),

http://random.mat.sbg.ac.at/links/rando.html ,
and http://www.csse.monash.edu.au/research/mdmc/software/random/index.shtml ,
although you should feel free to use any decent (pseudo-)random number generator that you like.
Note (about random coding in MML) :
Please ask if you would like to know about random coding in MML.
Question 4 (worth 1 mark)
For this cos^2 model, obtain a (possibly implicit) formula for the maximum likelihood estimator of μ, $\hat{\mu}_{ML}$, given data $\vec{\theta} = \{\theta_1, \ldots, \theta_i, \ldots \theta_N\}$. Do this for general n.

Question 5 (worth 1 mark)
Assuming a uniform prior on μ, use the above to derive a message length for the cos^2 model and given data $\vec{\theta} = \{\theta_1, \ldots, \theta_i, \ldots \theta_N\}$. Do this for general n.

Question 6 (worth 1 mark)
Minimise this message length expression to obtain the minimum message length (MML) estimator, $\hat{\mu}_{MML}$, of μ. What can you say, if anything, about the relationship between the maximum likelihood estimator, $\hat{\mu}_{ML}$ and the minimum message length (MML) estimator, $\hat{\mu}_{MML}$?

Your name and student id, etc. should be attached to the green sheet at the front of your assignment. Let this be your own personal seed, *seed*.

Question 7 (worth 0 marks)
What is your value of *seed*?

Question 8 (worth 5 marks and possibly bonus marks)
Consider your chosen data-set, which is possibly the "van't Veer" data-set. (Please confer with me if you have chosen a different oncological data-set.) Use your value of *seed* to seed a random number generator and select appropriately a "handful" of attributes from the approximately 25000 in the data-set. Give the numbers of the selected attributes. Use the (Tan and Dowe, 2002) MML decision graph program, MML $\cos^2$ regression and/or any (other?) techniques you deem appropriate to analyse the data-set. Please give the decision graph with the shortest message length you could find, clearly stating the decision graph and clearly stating the message length (in bits and nits). Please give the $\cos^2$ (and $\sin^2$) regression with the shortest message length you could find, clearly plotting a graph of the function and stating the message length (in bits and nits).

Question 9 (worth 0 marks)
List some of your favourite colours and describe their suitability for data mining.

Submission requirements - *please read carefully*

Any programs should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java.

Submit any source code written (*both* in hard copy *and* in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission and be submitted as on page 1 of this assignment. The soft copy of your source code should be sent as plain ASCII text with Subject line: "CSE455 Assignment 1" to dld@csse.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets (such as those in Question 8) readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of CSE455 Assignment 1, 2003.

CSE455 LEARNING AND PREDICTION II: MML "Data Mining"

Assignment 2, 2nd Semester, 2003 - *and* **worth 30%**.

DUE: 12:00 noon, Monday 6th October 2003,
at the Computer Science and Software Eng. (Clayton) General Office

This assignment is worth 30% of the total assessment for this subject.
Please read *carefully* the submission requirements above and on page 4.
Total marks: $10 + 0 + 0 + 20 = 30$.

Introduction
The notion of (logarithm of) probabilistic scoring dates back to I.J. Good (1952) and I.J. Good (1968) for binomial target distributions, and this has been re-iterated in Needham and Dowe (2001). This has been extended in Dowe and Krusel (1993) (to multinomials), Dowe, Farr, Hurst and Lentin (1996) (to Gaussian distributions), Tan and Dowe (2002) (again to multinomials) and Tan and Dowe (2003) (once more again to multinomials). Some fuller references are:

Needham, S.L. and D.L. Dowe (2001). Message Length as an Effective Ockham's Razor in Decision Tree Induction. Proc. 8th International Workshop on Artificial Intelligence and Statistics (AI+STATS 2001), pp253-260, Key West, Florida, U.S.A., Jan. 2001

Tan, P.J. and D.L. Dowe (2002). MML Inference of Decision Graphs with Multi-Way Joins. Proc. 15th Australian Joint Conference on Artificial Intelligence, Canberra, Australia, 2-6 Dec. 2002, Published in Lecture Notes in Artificial Intelligence (LNAI) 2557, Springer-Verlag, pp131-142 (this has been handed out in class, and was used in CSE455 Ass't 1 from www.csse.monash.edu.au/∼ptan or www.csse.monash.edu.au/∼ptan/dgraph.zip)

and

P. J. Tan and D. L. Dowe (2003). MML Inference of Decision Graphs with Multi-Way Joins and Dynamic Attributes, (to appear) In Proc. 16th Australian Joint Conference on Artificial Intelligence (AI'03), Perth, Australia, 3-5 Dec. 2003 (which is downloadable from http://www.csse.monash.edu.au/∼dld/Publications/2003/Tan+Dowe2003.ref as 12 pages of .pdf or .ps).

If anyone is interested in C5.0, a licensed version of C5.0 is available from (a machine called) nexus.csse in /local/lib/c5/bin and both (Tan and Dowe, 2002) and (Tan and Dowe, 2003) give empirical comparisons of MML decision graph schemes with both C4.5 and C5. More about C5.0 (for anyone wanting this) is at www.rulequest.com .

Some four-limbed biped primates have written and published papers with arguments suggesting that *Ockham's razor* (see, e.g., http://www.csse.monash.edu.au/~dld/Ockham.html) is false. Closer examination of such writings tends to suggest that the authors actually believed what they were writing at the time. In my own attempts to discuss such published papers with the relevant authors, the only responses I have heard at the time of writing have been either recanting or unclear.

The assignment shall be in two parts. Re-visiting the MML decision tree analysis of (Needham and Dowe, 2001) using decision graphs will be the first part of CSE455 Assignment 1. The second (and last) part of the assignment shall pertain to (decision tree/graph or possibly other) analysis of some (DNA micro-array) data pertaining to (colon, rectal or colorectal) oncological (or cancer) data.

In CSE455 Assignment 1, we used the "van't Veer" breast cancer data-set, which is downloadable from http://www.rii.com/publications/2002/vantveer.htm . If you are having trouble because of your InterNet browser, then this "van't Veer" data-set should also be obtainable from http://www.csse.monash.edu.au/~ptan/ or
http://www.csse.monash.edu.au/~ptan/ArrayData_less_than_5yr.zip .
The "van't Veer" data-set apparently has 25000 continuous input attributes, 1 binary output attribute and 78 data things.

If you would rather analyse a colon, rectal or colorectal cancer data-set for CSE455 Assignment 2 and are also capable of finding and obtaining such a data-set, that would be desirable. Here are the WWW URLs of some such colorectal cancer data-sets:
(Notterham's data) http://microarray.princeton.edu/oncology/
(Alon's data) http://microarray.princeton.edu/oncology/affydata/index.html
(Agrawal's data) http://cancer.tigr.org/c_pooling.shtml , and
(Ramaswamy's data) http://www-genome.wi.mit.edu/cgi-bin/cancer/datasets.cgi .

First part of the assignment - Ockham's razor

Question 1 ($0 + 2 + 8 = 10$ marks)

1 (a) Give a verbal, English-language, statement of Ockham's razor.

1 (b) Interpret and re-state this in a message length framework.

1 (c) Considering, e.g., the boolean function of 5 binary attributes X, Y, Z, A and B:
(X and Y and Z) or (A and B),
re-visit the decision tree study in (Needham and Dowe, 2001) using decision graphs.

Any comments or conclusions?

Second part of the assignment: oncological data-set(s)

Your name and student id, etc. should be attached to the green sheet at the front of your assignment. Let your student id be your own personal seed, *seed*.

Question 2 (worth 0 marks)
What is your value of *seed*?

Question 3 (worth 0 marks)
3 (a) Which data-set have you chosen?
3 (b) How many attributes does your data-set have?
3 (c) How many things does your data-set have?

Question 4 (worth 20 marks)
Consider your chosen data-set, which is probably one of the colorectal data-sets previously mentioned (on page 2) and probably not the "van't Veer" data-set (unless you have obtained permission from the lecturer). (Please confer with the lecturer if you have chosen a different oncological data-set.)

If necessary, use your value of *seed* to seed a random number generator and select appropriately a "handful" of attributes from the your (colorectal) data-set. Give the numbers of the selected attributes.

Use the (Tan and Dowe, 2002) MML decision graph program, MML $\cos^2$ regression, random coding (see CSE455 Assignment 1) and/or any (other?) techniques you deem appropriate to analyse the data-set.

Of the models you use (decision graph or other), please give the model (decision graph or other) with the shortest message length you could find, clearly stating the decision graph (or other function) and clearly stating the message length (in bits and nits).
If appropriate, please give the $\cos^2$ (and $\sin^2$) regression with the shortest message length you could find, clearly plotting a graph of the function and stating the message length (in bits and nits).

Note (about random number generation) :

For the duration of the assignment, software - should you need it - will be available to generate multinomial, Gaussian, Poisson and von Mises distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ ,
http://www.csse.monash.edu.au/~dld/datalinks.html ,
(maybe) www.csse.monash.edu.au/~dld/Hons/2001/dldprojects (under "(16)") (maybe),
http://random.mat.sbg.ac.at/links/rando.html ,
and http://www.csse.monash.edu.au/research/mdmc/software/random/index.shtml ,
although you should feel free to use any decent (pseudo-)random number generator that you like.

Submission requirements - *please read carefully*

Any programs should be written in a Linux/Unix environment at Monash CSSE and should use one of the languages C, C++ or Java or some other language (you can haggle about Perl, PHP, etc.) which the lecturer has agreed to.

Submit any source code written (*both* in hard copy *and* in soft copy) along with your assignment solutions and answers. The hard copy of your source code should appear as an Appendix to your assignment submission and be submitted as on page 1 of this assignment. The soft copy of your source code should be sent as plain ASCII text with Subject line: "CSE455 Assignment 1" to dld@csse.monash.edu.au . It should be sent from one of the Linux/Unix machines at Monash CSSE on which you did your work.

Make your data-sets (such as those in Question 3 and 4) readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

End of CSE455 Assignment 2, 2003.

CSE455 MINIMUM MESSAGE LENGTH

Assignment 1

DUE: 12:00 noon, Wednesday 5 April 2006,
at the CSSE (Clayton School of I.T.) Bldg. 75 General Office

This assignment is worth 20% of the total assessment for this subject/unit.
Please read carefully the submission requirements on page 3.

Introduction
This assignment asks students to use the Poisson distribution (C. S. Wallace and D. L. Dowe, "MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions", *Statistics and Computing*, Vol. 10, No. 1, Jan. 2000, pp73-83) to analyse some "*secret*" but not unfriendly data.
(It has previously been [and might again be] used to analyse word-count data, dog-bite data and DNA micro-array data, and could be used to analyse radioactive decay data.)

The following two sets of points outline the progression throught the assignment and some of the concepts and issues to be encountered.

- Brief justification of Minimum Message Length (MML)
- Poisson likelihood, Fisher information, Bayesian prior
- Wallace-Freeman (J. Royal Stat. Soc.) 1987 approximation to message length
- Maximum Likelihood estimate
- Wallace-Freeman 1987 MML estimate
- Data could be decays/transitions/changes of elements/structures
- Real-world data
- Data-set could be self-contradictory (e.g., same thing/item could have two contradictory values)
- There could be outliers, such as transitions going in a seemingly impossible direction.

As such (from the last two points above), we might wish to generalise our model space. Addressing this second set of points will form a substantial part of the assignment.

- The data could possibly be better summarised by a mixture of two or more Poisson distributions than just one

- The data could possibly be better summarised by a mixture of an outlier distribution (possibly uniform) and one or more Poisson distributions than just a lone Poisson distribution
- The data could possibly be better summarised as having one or more changes/cut-points into two or more segments.

By way of introducing questions 2 and 3, we introduce (a version of) the Poisson distribution.
With rate r and duration (or length) t_i [which you might choose to think of as the time for the counted events] and count c_i, the likelihood fuction of the Poisson distribution is $f(\vec{c}|\vec{t},r) = f(c_1,...,c_N|t_1,...,t_N,r) = \prod_{i=1}^{N} f(c_i|t_i,r)$ where $f(c_i|t_i,r) = e^{-rt_i}\frac{(rt_i)^{c_i}}{c_i!}$.
$L_i = -\log f(c_i|t_i,r) = rt_i - c_i\log(r) - c_i\log(t_i) + \log(c_i!)$ and $L = -\log f = \sum_{i=1}^{N} L_i$ and $\frac{\partial L_i}{\partial r} = t_i - \frac{c_i}{r}$.
For some $\alpha > 0$, assume a Bayesian prior on r of $h(\alpha) = \frac{1}{\alpha}e^{-\frac{r}{\alpha}}$.

Question 0 (worth 0 marks)
Given r (the rate) and t_i, what is the "expected" value of c_i ?

Question 1 (worth 4 marks)
Appealing to any/some/all of
(i) Bayes's theorem,
(ii) (Universal) Turing Machines and/or Kolmogorov complexity,
(iii) file compression,
(iv) Ockham's razor, and
(v) anything else,
give an intuitive justification of Minimum Message Length (MML).

Question 2 (worth 6 + 4 + 2 + possibly bonus = 12 + possibly bonus marks)
Calculate a likelihood, message length or other viable objective function for the problems referred to above.
State or explain how you will minimise the message length.
Develop software to calculate the objective function(s) and find the optima.
Test your software using (see below) appropriate (pseudo-)random number generator software.

Question 3 (worth 4 + possibly bonus marks)
Apply this to the relevant real-world data at the CSE455 courseware WWW page.
A sample of some of this is given below.

Note (about random number generation) :
For the duration of the assignment, software will be available to (pseudo-)randomly generate multinomial, Gaussian, Poisson and other distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ (or elsewhere),
although you should feel free to use any decent (pseudo-)random number generator that you like.

Submission requirements - *please read carefully*
Submit any source code written along with your assignment solutions and answers.
Make your data-sets readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

Appendix - sample data

YearCon dateI pc1 pc2 pc3 pc4 Entity No.
1975 1997 0 0 100 0 1
1975 1999 100 0 0 0 1
1960 1999 98 2 0 0 2
1960 1999 98 2 0 0 3
1970 1996 0 80 20 0 4
1970 1999 50 50 0 0 4
1971 1996 100 0 0 0 5
1971 1999 98 2 0 0 5
1971 1996 100 0 0 0 6
1971 1999 80 20 0 0 6
1960 1996 0 40 50 10 7
1960 1999 23 40 30 7 7
1962 1996 70 30 0 0 8
1962 1999 90 10 0 0 8
1965 1996 80 20 0 0 9
1971 1996 70 0 30 0 10
1971 1999 80 20 0 0 10
1963 1996 80 20 0 0 11
1963 1999 80 20 0 0 11
1964 1996 0 75 25 0 12
1964 1999 75 20 5 0 12
1962 1996 75 25 0 0 13
1962 1999 70 20 10 0 13

1961 1996 90 10 0 0 14
1961 1999 100 0 0 0 14
1960 1996 100 0 0 0 15
1960 1999 100 0 0 0 15
1979 1996 100 0 0 0 17
1979 1999 85 15 0 0 17
1961 1999 95 5 0 0 18
1972 1996 90 10 0 0 19
1970 1999 100 0 0 0 20
1960 1996 80 20 0 0 21
1960 1999 100 0 0 0 21
1999 2000 100 0 0 0 22

CSE455 MINIMUM MESSAGE LENGTH

Assignment 2

DUE: 12:00 noon, Friday 28 April 2006,
at the CSSE (Clayton School of I.T.) Bldg. 75 General Office

This assignment is worth 30% of the total assessment for this subject/unit.
Please read carefully the submission requirements on page 4.

Introduction
This assignment asks students to use the Poisson distribution (C. S. Wallace and D. L. Dowe, "MML clustering of multi-state, Poisson, von Mises circular and Gaussian distributions", *Statistics and Computing*, Vol. 10, No. 1, Jan. 2000, pp73-83) to analyse some "*secret*" but not unfriendly data.
(It has previously been [and might again be] used to analyse word-count data, dog-bite data and DNA micro-array data, and could be used to analyse radioactive decay data or frequency of WWW page usage.)

The assignment then goes on to look at (circular) ring data.
To some degree, this assignment is a continuation of or an extension of Assignment 1.

The following three sets of points outline the progression throught the assignment and some of the concepts and issues to be encountered.

- Poisson likelihood, Fisher information, Bayesian prior
- Wallace-Freeman (J. Royal Stat. Soc.) 1987 approximation to message length
- Maximum Likelihood estimate
- Wallace-Freeman 1987 MML estimate
- Data could be decays/transitions/changes of elements/structures, etc.
- Real-world data
- Data-set could be self-contradictory (e.g., same thing/item could have two contradictory values)
- There could be outliers, such as transitions going in a seemingly impossible direction.

As such (from the last two points above), we might wish to generalise our model space. Addressing this second set of points will form a substantial part of the assignment.

- The data could possibly be better summarised by a mixture of two or more Poisson distributions than just one
- The data could possibly be better summarised by a mixture of an outlier distribution (possibly uniform) and one or more Poisson distributions than just a lone Poisson distribution
- The data could possibly be better summarised as having one or more changes/cut-points into two or more segments

Of course, shifting to the rest of the assignment, the data could come from entirely difference sources (again, with or without outliers), such as:

- Circles (or rings) of data
- Mixtures of circles (or rings) (or even ellipses) of data, with or without outliers - such as the Olympic rings (www.olympic.org), with or without outliers.

Now, by way of introducing questions 1, 2 and 3, we introduce (a version of) the Poisson distribution.
With rate r and duration (or length) t_i [which you might choose to think of as the time for the counted events] and count c_i, the likelihood fuction of the Poisson distribution is $f(\vec{c}|\vec{t}, r) = f(c_1, ..., c_N|t_1, ..., t_N, r) = \prod_{i=1}^{N} f(c_i|t_i, r)$ where $f(c_i|t_i, r) = e^{-rt_i}\frac{(rt_i)^{c_i}}{c_i!}$.
$L_i = -\log f(c_i|t_i, r) = rt_i - c_i \log(r) - c_i \log(t_i) + \log(c_i!)$ and $L = -\log f = \sum_{i=1}^{N} L_i$ and $\frac{\partial L_i}{\partial r} = t_i - \frac{c_i}{r}$.
For some $\alpha > 0$, assume a Bayesian prior on r of $h(\alpha) = \frac{1}{\alpha}e^{-\frac{r}{\alpha}}$.

Question -1 (worth 0 marks) [from Assignment 1]
Appealing to any/some/all of
(i) Bayes's theorem,
(ii) (Universal) Turing Machines and/or Kolmogorov complexity,
(iii) file compression,
(iv) Ockham's razor, and
(v) anything else,
give an intuitive justification of Minimum Message Length (MML).

Question 0 (worth 0 marks) [from Assignment 1]
Given r (the rate) and t_i, what is the "expected" value of c_i ?

Question 1 (worth 2 marks) [partly from Assignment 1, but now repeated]
Calculate a likelihood and message length for the inference of a single Poisson distribution (no mixtures, no cut-points).

Give the Kullback-Leibler divergence (or Kullback-Leibler "distance") from a Poisson distribution parameterised by (the true) r to a Poisson distribution parameterised by some estimate, $\hat{r}$.

Question 2 (worth 6 + 4 + 2 + possibly bonus = 12 + possibly bonus marks)
Calculate a likelihood, message length or other viable objective function for the other problems (not in Question 1) referred to in the introduction above - this includes mixtures (at least two Poissons or an outlier distribution with at least one Poisson) and/or cut-points.
State or explain how you will minimise the message length.
Develop software to calculate the objective function(s) and find the optima.
Test your software using appropriate (pseudo-)random number generator software.

Question 3 (worth 4 + possibly bonus marks)
Apply your answer and software from Question 2 to the relevant real-world data at the CSE455 courseware WWW page. (A small sample of some of this is given below.) State any and all assumptions explicitly and very clearly. Discuss your results.

Question 4 (worth 8 + 2 + possibly bonus marks = 10 + possibly bonus marks)
Changing the topic, consider the following distribution for points around a circle:
$f((x,y)|x_0,y_0,r,n) = \mathrm{Norm(r,n)}.(((\mathrm{x}-\mathrm{x_0})^2+(\mathrm{y}-\mathrm{y_0})^2)/\mathrm{r}^2)^{\mathrm{n}}\mathrm{e}^{-\mathrm{n}((\mathrm{x}-\mathrm{x_0})^2+(\mathrm{y}-\mathrm{y_0})^2)/\mathrm{r}^2}$, where
$\mathrm{Norm(r,n)} = 1/(\pi\mathrm{r}^2\mathrm{n}!/(\mathrm{n}^{\mathrm{n}+1})) = \mathrm{n}^{\mathrm{n}+1}/(\pi\mathrm{r}^2\mathrm{n}!) = \mathrm{n}^{\mathrm{n}}/(\pi\mathrm{r}^2(\mathrm{n}-1)!) = \mathrm{n}^{\mathrm{n}}/(\pi\mathrm{r}^2\Gamma(\mathrm{n}))$.

With $r \geq 0$ and $n > 0$, the circle is centred at (x_0, y_0) with radius r, and n gives a measure of how tightly the data clusters around the circumference.

The things to look out for here (initially) are that
(i) this (the likelihood) has a minimum at $(x,y) = (x_0,y_0)$ (in the centre)
(ii) this peaks at $(x-x_0)^2+(y-y_0)^2 = r^2$ (on the circumference)
(iii) the value at the minimum is 0 and at the peak is $\mathrm{Norm(r,n)}.\mathrm{e}^{-\mathrm{n}}$
(iv) we have the normalisation constant correct, and
(v) the peak gets tighter for larger n.

x_0, y_0 and $r \geq 0$ are continuous whereas n could be either
4a) continuous, $n > 0$, or 4b) a positive integer.
Choose one of 4a) and 4b).

Develop formulas and software to calculate the Maximum Likelihood estimate of x_0, y_0, r and n.
Choosing suitable Bayesian priors, do your best to calculate an MML estimate of x_0, y_0, r and n.

4c) Cost an encoding of the data as background noise.

For both 4c) and your choice out of 4a) and 4b), give message lengths.

Question 5 (worth 2 + bonus marks)
5a) More generally, infer a way of modelling data as a mixture (model) (or cluster) of one or more circles (and possible background noise). Give message lengths.

5b) Test your software from Question 4 and Question 5a) on artificially generated data.

Note (about random number generation) :
For the duration of the assignment, software will be available to (pseudo-)randomly generate multinomial, Gaussian, Poisson and other distributions at
http://www.csse.monash.edu.au/~dld/random.numbers/ (or elsewhere),
although you should feel free to use any decent (pseudo-)random number generator that you like.

Submission requirements - *please read carefully*
Submit any source code written along with your assignment solutions and answers.
(Where possible, submit your written work in LaTeX.)
Make your data-sets readable from the time of submission until the assignment is returned, *and* include the path and file names of the data-sets in your printed submission.

Appendix - sample data for Poisson distribution and variants

```
1975 1997 0 0 100 0 1
1975 1999 100 0 0 0 1
1960 1999 98 2 0 0 2
1960 1999 98 2 0 0 3
1970 1996 0 80 20 0 4
1970 1999 50 50 0 0 4
1971 1996 100 0 0 0 5
1971 1999 98 2 0 0 5
1971 1996 100 0 0 0 6
1971 1999 80 20 0 0 6
1960 1996 0 40 50 10 7
1960 1999 23 40 30 7 7
   :
   :
1960 1999 100 0 0 0 21
1999 2000 100 0 0 0 22
```

References

[1] D.L. Dowe, R.A. Baxter, J.J. Oliver, and C.S. Wallace. Point Estimation using the Kullback-Leibler Loss Function and MML. In *Proc. 2nd Pacific Asian Conference on Knowledge Discovery and Data Mining (PAKDD'98)*, pages 87–95, Melbourne, Australia, April 1998. Springer Verlag.

[2] C.S. Wallace and D.L. Dowe. Minimum Message Length and Kolmogorov Complexity. *Computer Journal*, 42(4):270–283, 1999. Special issue on Kolmogorov Complexity; http://www3.oup.co.uk/computer_journal/hdb/Volume_42/Issue_04/pdf/420270.pdf.