

Bayesian Computation: Practical Exercises

Malcolm Farrow
University of Newcastle upon Tyne

November 2, 2005

This is a collection of practical exercises from old courses, mostly my old module in “Bayesian Computation”. Unless otherwise stated, references to lecture notes refer to the “Bayesian Computation” notes. The exercises use R and BUGS. Both R and BUGS are available in the School of Mathematics and Statistics.

You can download R from the Web, at no cost. There are versions for several operating systems including Unix, Mac OS and Windows. For further information see the following web site.

<http://cran.r-project.org/>

You can download BUGS from the Web, at no cost. There is a Windows version, WinBUGS, as well as the Unix version. The practicals are written in terms of the Unix version. For further information see the following web site.

<http://www.mrc-bsu.cam.ac.uk/bugs/>

1 Introduction

1.1 Smoking Students

1. Recall the experiment which involves finding twenty students and counting the number who smoke. Let us first look at the binomial distributions for this number, given various different possible values for the true proportion of smokers. Try making plots with $\theta = 0.1, 0.2, \dots, 0.5$. Look at how the probability of three smokers changes. Just plot numbers of smokers from 0 to 8.

R

Defining a function saves on repetitive typing. Don't forget to reset the `mfrow` parameter in the last line.

```
smokers<-seq(0,8)
ylim<-c(0.0,0.3)
theta<-0.1*seq(1,5)
probplot<-function(sm,th,yl)
{par(mfrow=c(3,2))
for (i in 1:5)
{prob<-dbinom(sm,20,th[i])
plot(sm,prob,type="h",ylim=yl)
}
par(mfrow=c(1,1))
}
probplot(smokers,theta,ylim)
```

2. Plot the prior and posterior probability density functions for θ in the smoking students example, with three smokers observed in the sample of twenty. Use each of the following prior distributions.

- (a) A beta distribution with $a = 1$, $b = 1$.
- (b) A beta distribution with $a = 2$, $b = 1$.
- (c) A beta distribution with $a = 1$, $b = 2$.
- (d) A beta distribution with $a = 2$, $b = 2$.
- (e) A mixture of two beta distributions with $p = 0.5$, $a_1 = 3$, $b_1 = 2$, $a_2 = 2$, $b_2 = 3$.
- (f) Anything else you would like to try.

See notes section 1.3.3.

R

```
> theta<-0.01*seq(1,99)
> prior<-dbeta(theta,1,1)
> density<-dbeta(theta,4,18)
> plot(theta,density,type="l")
> lines(theta,prior)
> text(0.32,3,"posterior")
> text(0.6,1.2,"prior")
```

2 Probability in Models

1. We will reproduce Figure 2 from the notes. Note that we need to divide the density by n because of the transformation from $\log(n)$. See “Supplementary Theory.”

R

```
> n<-seq(1,1000)
> logn<-log(n)
> var0<-0.3751
> sd0<-sqrt(var0)
> before<-dnorm(logn,5.605,sd0)
> before<-dnorm(logn,5.605,sd0)/n
> plot(n,before,type="l")
> var1<-0.04096
> sd1<-sqrt(var1)
> density<-dnorm(logn,5.863,sd1)/n
> plot(n,density,type="l")
> lines(n,before)
> text(100,0.003,"Before")
> text(500,0.003,"After")
```

3 Bayes’ Rule

1. Try the calculations described in Section 3.4 of the lecture notes for the Chester Road data.
2. Try changing the prior. For example you could try the following for the prior.
 - (a) A gamma distribution with $\alpha = 2$, $\beta = 10$.
 - (b) A gamma distribution with $\alpha = 8$, $\beta = 40$.
 - (c) A mixture of two gamma distributions with $p = 0.5$, $\alpha_1 = 2$, $\beta_1 = 20$, $\alpha_2 = 18$, $\beta_2 = 60$.
3. To illustrate the principles involved in using a non-conjugate prior, try using a uniform(0.0, 0.5) prior. The calculations are quite simple in this case but we will do them using a general method. The likelihood is proportional to

$$\lambda^{115} e^{-1200\lambda}.$$

It is usually better to work in terms of logs, for numerical reasons. The log likelihood, apart from an additive constant, is

$$115 \log(\lambda) - 1200\lambda.$$

We will also take logs of the prior density. Note that we could not do this where the prior density is zero but we really do not need to.

Having added the log prior to the log likelihood we subtract the maximum of this so that the maximum becomes zero. We then take exponentials and the maximum of this will be 1.0. The reason for doing this is to avoid very large or very small numbers. We then normalise by finding the integral and dividing by this. The integral is found numerically using a simple trapezium rule.

R

```
> lambda<-0.05+0.001*seq(1,100)
> prior<-rep(2.0,100)
> logprior<-log(prior)
> loglik<-115*log(lambda)-1200*lambda
> logpos<-logprior+loglik
> logpos<-logpos-max(logpos)
> posterior<-exp(logpos)
> density<-posterior/(sum(posterior)*0.001)
> plot(lambda,density,type="l")
> lines(lambda,prior)
```

4. We can, of course, calculate the statistics we need for the likelihood from the raw data. In this case it is actually just the number of vehicles which passed. However we can also draw a histogram of the data, for example. First you should obtain a copy of the file containing the raw data. The data are available in the file `chester.dat` which can be obtained from my “Additional Teaching Information” Web Page.

R

```
> headway<-read.table("chester.dat")
> headway<-headway[,1]
> n<-length(headway)
> n
[1] 115
> hist(headway)
```

4 Practical: More Than One Parameter and Normal Distribution

4.1 Introduction

In this practical we are going to look at some calculations of posterior distributions where there is more than one parameter and where it is necessary to use numerical integration.

4.2 Normal distribution with unknown mean and variance, done numerically

Here we illustrate a two-parameter case analysed numerically so that we are not restricted to using the conjugate prior distribution.

Use as data the following 30 observations from a normal distribution. Enter these into a vector `y`. You can do this by typing the data into a single column in a text file, called, for example, `example.dat`. Then, in R, type.

```
ytable<-read.table("example.dat")
y<-ytable[,1]
```

```
21.3 19.9 22.6 22.7 18.8 19.5
18.1 23.0 18.7 26.3 23.2 23.4
26.1 24.7 20.0 25.4 18.6 25.2
24.2 23.3 22.9 25.0 22.9 23.8
20.2 21.4 22.9 19.5 24.7 26.4
```

Suppose the prior distribution for $p = 1/\sigma^2$, where σ^2 is the population variance, is negative exponential with mean 0.1.

Suppose the prior distribution for the population mean μ is normal with mean 20 and variance 16, independently of p .

Now you can use the procedure in Section 5.2 of the lecture notes. In fact you might like to define a R function so that you can change the range of values used without having to type everything again.

Alternatively, we can use a purely numerical procedure as follows.

Set up a grid of μ and p values, with μ between 21 and 24 in steps of 0.05 and p between 0 and 0.3 in steps of 0.005, as follows.

```
muvals<-seq(21,24,0.05)
> muvals
[1] 21.00 21.05 21.10 21.15 21.20 21.25 21.30 21.35 21.40 21.45 21.50 21.55
[13] 21.60 21.65 21.70 21.75 21.80 21.85 21.90 21.95 22.00 22.05 22.10 22.15
[25] 22.20 22.25 22.30 22.35 22.40 22.45 22.50 22.55 22.60 22.65 22.70 22.75
[37] 22.80 22.85 22.90 22.95 23.00 23.05 23.10 23.15 23.20 23.25 23.30 23.35
[49] 23.40 23.45 23.50 23.55 23.60 23.65 23.70 23.75 23.80 23.85 23.90 23.95
[61] 24.00
> mu<-rep(muvals,61)
> x<-rep(61,61)
> pvals<-seq(0.0,0.3,0.005)
> pvals
[1] 0.000 0.005 0.010 0.015 0.020 0.025 0.030 0.035 0.040 0.045 0.050 0.055
[13] 0.060 0.065 0.070 0.075 0.080 0.085 0.090 0.095 0.100 0.105 0.110 0.115
[25] 0.120 0.125 0.130 0.135 0.140 0.145 0.150 0.155 0.160 0.165 0.170 0.175
[37] 0.180 0.185 0.190 0.195 0.200 0.205 0.210 0.215 0.220 0.225 0.230 0.235
[49] 0.240 0.245 0.250 0.255 0.260 0.265 0.270 0.275 0.280 0.285 0.290 0.295
[61] 0.300
> p<-rep(pvals,x)
```

Now calculate the natural logarithm of the joint prior density as follows.

```
> muprior<-dnorm(mu,20,4)
> pprior<-dexp(p,10)
> logprior<-log(muprior)+log(pprior)
```

Now the likelihood function is

$$\begin{aligned} L &= \prod_{i=1}^n (2\pi\sigma^2)^{-1/2} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \mu)^2\right\} \\ &= (2\pi\sigma^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - \mu)^2\right\} \end{aligned}$$

and the log likelihood is

$$l = -(n/2) \ln(2\pi) + (n/2) \ln(p) - (1/2)p(\sum y_i^2 - 2\mu \sum y_i + n\mu^2).$$

So we calculate (up to a constant) the log posterior as follows.

Age-group	Mid-point	Patients	Deaths
1	10	4272	9
2	20	2835	23
3	30	2785	19
4	40	1930	16
5	50	1497	59
6	60	960	101
7	70	652	185
8	80	186	97

Table 1: Deaths in surgery

```

> n<-length(y)
> sy<-sum(y)
> syy<-sum(y*y)
> loglik<-(n/2)*log(p) - (p/2)*(syy - 2*mu*sy + n*mu*mu)
> logpos<-logprior+loglik

```

Before exponentiating (i.e. taking antilogs) we add a constant to rescale so that the calculations do not overflow.

```

> logpos<-logpos-max(logpos)
> posterior<-exp(logpos)

```

We now integrate numerically to find the missing constant of proportionality and rescale the posterior density. This can now be displayed, for example, in a contour plot.

```

> k<-sum(posterior)*0.05*0.005
> posterior<-posterior/k
> dim(posterior)<-c(61,61)
> contour(muvals,pvals,posterior)
> persp(muvals,pvals,posterior)
> image(muvals,pvals,posterior)

```

To calculate the marginal posterior density of μ or p it is convenient to rearrange the array so that we can integrate along the rows or columns. In fact we already did this to make the 3-d plots.

```

> mupost<-rowSums(posterior)*0.005
> plot(muvals,mupost,type="l")
> ppost<-colSums(posterior)*0.05
> plot(pvals,ppost,type="l")

```

We can use the marginal density to calculate the posterior mean of μ .

```

> mumupost<-muvals*mupost
> pmmu<-sum(mumupost)*0.05
> pmmu
[1] 22.45547

```

4.3 Logistic regression

Table 1 gives a subset of some data given by Mosteller and Tukey (1977). The data refer to male patients undergoing surgery at a particular location in the USA. For each of a number of age groups, the total number undergoing surgery and the number of these who died are given. The mid-point of the age-range is given for each group.

Suppose that we adopt the following model. Let the mid-point for age-group i be x_i . Let the number undergoing surgery be n_i and the number of deaths be d_i in group i . Then we propose that d_i is an observation from the binomial(n_i, p_i) distribution. Furthermore, we propose that

$$\log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta x_i,$$

where x_i is the age-group mid-point minus 40, so that the probability of death depends on age. This is called a *logistic regression*. The reason why we subtract 40 is simply because it is probably easier to construct a prior by thinking about the middle of the age range than one of the ends. In this way α refers to the 40-year-old group.

Suppose that we give α a $N(-3.0, 1.0)$ prior distribution and β , independently, a $N(0.0, 1.0)$ prior distribution.

The likelihood is proportional to

$$\begin{aligned} L &= \prod_{i=1}^8 p_i^{d_i} (1-p_i)^{n_i-d_i} \\ &= \prod_{i=1}^8 \left(\frac{p_i}{1-p_i}\right)^{d_i} (1-p_i)^{n_i} \end{aligned}$$

Therefore, apart from an additive constant, the log likelihood is

$$\sum_{i=1}^8 d_i[\alpha + \beta x_i] + n_i \log(1-p_i)$$

and $\log(1-p_i) = -\log[1 + \exp(\alpha + \beta x_i)]$.

The data are in a file on the computer system. Obtain a copy of the file `surgery.dat` from my “Additional Teaching Information” Web Page.

Start R then read the data and subtract 40 from the ages as follows.

```
data<-read.table("surgery.dat")
data[,1]<-data[,1]-40
```

We will define some functions to save having to type things more than once.

```
> valgen<-function(lower,upper,nstep)
+ {step<-(upper-lower)/nstep
+ values<-seq(lower,upper,step)
+ values
+ }
> alpha<-valgen(-5,-2,100)
> beta<-valgen(0,0.2,100)
```

Define some vectors containing the prior means and variances.

```
> aprior<-c(-3,1)
> bprior<-c(0,1)
```

Now define a function to calculate the posterior density.

```
> logreg<-function(alpha,beta,data,aprior,bprior)
+ {astep<-alpha[2]-alpha[1]
+ bstep<-beta[2]-beta[1]
+ n<-data[,2]
+ y<-data[,3]
+ x<-data[,1]
+ na<-length(alpha)
+ nb<-length(beta)
+ nd<-length(n)}
```

```

+ amat<-rep(alpha,nb)
+ bmat<-rep(beta,na)
+ amat<-matrix(amat,nrow=na)
+ bmat<-matrix(bmat,nrow=na,byrow=T)
+ sda<-sqrt(aprior[2])
+ sdb<-sqrt(bprior[2])
+ logprior<-log(dnorm(amat,aprior[1],sda))+log(dnorm(bmat,bprior[1],sdb))
+ logpos<-logprior
+ for (i in 1:nd)
+ {q<-1+exp(amat+bmat*x[i])
+ logpos<-logpos+y[i]*(amat+bmat*x[i]) - n[i]*log(q)
+ }
+ logpos<-logpos-max(logpos)
+ posterior<-exp(logpos)
+ int<-sum(posterior)*astep*bstep
+ posterior<-posterior/int
+ posterior
+ }
> posterior<-logreg(alpha,beta,data,aprior,bprior)
> contour(alpha,beta,posterior)

```

We see that our ranges are too wide so we change them.

```

> alpha<-valgen(-4.5,-3.5,100)
> beta<-valgen(0.07,0.12,100)
> posterior<-logreg(alpha,beta,data,aprior,bprior)
> contour(alpha,beta,posterior)
> persp(alpha,beta,posterior)

```

You might try further refinement of the range.

Now we could calculate, for example, the predictive probability of death for a patient aged, say 48. Perhaps you might like to try this.

4.4 Telephone queues

Try solving Inference Problems number 1.

5 Practical: An Old Miniproject

5.1 Miniproject 1994

What follows is the 1993-4 miniproject from the module “Modelling with Probability” in the M.Sc. Decision Support Systems course. See how far you can get with this. The data are in the file `assign94.dat`. You can obtain a copy of this file from my “Additional Teaching Information” Web Page.

You should be able to do the computations and plotting in R. It is probably best to define functions so that you can change the ranges of integration and plotting without having to type everything again.

Task 3 is actually a discrimination problem and we will be looking at such problems in a later lecture but you may be able to figure out what to do here.

Please ask for help if you need it.

5.2 Background

Components supplied to a company have lifetimes, in hours, which follow a Weibull distribution with probability density function

$$f(t) = \alpha\rho(\rho t)^{\alpha-1} \exp\{-(\rho t)^\alpha\}$$

If the supplier uses the correct manufacturing method then the lifetimes are relatively long. However suppliers sometimes use a cheaper method which tends to lead to shorter lifetimes.

Samples of 50 components made using each method were obtained and tested.

5.3 Prior beliefs

Suppose that when the current method is used the parameters are α_1, ρ_1 and when the cheap method is used the parameters are α_2, ρ_2 . The prior distributions for α_1 and α_2 are independent and negative exponential with mean 1.0. The prior distributions for ρ_1 and ρ_2 are independent of those for α_1 and α_2 and of each other and are $\text{gamma}(3,1000)$ for ρ_1 and $\text{gamma}(3,800)$ for ρ_2 (in the notation used in the course notes).

5.4 Tasks

1. Find and display the posterior distributions of the parameters.
(40 marks)
2. Find the predictive distribution of the lifetime of a new component
 - (a) given that it was manufactured correctly,
 - (b) given that it was manufactured by the cheap method.
 (20 marks)
3. (a) A new component is tested and lasts for t hours before failing. We are not sure whether the correct method was used in its manufacture. Investigate how the posterior probability that it was manufactured correctly depends on t and on the prior probability of correct manufacture.
 - (b) Repeat this investigation with the difference that the new component is still working after t hours.
 - (c) Illustrate how these results are affected if we test several new components instead of just one.

(20 marks)

5.5 Reports and Marking

You should discuss and explain your work carefully and illustrate your report with suitable graphs. The marks available are divided equally between:

- Method
- Discussion of method
- Results
- Conclusions
- Presentation

5.6 Data

Fifty correctly manufactured components failed after the following numbers of hours.

67	313	1391	630	627	573	2093	28	492	482
206	1166	165	1088	496	313	437	815	436	17
32	131	340	939	247	1859	57	132	813	254
950	1615	463	258	2285	672	506	50	637	246
178	431	306	662	33	254	858	187	344	545

Fifty cheaply manufactured components failed after the following numbers of hours.

74	239	565	840	793	97	128	35	71	40
63	359	783	4	120	1004	521	902	129	569
323	194	109	145	1292	147	323	57	517	480
210	145	30	320	77	187	157	108	439	365
289	1958	277	586	102	973	65	466	729	73

6 BUGS

6.1 A first BUGS program

Let us do the logistic regression example of Section 4.3 in Practical 4 using BUGS. (It is not really necessary to use BUGS to do this but doing so provides a simple first example).

1. Create a file called `surgery.bug` and type the following BUGS model into it.

```
model surgery;

const N=8;

var y[N],age[N],patients[N],p[N],alpha,beta;

data age,patients,y in "surgery.dat";

{for (i in 1:N)
  {y[i]~dbin(p[i],patients[i]);
   logit(p[i])<-alpha+beta*(age[i]-40);
  }

alpha~dnorm(-3.0,1.0);
beta~dnorm(0.0,1.0);

}
```

2. Run Bugs and enter the following commands.

- (a) Compile the model by typing:

```
compile("surgery.bug")
```

- (b) Execute 1000 iterations as a “burn-in.”

```
update(1000)
```

- (c) Monitor α and β and execute 5000 more iterations.

```
monitor(alpha)
monitor(beta)
update(5000)
```

- (d) Look at a summary of the results so far and leave BUGS.

```
stats(alpha)
stats(beta)
q()
```

3. Now you should have a file called `bugs.out` which will contain the output of your monitored iterations. There will be two columns. The first half of each column refers to α and the second half to β . Column 1 gives the iteration number and Column 2 the sampled value. It is not difficult to read these into R and look at the results.

- (a) Read the data and rearrange them.

```
> bugsout<-read.table("bugs.out")
> samples<-bugsout[,2]
> iteration<-bugsout[,1]
> samples<-matrix(samples, ncol=2)
> iteration<-matrix(iteration, ncol=2)
> alpha<-samples[,1]
> beta<-samples[,2]
> iteration<-iteration[,1]
```

- (b) Check convergence by plotting the sampled values against the iteration numbers. Also plot α against β to get an idea of the bivariate posterior distribution.

```
> plot(iteration,alpha, type="l")
> plot(iteration, beta, type="l")
> plot(alpha,beta)
```

We see that the “trace” plots suggest that the chain has converged but there is a suggestion of slow “mixing”. This might be caused by the negative correlation between α and β which can be seen in the bivariate scatter plot. (This is despite our subtraction of 40 from the ages).

- (c) Plot an approximation to the posterior marginal densities by applying a density estimator.

```
> plot(density(alpha))
> plot(density(beta))
```

You may feel that these densities look odd in some way. This may be because we need to modify the density estimator or it may be that we need to run the Gibbs sampler for longer (because of the poor mixing).

- (d) We can add the prior densities to the marginal posterior plots.

```
> plot(density(alpha))
> lines(alpha,dnorm(alpha,-3,1))
> plot(density(beta))
> lines(beta,dnorm(beta,0,1))
```

We see that the priors are very flat compared to the posteriors.

4. We might want to find the predictive probability that a new patient with a particular age, say 48, will die in surgery.

- (a) Modify the BUGS model as follows.

```
model surgery;

const N=8;

var y[N],age[N],patients[N],p[N],alpha,beta,pred;

data age,patients,y in "surgery.dat";

{for (i in 1:N)
  {y[i]~dbin(p[i],patients[i]);
```

```

    logit(p[i])<-alpha+beta*(age[i]-40);
  }

  alpha~dnorm(-3.0,1.0);
  beta~dnorm(0.0,1.0);

  logit(pred)<-alpha+beta*8;

}

```

- (b) Run BUGS as before except that this time you should monitor `pred` rather than α and β .

```

compile("surgery.dat")
update(1000)
monitor(pred)
update(5000)
stats(pred)

```

The predictive probability is the posterior mean of `pred`, i.e. about 0.04.

6.2 Further BUGS exercises

The BUGS Web Site provides documentation, including a manual and a large set of examples. If you have difficulty obtaining any of these things, please let me know. The documentation can be obtained from:

<http://www.mrc-bsu.cam.ac.uk/bugs/documentation/contents.shtml>

The files for the examples can be obtained from:

<http://www.mrc-bsu.cam.ac.uk/bugs/examples/contents.shtml>

Try working your way through the BUGS examples. You will probably not want to do all of them but do as many as you like. Please ask for assistance if you need it. Those of you who are working on survival analysis might be particularly interested in the following three examples in Volume 1.

16. Mice. Weibull regression in censored survival analysis.

17. Kidney. Weibull regression with random effects.

18. Leuk. Survival analysis using Cox regression.

However you might like to try some of the easier examples first.

7 Discrimination and Classification: The Iris Data

7.1 Introduction

The data on specimens of three species of iris, given in Table 1.1, are taken from Fisher (1936). They seem to be a popular example.

The data are in the file `irises.dat` which is available from my “Additional Teaching Information” Web Page.

7.2 “Plug-in” analysis

We could do a fully Bayesian discriminant analysis. The conjugate prior is a little artificial and, if we use a non-conjugate prior, we need to use numerical methods, e.g. MCMC using BUGS. First we will look at the “plug-in” method which can be computed in R.

Iris Setosa				Iris versicolor				Iris virginica			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5.0	2.0	3.5	1.0	6.5	3.2	5.1	2.0
4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3.0	1.4	0.1	6.0	2.2	4.0	1.0	6.8	3.0	5.5	2.1
4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4
5.7	4.4	1.5	0.4	6.7	3.1	4.4	1.4	6.4	3.2	5.3	2.3
5.4	3.9	1.3	0.4	5.6	3.0	4.5	1.5	6.5	3.0	5.5	1.8
5.1	3.5	1.4	0.3	5.8	2.7	4.1	1.0	7.7	3.8	6.7	2.2
5.7	3.8	1.7	0.3	6.2	2.2	4.5	1.5	7.7	2.6	6.9	2.3
5.1	3.8	1.5	0.3	5.6	2.5	3.9	1.1	6.0	2.2	5.0	1.5
5.4	3.4	1.7	0.2	5.9	3.2	4.8	1.8	6.9	3.2	5.7	2.3
5.1	3.7	1.5	0.4	6.1	2.8	4.0	1.3	5.6	2.8	4.9	2.0
4.6	3.6	1.0	0.2	6.3	2.5	4.9	1.5	7.7	2.8	6.7	2.0
5.1	3.3	1.7	0.5	6.1	2.8	4.7	1.2	6.3	2.7	4.9	1.8
4.8	3.4	1.9	0.2	6.4	2.9	4.3	1.3	6.7	3.3	5.7	2.1
5.0	3.0	1.6	0.2	6.6	3.0	4.4	1.4	7.2	3.2	6.0	1.8
5.0	3.4	1.6	0.4	6.8	2.8	4.8	1.4	6.2	2.8	4.8	1.8
5.2	3.5	1.5	0.2	6.7	3.0	5.0	1.7	6.1	3.0	4.9	1.8
5.2	3.4	1.4	0.2	6.0	2.9	4.5	1.5	6.4	2.8	5.6	2.1
4.7	3.2	1.6	0.2	5.7	2.6	3.5	1.0	7.2	3.0	5.8	1.6
4.8	3.1	1.6	0.2	5.5	2.4	3.8	1.1	7.4	2.8	6.1	1.9
5.4	3.4	1.5	0.4	5.5	2.4	3.7	1.0	7.9	3.8	6.4	2.0
5.2	4.1	1.5	0.1	5.8	2.7	3.9	1.2	6.4	2.8	5.6	2.2
5.5	4.2	1.4	0.2	6.0	2.7	5.1	1.6	6.3	2.8	5.1	1.5
4.9	3.1	1.5	0.2	5.4	3.0	4.5	1.5	6.1	2.6	5.6	1.4
5.0	3.2	1.2	0.2	6.0	3.4	4.5	1.6	7.7	3.0	6.1	2.3
5.5	3.5	1.3	0.2	6.7	3.1	4.7	1.5	6.3	3.4	5.6	2.4
4.9	3.6	1.4	0.1	6.3	2.3	4.4	1.3	6.4	3.1	5.5	1.8
4.4	3.0	1.3	0.2	5.6	3.0	4.1	1.3	6.0	3.0	4.8	1.8
5.1	3.4	1.5	0.2	5.5	2.5	4.0	1.3	6.9	3.1	5.4	2.1
5.0	3.5	1.3	0.3	5.5	2.6	4.4	1.2	6.7	3.1	5.6	2.4
4.5	2.3	1.3	0.3	6.1	3.0	4.6	1.4	6.9	3.1	5.1	2.3
4.4	3.2	1.3	0.2	5.8	2.6	4.0	1.2	5.8	2.7	5.1	1.9
5.0	3.5	1.6	0.6	5.0	2.3	3.3	1.0	6.8	3.2	5.9	2.3
5.1	3.8	1.9	0.4	5.6	2.7	4.2	1.3	6.7	3.3	5.7	2.5
4.8	3.0	1.4	0.3	5.7	3.0	4.2	1.2	6.7	3.0	5.2	2.3
5.1	3.8	1.6	0.2	5.7	2.9	4.2	1.3	6.3	2.5	5.0	1.9
4.6	3.2	1.4	0.2	6.2	2.9	4.3	1.3	6.5	3.0	5.2	2.0
5.3	3.7	1.5	0.2	5.1	2.5	3.0	1.1	6.2	3.4	5.4	2.3
5.0	3.3	1.4	0.2	5.7	2.8	4.1	1.3	5.9	3.0	5.1	1.8

Table 2: Iris Data

7.2.1 Linear discriminant

1. Read the data.

```
> irises<-read.table("irises.dat")
```

2. Calculate (non-Bayesian) point estimates of the means for each variety.

```
> means<-mean(irises)
> setosa<-means[1:4]
> versicolor<-means[5:8]
> virginica<-means[9:12]
> meantab<-data.frame(setosa,versicolor,virginica)
```

3. Calculate (non-Bayesian) point estimates of the variances and covariances.

- (a) First we need to eliminate the differences in means between the varieties.

```
nrows<-length(irises[,1])
adjust<-rep(means,rep(nrows,12))
adjust<-array(adjust,c(nrows,12))
irisadj<-irises-adjust
```

- (b) Next we extract the four variables.

```
> seplen<-c(irisadj[,1],irisadj[,5],irisadj[,9])
> sepwid<-c(irisadj[,2],irisadj[,6],irisadj[,10])
> petlen<-c(irisadj[,3],irisadj[,7],irisadj[,11])
> petwid<-c(irisadj[,4],irisadj[,8],irisadj[,12])
> longvars<-data.frame(seplen,sepwid,petlen,petwid)
```

- (c) Finally we calculate the estimated variance matrix.

```
> varmat<-var(longvars)
```

4. We can now calculate likelihoods for the three varieties, assuming that our point estimates can be treated as the “true” values, multiply these by our prior probabilities for the varieties and hence calculate posterior probabilities for the groups. This is best done using a function such as the following.

```
> lindprob<-function(new,means,variance,prior)
+ {ngroup<-length(means)
+   prob<-1:ngroup
+   for (i in 1:ngroup)
+     {diff<-new-means[,i]
+     prob[i]<- -(diff%*%solve(variance,diff))/2
+     }
+   adj<-max(prob)
+   prob<-prior*exp(prob-adj)
+   prob<-prob/sum(prob)
+   prob
+ }
```

5. We can try this out by calculating the posterior probabilities for the first specimen in the data. We also need to set up our prior probabilities. These need not sum to 1. We just need proportionality.

```
> prior<-c(1,1,1)
> test<-c(5.1,3.5,1.4,0.2)
> lindprob(test,meantab,varmat,prior)
[1] 0.83193835 0.12400496 0.04405669
```

We see a posterior probability (with equal prior probabilities) of 0.832 that this specimen is *Setosa*, which is, in fact, what it is.

We could try testing all of the specimens in the data set, perhaps by writing a function, which would call the function `lindprob`, to do this.

7.2.2 Quadratic discriminant (R version)

1. Read the data (if necessary).

```
> irises<-read.table("/home/cs0mfa/data/datasets/irises.dat")
```

2. Collect the data into the three varieties.

```
> v1<-data.frame(irises[,1:4])
> v2<-data.frame(irises[,5:8])
> v3<-data.frame(irises[,9:12])
```

3. Calculate (non-Bayesian) point estimates of the variances and covariances.

```
> varmat1<-var(v1)
> varmat2<-var(v2)
> varmat3<-var(v3)
> variances<-list(varmat1,varmat2,varmat3)
```

4. Calculate (non-Bayesian) point estimates of the means for each variety (if necessary).

```
> means<-mean(irises)
> setosa<-means[1:4]
> versicolor<-means[5:8]
> virginica<-means[9:12]
> meantab<-data.frame(setosa,versicolor, virginica)
```

5. We can now calculate likelihoods for the three varieties, assuming that our point estimates can be treated as the “true” values, multiply these by our prior probabilities for the varieties and hence calculate posterior probabilities for the groups. This is best done using a function such as the following. (You can easily make this by editing `lindprob`. There is only a slight change).

```
> quaddprob<-function(new,means,variance,prior)
+ {ngroup<-length(means)
+   prob<-1:ngroup
+   for (i in 1:ngroup)
+     {diff<-new-means[,i]
+       prob[i]<- -(log(det(variance))+diff%*%solve(variance[[i]],diff))/2
+     }
+   adj<-max(prob)
+   prob<-prior*exp(prob-adj)
+   prob<-prob/sum(prob)
+   prob
+ }
```

6. We can try this out by calculating the posterior probabilities for the first specimen in the data. We also need to set up our prior probabilities. These need not sum to 1. We just need proportionality.

```

> prior<-c(1,1,1)
> test<-c(5.1,3.5,1.4,0.2)
> quadprob(test,meantab,variance,prior)
[1] 1.000000e+00 4.918517e-26 2.981541e-41

```

We see that (with equal prior probabilities) we are virtually certain that this specimen is *Setosa*, which is, in fact, what it is.

We could try testing all of the specimens in the data set, perhaps by writing a function, which would call the function `quadprob`, to do this.

7.3 Fully Bayesian quadratic discriminant (BUGS)

Now we will turn to a fully Bayesian analysis using BUGS. The following BUGS model specification fits a multivariate normal distribution for each variety. The mean vector and the variance matrix both vary between varieties. The multivariate normal distribution is built step-by-step, using the `k[,]` coefficients to generate the covariances.

The data, in a format suitable for BUGS, are in the file `iris.dat` which is available from my “Additional Teaching Information” Web Page. As well as reading the training data from `iris.dat`, BUGS will read the measurements for a new specimen from `iristest.dat`. I suggest that you try putting the numbers

```

5.9
3.0
5.1
1.8

```

in a single column in this file. It also reads the prior probabilities for the variety of the new specimen from `irisprior.dat`. In fact, the values in this file do not need to sum to 1 as they will be rescaled. They should not be negative though. For equal prior probabilities, put the numbers

```

1.0
1.0
1.0

```

in a single column in this file.

If you monitor `testcount` and obtain the posterior mean of this vector then this gives the posterior probabilities for the variety of the test specimen.

```

model iris;

const N=150;

var sn[N],variety[N],y[N,4],m[3,4],tau[3,4],k[3,6],mean[N,4],
    musl,musw,mupl,mupw,
    test[4],prior[3],p[3],testvar,testmean[4],testcount[3];

data sn,variety,y in "iris.dat", test in "iristest.dat",
    prior in "irisprior.dat";

{for (i in 1:N)
  {for (j in 1:4)
    {y[i,j]~dnorm(mean[i,j],tau[variety[i],j]);
    }
  mean[i,1]<-m[variety[i],1];
  mean[i,2]<-m[variety[i],2]+k[variety[i],1]*(y[i,1]-mean[i,1]);
  mean[i,3]<-m[variety[i],3]+k[variety[i],2]*(y[i,1]-mean[i,1])
    +k[variety[i],3]*(y[i,2]-mean[i,2]);
  }
}

```

```

    mean[i,4]<-m[variety[i],4]+k[variety[i],4]*(y[i,1]-mean[i,1])
    +k[variety[i],5]*(y[i,2]-mean[i,2])
    +k[variety[i],6]*(y[i,3]-mean[i,3]);
  }

for (j in 1:3)
  {m[j,1]~dnorm(musl,0.2);
  m[j,2]~dnorm(musw,1.0);
  m[j,3]~dnorm(mupl,0.1);
  m[j,4]~dnorm(mupw,1.0);
  tau[j,1]~dgamma(1,1);
  tau[j,2]~dgamma(1,1);
  tau[j,3]~dgamma(1,1);
  tau[j,4]~dgamma(1,0.5);
  for (i in 1:6)
    {k[j,i]~dnorm(0,0.01);
    }
  }

musl~dnorm(6,0.1);
musw~dnorm(3,0.1);
mupl~dnorm(4,0.1);
mupw~dnorm(1,0.1);

# Try the test specimen.

for (i in 1:3)
  {p[i]<-prior[i]/sum(prior[]);
  testcount[i]<-equals(i,testvar);
  }

testvar~dcat(p[]);

for (j in 1:4)
  {test[j]~dnorm(testmean[j],tau[testvar,j]);
  }
  testmean[1]<-m[testvar,1];
  testmean[2]<-m[testvar,2]+k[testvar,1]*(test[1]-testmean[1]);
  testmean[3]<-m[testvar,3]+k[testvar,2]*(test[1]-testmean[1])
  +k[testvar,3]*(test[2]-testmean[2]);
  testmean[4]<-m[testvar,4]+k[testvar,4]*(test[1]-testmean[1])
  +k[testvar,5]*(test[2]-testmean[2])
  +k[testvar,6]*(test[3]-testmean[3]);
  }
}

```

In fact BUGS allows direct specification of a multivariate normal distribution and a second model specification is given below showing its use. Unfortunately this one does not work. The multivariate normal specification sometimes leads to difficulties, at least in this version of BUGS. If you can get this to work then please let me know!

```

model iris;
# Not working properly. I don't know why. 14/10/03.

const N=150;

```



```

var sn[N],variety[N],y[N,4],m[3,4],T[3,4,4],R[4,4],
    musl,musw,mupl,mupw;

data sn,variety,y in "iris.dat", R in "iriscov.dat";

{for (i in 1:N)
  {y[i,]~dmnorm(m[variety[i],],T[variety[i],,]);
  }

for (j in 1:3)
  {m[j,1]~dnorm(musl,0.2);
  m[j,2]~dnorm(musw,1.0);
  m[j,3]~dnorm(mupl,0.1);
  m[j,4]~dnorm(mupw,1.0);
  T[j,,]~dwish(R[,],5);
  }

musl~dnorm(6,0.1);
musw~dnorm(3,0.1);
mupl~dnorm(4,0.1);
mupw~dnorm(1,0.1);

}

```

8 Practical: Road Vehicle Headways

This is an example of using MCMC when there are many unknowns. The data are taken from Cowburn (2003) and the model and analysis are based on work in Cowburn (2003) and Cowburn and Farrow (2003).

We are modelling the times, known as “headways”, between successive vehicles travelling in the same direction along a road. The basic idea is that some vehicles are unaffected by the vehicle in front (i.e. they are “free-flowing”) while others are constrained by the vehicle in front (i.e. they are “congested”). The overall distribution of headways is therefore a *mixture* distribution. There is a probability p that a particular vehicle will be free-flowing and a probability $1 - p$ that it will be congested. The value of p is unknown. Different probability distributions apply to the free-flowing and congested headways. In the model considered here the free-flowing headways are given an exponential distribution with pdf

$$f_1(t) = \beta_1 e^{-\beta_1 t}$$

and the congested headways are given a gamma distribution with pdf

$$f_2(t) = \frac{\beta_2^{\alpha_2} t^{\alpha_2-1} e^{-\beta_2 t}}{\Gamma(\alpha_2)}.$$

In each case the pdf applies for $t \geq 0$.

For the purpose of this exercise, successive headways are independent.

Prior distributions are given to the parameters $p, \beta_1, \beta_2, \alpha_2$. The constraints $\alpha_2 > 1$ and $\beta_2 > \alpha_2 \beta_1$ are applied. The second constraint ensures that the mean headway for congested vehicles is less than the mean headway for free-flowing vehicles. Such constraints help to overcome the problems of *identifiability* and *poor mixing* which may occur in such models. Nevertheless, this is a difficult problem, especially as the data do not exhibit two clear modes. You should use suitable plots to check convergence.

The way that this problem is handled is to define an indicator variable S_i for each observation, such that, if $S_i = 1$, headway i is free-flowing, i.e. pdf $f_1(t)$ applies. If $S_i = 2$, headway i is

congested, i.e. pdf $f_2(t)$ applies. These indicators are unknowns. Thus there are $n + 3$ unknowns for n observations. In this case $n = 150$.

The data are available in the file `dd01.dat` and there is also an initial value file. This is `headway0b.in`. Both of these are available from the “Additional Teaching Information” Web Page.

You may use the following BUGS model specification.

```
model headway;

# This version is for independent headways.
# a[2] has to be greater than 1.

const
N=150;

var t[N], p, aa, a[2], bb, b[2], s[N], sdash[N];

data t in "dd01.dat";
inits in "headway0b.in";

{

for (i in 1:N) {
sdash[i]~dbern(p);
s[i]<-2-sdash[i];
t[i]~dgamma(a[s[i]],b[s[i]]);
}

a[1]<-1;
a[2]<-1+aa;
aa~dgamma(4,1);
b[1]~dgamma(2,8) I(,bb);
b[2]<-a[2]*bb;
bb~dgamma(4,4) I(b[1],);
p~dbeta(1,2);

}
```

Investigate the posterior distribution of the parameters. You may also like to investigate the probabilities that individual headways are “congested.”

9 A simple Bayesian Network, using BUGS

The task here is to create a small Bayesian network for diagnosis of disease. The network will contain some binary variables and some continuous variables. We will use BUGS to evaluate probabilities given various diagnostic information.

Consider a clinic for people suffering from diseases of a particular type. Any person attending this clinic must be suffering from either disease A or disease B. It is so unlikely that someone would have both diseases when they first attend the clinic that this event can be given zero probability.

The clinic has available to it two diagnostic aids as follows.

- A simple test which either gives a positive result or a negative result.
- Measurement of the concentration of a certain protein P in the patient’s blood.

The outcomes of the test and P concentration are conditionally independent of all other variables given the disease.

In addition the patient's age and sex would normally be known.

From examination of records past cases where the correct diagnosis is known we conclude the following about patients who arrive at the clinic.

- The prior probability that a patient has disease A is 0.7.
- The probability that the test result is positive is 0.1 for patients with disease A and 0.8 for patients with disease B.
- The concentration of the protein P, in appropriate units, has a normal distribution. For patients with A the mean is 120 and the variance is 400. For patients with B the mean is 60 and the variance is 400.
- The proportion of patients with A who are male is 0.3. The proportion of patients with B who are male is 0.5.
- The distribution of age depends on the diseases and the sex of the patient. In each case the natural logarithm of age has a normal distribution. The parameters are as follows.

Disease	Sex	Mean	Variance
A	Male	4.0	0.01
A	Female	4.2	0.01
B	Male	3.5	0.04
B	Female	3.6	0.04

1. Draw a directed acyclic graph for this set of beliefs.
2. Specify a model in BUGS. You will need to read data on the test result, P concentration, age and sex from a file.
3. Run BUGS and find the probability of disease A. Try various different input data. You can omit a piece of data by entering "NA" for its value.

References

- Cowburn, G.J., 2003. Bayesian mixture modelling with application to road traffic flow, *Ph.D. Thesis*. University of Sunderland.
- Cowburn, G.J. and Farrow, M., 2003. Mixtures and Hidden Markov Models for Road Traffic Vehicle Headways, *Practical Bayesian Statistics 5 Conference*, Milton Keynes, July 2003.
- Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems, *Ann.Eugenics* **7**, pt.II, 179-188.
- Mosteller, F. and Tukey, J.W., 1977. *Data Analysis and Regression*. Reading, Massachusetts: Addison-Wesley.