

UNIVERSITY OF WOLLONGONG
School of Mathematics and Applied Statistics
STAT902. Advanced Data Analysis

ASSIGNMENT 11

Due: 5:00pm Monday 31st May, 2010.

Local students: please put under lecturer's door.

Remote students: please e-mail or post by this date.

1. Recall the BPD example for Bayesian logistic regression described in the **BUGS preliminaries** of Assignment 8. The files used there, `bpdBayesModel.txt` and `bpdBayes.Rs`, conveyed the essential aspects of modelling and fitting using `BRugs` and `WinBUGS`. At the time of Assignment 8 this simplified version was avoiding cluttering with too many new ideas. However, fancier versions, corresponding to `bpdBayesModelFanc.txt` and `bpdBayesFanc.Rs` on the **Computer Code and Data** page of the course web-site, are better in practice. These new scripts have the following embellishments:

- Better initialisation based on the maximum likelihood estimates of β_0 and β_1 .
 - Longer burn-in period (5000) to lessen the effect of starting values and allow the MCMC schemes more time to converge.
 - Thinning of the post-burn-in samples. Now 5000 iterations are carried out after burn-in, but only every fifth member of the sample is retained. This strategy aims to counteract serial correlation that often occurs with MCMC schemes and makes the samples more independent.
 - The final results are reported in terms of the original `birthweight` units (i.e. grammes) rather than the standardised version `sbirthweight`. This involves (a) standardising the birthweight measurements first, (b) using the standardised birthweight measurements in the MCMC phase, (c) undoing the standardisation in the reporting of results. Note that these steps are quite important in Bayesian regression analysis with diffuse priors, since it makes the results scale invariant. If you didn't do this then the answers would change depending on whether birthweights are recorded in grammes, milligrammes, kilogrammes or tonnes.
- (a) Making sure that the data file `bpd.txt` is available to the current session, run the code in `bpdBayesFanc.Rs` and `bpdBayesModelFanc.txt` and report the results. Include brief comparison with the answers obtained from maximum likelihood fitting of the logistic regression model via the function `glm()` (using the original `birthweight` data).
- (b) Note that `bpdBayesModelFanc.txt` uses the following lines of code to undo the standardisation:

```
beta1 <- sbeta1/sdBirthweight  
beta0 <- sbeta0 - beta1*meanBirthweight
```

Derive the mathematical justification of this code. Work with a general continuous predictor x , where

$$\beta_0 + \beta_1 x_i$$

is the linear predictor in terms of the original units, and $x_i^* = (x_i - \bar{x})/s$ is the standardisation of x (where \bar{x} and s are the sample mean and standard deviation of the x_i 's).

2. The LIDAR data consist of data from an air pollution experiment using the Light Detection and Ranging Technique. The response variable is a measure of the cumulative particle concentration, and the predictor variable is range of the measuring device.

- (a) Make sure that the file `lidar.txt` is available.
(b) Start an R session and issue the following commands to fit the cubic regression model

$$y_i = \alpha_0 + \alpha_1 x_i + \alpha_2 x_i^2 + \alpha_3 x_i^3 + \varepsilon_i,$$

plot its fit, and examine the residuals:

```
lidar <- read.table("lidar.txt",header=TRUE)
x <- lidar$x ; y <- lidar$y
xsq <- x^2 ; xcb <- x^3
fitCubic <- lm(y~x+xsq+xcb)
yHatCubic <- predict(fitCubic)
plot(x,y) ; lines(x,yHatCubic,col="red",lwd=3)
plot(yHatCubic,y-yHatCubic) ; abline(0,0,col="blue",lwd=3)
```

Include the two plots in your submission.

- (c) Issue the command:

```
trLin <- function(x,kappa)
  return((x-kappa)*(x>kappa))
```

to define the truncated line function with a knot at κ :

$$(x - \kappa)_+ = \begin{cases} 0, & x \leq \kappa \\ x - \kappa, & x \geq \kappa. \end{cases}$$

- (d) Then issue the following commands to fit and plot the spline regression model:

$$y_i = \beta_0 + \beta_1 x_i + \beta_{11} (x_i - Q_1)_+ + \beta_{12} (x_i - Q_2)_+ + \beta_{13} (x_i - Q_3)_+ + \varepsilon_i$$

where Q_1, Q_2 and Q_3 are the quartiles of the x_i 's:

```
quats <- quantile(x,c(0.25,0.5,0.75))
fitTLQ <- lm(y~x+trLin(x,quats[1])+trLin(x,quats[2])+trLin(x,quats[3]))
yHatTLQ <- predict(fitTLQ)
plot(x,y) ; lines(x,yHatTLQ,col="red",lwd=3)
plot(yHatTLQ,y-yHatTLQ) ; abline(0,0,col="blue",lwd=3)
```

Include the two plots in your submission.

- (e) Write an R script that uses `lm()` to fit the spline regression model

$$y_i = \beta_0 + \beta_1 x_i + \sum_{j=1}^9 \beta_{1j} (x_i - D_j)_+ + \varepsilon_i$$

where D_1, \dots, D_9 are the deciles of x . The script should then obtain a plot of the data, with the fitted curve added; and then a plot of the residuals against the fitted values. Your submission should include the script, and the two plots.

- (f) Re-do part (e), but this time programming the direct matrix expression:

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

In R this is:

```
decs <- quantile(x,seq(0.1,0.9,by=0.1))
X <- cbind(rep(1,length(y)),x)
splinePart <- outer(x,decs,"-")
splinePart <- splinePart*(splinePart>0)
X <- cbind(X,splinePart)
yHatTLD <- X%%solve(t(X)%*%X,t(X)%*%y)
```

- (g) Now re-do part (f), but this time with

```
Dmat <- diag(c(rep(0,2),rep(1,9)))
print(Dmat)
lambda <- 2000
yHatPS <- X%%solve(t(X)%*%X+lambda*Dmat,t(X)%*%y)
```

You have just programmed a penalised spline fit to the LIDAR data. This is an example of *nonparametric regression*. The residuals should be looking better than the cubic fit, in that there is no strong trend. However, there is still a pronounced funnel pattern due to the constant variance assumption being far from being satisfied for these data.

3. This question concerns the mixed model representation of penalised splines and fitting with mixed model software.

- (a) Make sure that the file `lidar.txt` is available and issue the commands in R:

```
lidar <- read.table("lidar.txt",header=TRUE)
x <- lidar$x ; y <- lidar$y
knots <- quantile(x,seq(0.05,0.95,by=0.05))
X <- cbind(rep(1,length(y)),x)
Z <- outer(x,knots,"-")
Z <- Z*(Z>0)
library(nlme)
group <- rep(1,length(y))
data.fr <- groupedData(y~x|group,data=data.frame(x,y))
fitLMM <- lme(y~-1+X,random=pdIdent(~-1+Z),data=data.fr)
betaHat <- fitLMM$coef$fixed
uHat <- unlist(fitLMM$coef$random)
yHatLMM <- X%%betaHat + Z%%uHat
plot(x,y) ; lines(x,yHatLMM,col="red",lwd=3)
plot(yHatLMM,y-yHatLMM) ; abline(0,0,col="blue",lwd=2)
```

Include the resulting two plots in your submission.

- (b) Issue the commands:

```
sigepsHat <- fitLMM$sigma
siguHat <- sigepsHat*exp(unlist(fitLMM$modelStruct))
lambdaREML <- (sigepsHat/siguHat)^2
print(lambdaREML)
```

Report the result. This is the value of λ chosen via restricted maximum likelihood (REML).

- (c) Issue commands:

```
Cmat <- cbind(X,Z)
Dmat <- diag(c(rep(0,2),rep(1,ncol(Z))))
CTC <- t(Cmat)%*%Cmat
dfREML <- sum(diag(solve(CTC+lambdaREML*Dmat,CTC)))
print(dfREML)
```

Report the result. This is the number of effective degrees of freedom of the fit chosen by REML.

4. Let T_1 , T_2 and T_3 be three functions on $[0, 1]$ given by

$$T_1(x) = 1, \quad T_2(x) = x, \quad T_3(x) = (x - \frac{1}{2})_+.$$

Also, let B_1 , B_2 and B_3 be three functions on $[0, 1]$ given by

$$B_1(x) = (\frac{1}{2} - x)_+, \quad B_2(x) = \frac{1}{2} - |x - \frac{1}{2}|, \quad B_3(x) = (x - \frac{1}{2})_+.$$

- (a) Draw two sets of axes, one underneath the other. Each set of axes should have both x and y ranging between 0 and 1. Sketch T_1 , T_2 and T_3 on the upper set of axes, and B_1 , B_2 and B_3 on the lower set of axes.
- (b) Clearly $B_3 = T_3$. Find expressions for B_1 and B_2 in terms of T_1 , T_2 and T_3 .

Hint: The graphs, rather than algebraic expressions, may be useful. Also, what is $B_1 + B_2 + B_3$?

- (c) Write down the 3×3 matrix \mathbf{L} such that

$$[B_1(x) \ B_2(x) \ B_3(x)] = [T_1(x) \ T_2(x) \ T_3(x)]\mathbf{L}$$

for any $x \in [0, 1]$.

- (d) Find the determinant of \mathbf{L} and establish that \mathbf{L} is invertible. In linear algebra language, this implies that $\{B_1, B_2, B_3\}$ is an alternative basis for the vector space of functions spanned by $\{T_1, T_2, T_3\}$. It is known as the linear *B-spline* basis, and has better numerical properties than the truncated line basis $\{T_1, T_2, T_3\}$.