STAT40850 - Bayesian Analysis

Computer Lab 5 – Advanced use of JAGS

Submission deadline: May 2nd at 6pm.

1 The `step` function

When fitting a change-point model, we often want to write code such as:

```
for(i in 1:k) {
    y[i] ~dnorm(theta[1],tau)
}
for(i in (k+1):N) {
    y[i] ~dnorm(theta[2],tau)
}
```

This code specifies that y will come from a normal distribution with mean θ_1 and precision τ for the first k data points, and from a normal distribution with mean θ_2 and precision τ for the remaining data points. A prior distribution would need to be given for $k, \theta_1, \theta_2, \tau$.

Unfortunately, JAGS does not allow us to include a random quantity in a `for` loop so the above code will fail. However, we can get around this problem by using the `step` function. The step function takes a single argument and returns 1 if the argument is greater than zero, and 0 otherwise. The code above would be re-written as:

```
for(i in 1:N) {
    y[i] ~dnorm(theta[J[i]],tau)
    J[i] = 1 + step(i - k - 0.001)
}
```

Now we have introduced a new variable, J_i which will be 1 for values of i that are less than or equal to k, and 2 for values of i that are greater than k. (Note that the 0.001 is required to allow for the possibility of $k = i$.) The above code will now run when prior distributions are included for $k, \theta_1, \theta_2, \tau$.

Tasks:

- The files `mining.R` and `mining.model` contain code to fit the change-point model to the Mining disasters data we met in Lab 3. Familiarise yourself with the code and run it.
- What is the posterior 95% credible interval for the change-point? (remember $i = 1851$ is the first year of the data) Did the rate go up or down?

2 Predictive distributions

When checking the quality of a fitted model, it is often useful to compare the data we have observed with that predicted by the model. If we have observed data $\boldsymbol{y}$, and we have a posterior distribution $\boldsymbol{\theta}|\boldsymbol{y}$ we can create predicted values $\boldsymbol{y}^*$ from:

$$p(\boldsymbol{y}^*|\boldsymbol{y}) = \int p(\boldsymbol{y}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}$$

This probability distribution (known as the posterior predictive distribution) is the posterior distribution times the likelihood of the predicted data, integrated over the parameter values. Whilst this may appear complicated, we can compute Monte Carlo estimates of $\boldsymbol{y}^*|\boldsymbol{y}$ very simply by simulating parameter values from the posterior (which we have already done), and then simulating values from the likelihood given these new values.

Calculating a predictive distribution is very simple in JAGS, usually requiring just a few extra lines of code. The new values $\boldsymbol{y}^*$ are treated as parameters, and can thus be monitored using the sample monitor tool. They can be given starting values. These can be specified with the other starting values, but as they are often not of primary interest in convergence, it is often more convenient to let JAGS generate initial values.

Tasks:

- The files `ratstumour_pred.R` and `ratstumour_pred.model` contain code to produce predictive distributions for the over-dispersed Binomial data on tumours in rats (now ordered by rate of tumours). Monitor the parameters `ystar`, `pi`, `alpha` and `beta`. Run the model for 10000 iterations with multiple starting values, and generate initial values where appropriate. Remove the first 1000 as burn-in and thin to every 10th iteration. Check that convergence has been obtained. Run more iterations if required.
- Make a scatterplot of `y` versus the mean of `ystar`. If the model is fitting well this should be strongly correlated. Does this seem to be the case?
- The 56th and 70th values of `y` are 5 and 9 respectively. Does the posterior distribution of `ystar` for these values contain the true data values?
- Open the files `putting_pred.R` and `putting_quadratic.model`. This code contains a model for the putting data at various distances. Add code to produce the predicted number of successes at each distance. Does the model fit well?

3 Homework: JAGS Smoothing

We will examine data on simulated motorcycle crashes used to test helmets. `motor.dat` is a series of head acceleration measurements over time. Clearly visible is the initial impact, rebounds, and return to rest.

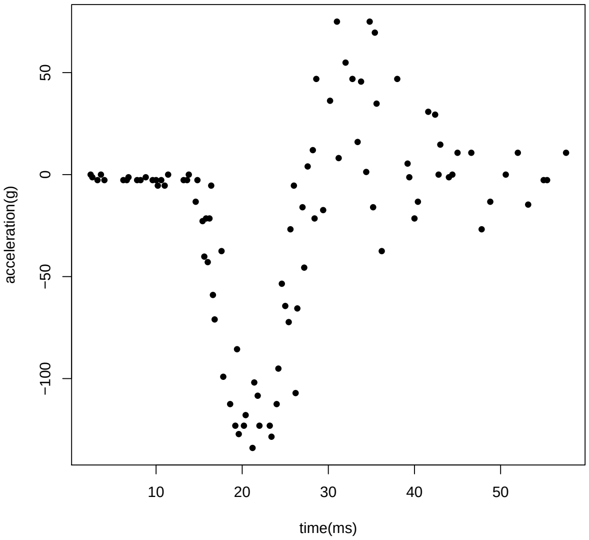

Figure 1: The data.

This follows a smooth curve and the observed accelerations are scattered around this curve. We don't wish to restrict ourselves to parametric (e.g. polynomial) functions of time to model this curve but we want a curve that is smooth over time.

One option is to use *splines*. These are a common choice when modelling smooth functions of unknown form. A spline is a piecewise polynomial function; you will use cubic splines (splines of degree 3) to fit a smooth curve to the above data.

K "knots" are placed at points spread evenly across quantiles of the observed times. A "design" matrix X is then calculated where the entries of X contain the weights for the contribution of each of the K *basis splines* to each observation. i.e. $X[t, k]$ is the weight for basis k on observation t and these sum to one over basis splines. Figure 2 shows 5 basis splines across 5 evenly spread knots on the observed times.

Rather than fit a polynomial (e.g. linear) regression, we will use this X matrix as the spline regression model predictors. As the bases are shared across multiple observations, this will naturally lead to a smooth curve.

Model

The likelihood is that each observed acceleration is Normally distributed around the curve with precision τ

$$\text{accel}_t \sim N(X\beta, \frac{1}{\tau})$$

You should choose the same prior for each of the β coefficients that is Normal with mean 0 and precision λ.

$$\beta_k \sim N(0, \frac{1}{\lambda})$$

λ controls the smoothness of the prediction curve. Finally, specify approximate Jeffrey's priors on τ and λ

$$\tau \sim Ga(0.001, 0.001)$$

$$\lambda \sim Ga(0.001, 0.001)$$

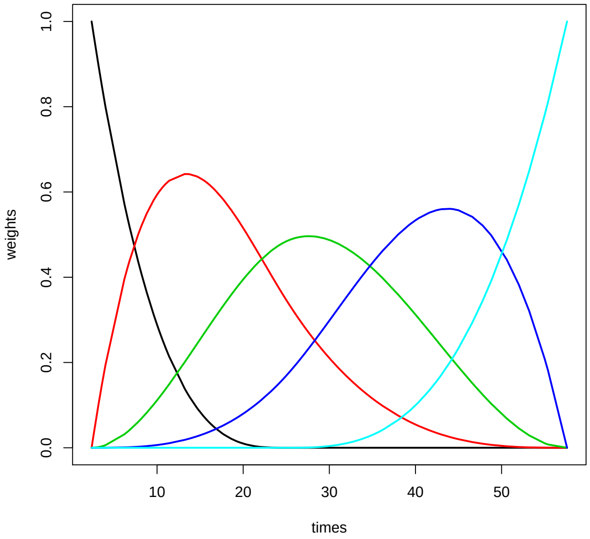

Figure 2: 5 basis splines of degree 3 for the observed times. For an observed time t on the x-axis, the weights $X_{t,k}$ for each basis k are simply the height of basis k at that location.

Tasks:

1. Perform MCMC on the unknown parameters λ, τ and β.
2. Check for convergence.
3. Use DIC to decide the number of basis functions K.
4. For a converged chain from the best model, create a plot that shows the data along with posterior mean and 95% credible intervals for the curve at each observed time i.e. $X\beta$.
5. Similarly, for a converged chain from the best model, create a plot that shows the data along with mean and 95% credible intervals from the posterior predictive distribution for acceleration at each observed time. Compare to the previous plot. Why are the credible intervals wider?
6. Write a *short* report detailing your results.

Hints:

1. You can use the function `bs` in the splines package in `R` to create the required basis functions and design matrix X as follows:
   ```
   library(splines)
   X = bs(motor$times, K, intercept=TRUE)
   ```
2. JAGS can multiply a matrix by a vector; e.g. `X %*% b` matrix multiplies `X` by `b`. Or use a loop.