

# Module 5

An example of what can go wrong with Gibbs and more on diagnostics

Prof. Stephen B. Vardeman  
Statistics and IMSE  
Iowa State University

March 5, 2008

We continue our introduction to Bayes computation with WinBUGS and discussion of critical examination of its output. This will bring us to a further discussion on diagnostics for MCMC aimed at detection of "failure to mix"/"long term dependence upon initializations"/"islands of probability" problems and the parallel issue of "assessing when a simulation has burned in."

## Example 3

As a small fairly innocent-looking example of what can go wrong with a Gibbs sampling Bayes analysis, consider a "mean non-conformities per unit" situation. Below is a series of 10 Poisson observations, the first  $K = 3$  of which were generated using a mean of  $\mu_1 = 2$ , the last  $7 = 10 - K$  of which were generated using a mean of  $\mu_2 = 6$ .

1, 4, 2, 4, 7, 4, 5, 7, 6, 4

Suppose that neither the numbers ( $K$  and  $10 - K$ ) of observations from the two different means, nor the values of those means ( $\mu_1$  and  $\mu_2$ ) were known and one wished to do a Bayes analysis for the 3 parameters  $K$ ,  $\mu_1$ , and  $\mu_2$ . This is a "change-point problem" and the discussion of "where the size of a set is a random quantity" in the Model Specification section of the WinBUGS user manual is relevant to implementing a Bayes analysis of this problem.

## Example 3 (cont.)

The file

BayesASQEx3.odc

contains WinBUGS code for implementing an analysis based a on a model that says *a priori*

$$K \sim \text{Uniform on } \{1, 2, 3, \dots, 10\}$$

independent of

$$\mu_1 \text{ and } \mu_2 \text{ independent Exp}(1)$$

together with 6 different starting vectors for the MCMC. The code is on the next 2 panels.

## Example 3 (cont.)

```
model {  
  for (j in 1:2) {  
    mu[j]~dexp(1)  
  }  
  K~dcat(p[])  
  for (i in 1:10) {  
    ind[i]<- 1+step(i-K-.01)  
    #will be 1 for all i<=K, 2 otherwise  
    y[i]~dpois(mu[ind[i]])  
  }  
}  
  
#here are the data for this problem  
list(p=c(.1,.1,.1,.1,.1,.1,.1,.1,.1,.1),  
     y=c(1,4,2,4,7,4,5,7,6,4))
```

## Example 3 (cont.)

#here are some possible initializations

```
list(K=1,mu=c(1,10))
```

```
list(K=6,mu=c(1,10))
```

```
list(K=10,mu=c(1,10))
```

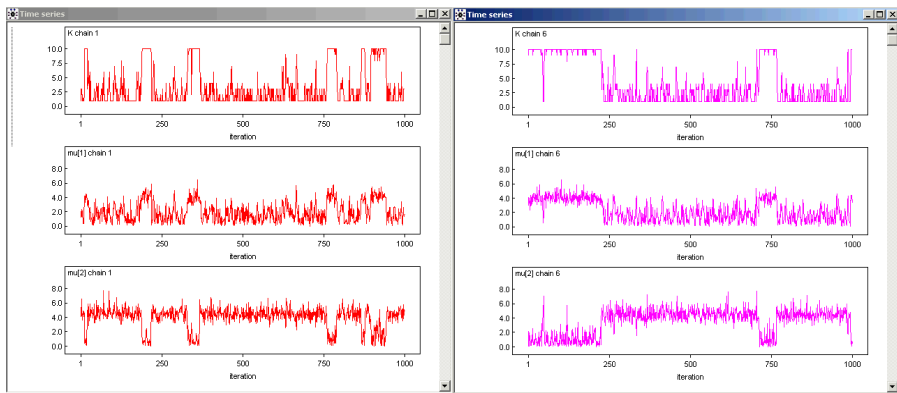
```
list(K=1,mu=c(10,1))
```

```
list(K=6,mu=c(10,1))
```

```
list(K=10,mu=c(10,1))
```

## Example 3 (cont.)

Summary statistics for 1,000,000 iterations from the each of the 6 different starts indicates no "long term" / "huge numbers of iterations" differences in the behaviors of the 6 chains. But consider the summary below of the first 1000 iterations for the first and last initializations.



## Example 3 (cont.)

The early histories of the chains clearly depend upon the starting values, and the step changes in values of parameters indicate sudden movements between fairly isolated parts of the posterior distribution (that are connected by somewhat "narrow" isthmuses). So the related questions of "How big should the burn-in be before we start keeping track of iterations for purposes of doing inference?" and "How can I tell when I have a large enough number of iterations to represent the *whole* posterior?" start to come into focus. Some looking at plots such as these and at summary statistics for parts of the various runs of 1,000,000 iterations makes it clear that discarding the first 100,000 iterations is clearly adequate, and the figure on the next panel is based on 900,000 iterations for each of 6 chains, and probably gives a fair representation of the posterior distribution of the 3 parameters  $K$ ,  $\mu_1$ , and  $\mu_2$ .

## Example 3 (cont.)

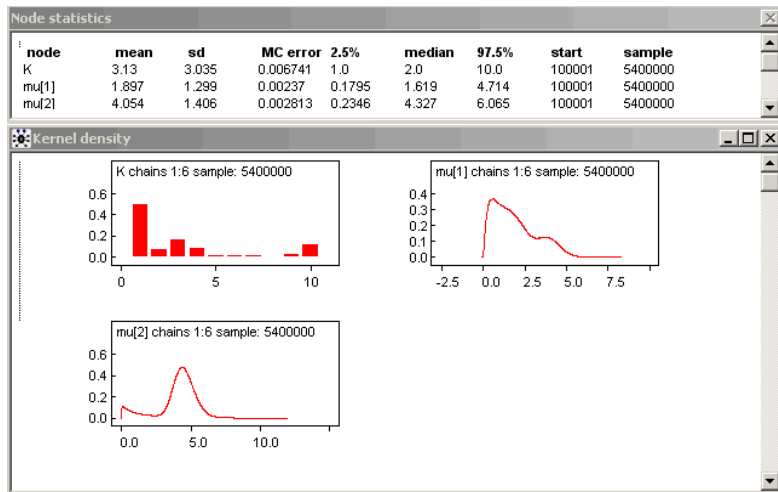


Figure: Summary of the approximate posterior for the first change-point problem

## Example 3 (cont.)

The overall message presented in the figure is that not very much has been learned about the parameters from the 10 observations. (This is consistent with the general intuition that attributes data typically carry relatively little information about process performance, i.e. many of them are usually required to come to any kind of definitive conclusions.) The completely fortuitous event that the smallest observation came first has the effect of putting the biggest posterior probability for  $K$  on the possibility that  $K = 1$ .

## Example 3 (Version 2)

Returning to the main point of this example, for sake of illustration change the data set under discussion to

1, 4, 2, 4, 7, 4, 50, 70, 60, 40

If the code for this problem is modified and again run through 1 million iterations from the earlier 6 different initializations, a very disconcerting picture arises. Through 1,000,000 iterations the effects of the initialization do not "wash out" at all. On the next panel is a table indicating "posterior" means and standard deviations produced from the 6 different initializations.

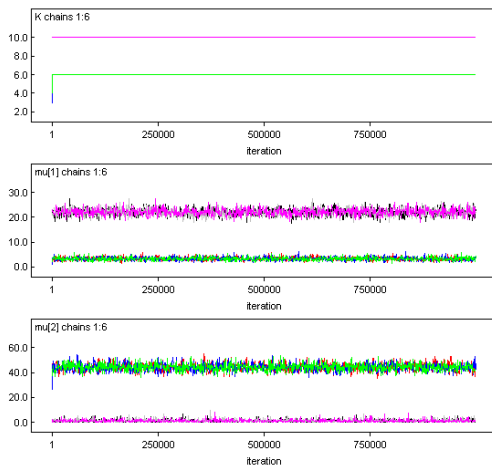
## Example 3 (Version 2 cont.)

**Table 1** "Posterior" Summaries for the Second Change-Point Problem  
(Means Bold, Standard Deviations in Parentheses)

Start	$K$	$\mu_1$	$\mu_2$
<code>list(K=1,mu=c(1,10))</code>	<b>6.0</b> (0.003)	<b>3.285</b> (0.6848)	<b>44.2</b> (2.972)
<code>list(K=6,mu=c(1,10))</code>	<b>6.0</b> (0.003)	<b>3.286</b> (0.6852)	<b>44.2</b> (2.972)
<code>list(K=10,mu=c(1,10))</code>	<b>6.0</b> (0.002)	<b>3.284</b> (0.6851)	<b>44.2</b> (2.973)
<code>list(K=1,mu=c(10,1))</code>	<b>10.0</b> (0.0)	<b>22.09</b> (1.417)	<b>1.001</b> (1.0)
<code>list(K=6,mu=c(10,1))</code>	<b>10.0</b> (0.0)	<b>22.09</b> (1.416)	<b>0.9996</b> (0.9993)
<code>list(K=10,mu=c(10,1))</code>	<b>10.0</b> (0.0)	<b>22.09</b> (1.417)	<b>1.001</b> (1.001)

A corresponding history plot is on the next panel.

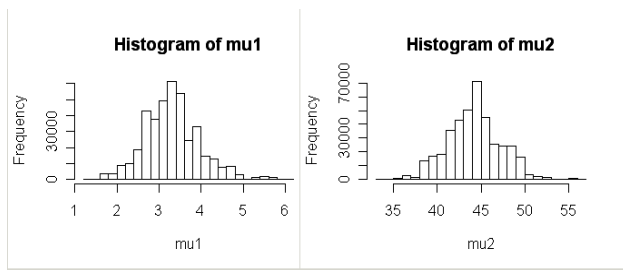
## Example 3 (Version 2 cont.)



It is clear that the posterior has *at least 2* "islands of probability" and that any isthmus between them must be so narrow as to be unlikely to be crossed in 1,000,000 iterations!

## Example 3 (Version 2 cont.)

It is possible to actually do calculus to find the ratio of the posterior probability that  $K = 10$  to the posterior probability that  $K = 6$ . This is about  $7 \times 10^{-49}$ !!! The island of probability corresponding to  $K = 10$  is tiny indeed! In fact, with virtual posterior certainty  $K = 6$  (and NOT 10 or any other value). Further, using a different type of MCMC, Vardeman was able to get an approximate posterior that he believes. This has approximate posteriors for  $\mu_1$  and  $\mu_2$  as pictured below.



## Example 3 (Version 2 cont.)

The posterior Vardeman computed (using a so-called "Metropolis-Hastings" MCMC algorithm) has (respectively) posterior means and standard deviations

3.310537 and 0.6675818 for  $\mu_1$

44.2006 and 3.039628 for  $\mu_2$

Looking again at Table 1, these are like what is obtained from WinBUGS from the first 3 initializations. The troubling thing here is that if one were so foolish as to pick only one starting vector, and so unlucky as to have that turn out to be a "bad" one, one would have a Gibbs sampler that appears to behave nicely, but in reality is completely "wrong" in the sense of fairly representing the posterior.

# Trying to Avoid Disaster Using MCMC

What saved us from disaster in the second change-point example was

- using multiple widely dispersed initializations for the MCMC, and
- looking separately at the results from the multiple chains to see if they are in agreement.

We did the comparison between different chains largely "by eye" looking at Table 1 and the history plots. There are more formal tools that can be used to compare multiple chains looking 1) for problems and 2) for completion of burn-in where the effects of different initializations do not persist (as in the first version of the change-point problem). One that is implemented in `WinBUGS` is due to Gelman and Rubin in its original form and to Brooks and Gelman as actually implemented (hence the name `bgrdiag` ).

## "bgr diagnostic" Plots

The idea is this. Suppose one runs  $m$  parallel chains through  $n$  iterations. For each quantity monitored (like an entry of  $\theta = (\theta_1, \theta_2, \dots, \theta_k)$  or the value of some  $X_{\text{new}}$ ) let

$L_n^j$  = the lower 10% point of the  $n$  values from chain  $j$

$U_n^j$  = the upper 10% point of the  $n$  values from chain  $j$

$L_n$  = the lower 10% point of all  $mn$  values

$U_n$  = the upper 10% point of all  $mn$  values

Then

$$U_n - L_n$$

is a kind of "spread of the middle 80% of all values simulated," For each  $j$ ,

$$U_n^j - L_n^j$$

is a similar measure for the  $j$ th chain only, and

## "bgr diagnostic" Plots (cont.)

$$\frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)$$

is an "average across  $j$  spread of the middle 80% of values simulated from chain  $j$ ." If the  $m$  chains are doing the same thing, one expects that

$$U_n - L_n \approx \frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)$$

while big differences between chains should make

$$U_n - L_n \gg \frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)$$

This can be phrased as

$$\frac{U_n - L_n}{\frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)} \approx 1$$

indicating that the  $m$  chains have "burned in" to the same behavior, ☰



## "bgr diagnostic" Plots (cont.)

with

$$\frac{U_n - L_n}{\frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)} \gg 1$$

indicating that the Gibbs samplers have "failed to mix." WinBUGS plots the statistic

$$bgr = \frac{U_n - L_n}{\frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)}$$

in red on a set of axes where it also plots

$$\frac{U_n - L_n}{C} \text{ in green and } \frac{\frac{1}{m} \sum_{j=1}^m (U_n^j - L_n^j)}{C} \text{ in blue}$$

for  $C$  a constant chosen to put these quantities on a scale comparable to that of  $bgr$ .

## "bgr diagnostic" Plots (cont.)

WinBUGS plots the *bgr* value at multiples of 50 iterations and one looks at a plot hoping to find that

- *bgr* (in red) is essentially 1.0, and
- both the green and blue plots have stabilized (i.e. ceased to change with iteration) indicating that the transient effects of initialization have "washed out."

The figure on the next panel is a WinBUGS *bgr* diag plot for the first version of the change-point problem. It shows that 1000 iterations are probably adequate to assure "burn-in" has occurred and that using for analysis values produced by the Gibbs samplers after 2000 iterations appears to be safe practice in this more "tame" version of the problem.

## Example 3 (Version 1 cont.)

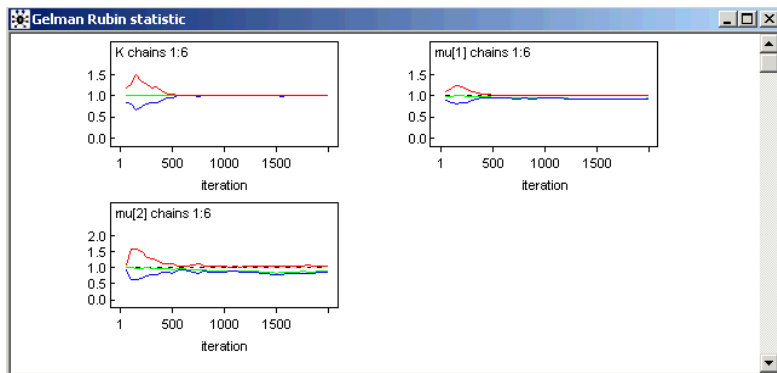


Figure: WinBUGS bgr diag plot for the first version of the change-point problem