Module 4: A First Look at WinBUGS and Practical Bayes Computation

"Real" professional Bayesians program their own MCMC algorithms, tailoring them to the models and data sets they face. The most widely used Bayes software available for non-specialists like you and me derives from the Biostatistics Unit at Cambridge University. The windows version is WinBUGS and there is apparently an open source version (that can be run in batch mode) called OPENBUGS. We'll illustrate WinBUGS in the balance of this workshop. WinBUGS has its own user manual and its own discussion list. These sources are far more authoritative than I will be. I am NOT a real expert with the system, and my intention is not to give you an exhaustive look at the software. Rather, it is my intention to give you a series of examples that will illustrate the power of the Bayes paradigm and the software in addressing important problems of inference in industry.

In order to make a WinBUGS analysis, one must

- write and have the software check the syntax of a model statement for the problem
- load any data needed for the analysis not specified in the model statement
- compile the program that will run the Gibbs sampler, and

- one way or another (either by supplying them or by generating them from the model itself) provide initial values for the sampler(s)/chain(s) that will be run

One then

- updates the sampler(s) as appropriate
- monitors the progress of the sampler(s), and

- ultimately summarizes what the sampler(s) indicate about the posterior distribution(s)
**Example 1**  As a first example, we will do the WinBUGS version of the small normal-normal model used in Module 2. The code for this is in the file BayesASQEx1.odc

Remember that the model is

\[ X \sim N(\theta, 1) \]
\[ \theta \sim N(5, 2) \]

(the prior variance is 2, so that the prior precision is .5) and we are assuming that \( X = 4 \) is observed.

Here is the code

```
model {
  X~dnorm(theta,1)
  Xnew~dnorm(theta,1)
  theta~dnorm(5,.5)
  #WinBUGS uses the precision instead of the variance or #standard deviation to name its normal distributions
  #so the prior variance of 2 is expressed as a prior #precision of .5
}
```

#here is a list of data for this example

```
list(X=4.0)
```

#here are 4 possible initializations for Gibbs samplers

```
list(theta=7,Xnew=3)
list(theta=2,Xnew=6)
list(theta=3,Xnew=10)
list(theta=8,Xnew=10)
```

The model is specified using the Specification Tool under the Model menu. One first uses the check model function, then the load data function to enter the list(X=4.0), then (for example choosing to run 4 parallel Gibbs samplers) employs the compile function. To initialize the simulation, one may either ask WinBUGS to generate initial values from the model, or one at a time enter 4 initializations for the chains like those provided above.

The Update Tool on the Model menu is used to get WinBUGS to do Gibbs updates of a current sampler state (in this case, a current \( \theta \) and value for \( X_{\text{new}} \)). The progress of the iterations can be watched and summarizations of the resulting simulated parameters (and new observations) can be produced using the Sample Monitor Tool under the Inference menu. Here are screen shots of what one gets for summaries of a fairly large number of iterations using the history, density, and stats functions of the Sample Monitor Tool. (One must first use the set function before updating, in order to alert WinBUGS to the fact that values of \( \theta \) and \( X_{\text{new}} \) should be collected for summarization.)
Figure 1: History plots for 50,000 iterations for 4 parallel chains (thinned to every 200th iteration for plotting purposes) for the toy normal-normal problem.

Figure 1 shows no obvious differences in the behaviors of the 4 chains (started from the fairly "dispersed" initializations indicated in the example code), which is of comfort if one is worried about the possibility of Gibbs sampling failing. Figures 2 and 3 are in complete agreement with the pencil and paper analyses offered in Module 2. Both the posterior for $\theta$ and the posterior predictive distribution of $X_{\text{new}}$ look roughly "normal" and the means and standard deviations listed in the "node statistics" are completely in line with posterior means and standard deviations. In fact, these can be listed in tabular form for comparison purposes as in Table 1.

Figure 2: Summary statistics for 50,000 iterations for 4 parallel chains for the toy normal-normal problem.

<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>HMC error</th>
<th>2.5%</th>
<th>median</th>
<th>97.5%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta</td>
<td>4.201</td>
<td>0.8144</td>
<td>0.003866</td>
<td>2.731</td>
<td>4.301</td>
<td>5.924</td>
<td>1</td>
<td>200000</td>
</tr>
</tbody>
</table>

Figure 3: Approximate densities estimated from 50,000 iterations for 4 parallel chains for the toy normal-normal problem.
Table 1: Theoretical (Pencil and Paper Calculus) and MCMC (Gibbs Sampling) Means and Standard Deviations for the Toy Normal-Normal Example

<table>
<thead>
<tr>
<th></th>
<th>( \theta )</th>
<th>( X_{new} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical Mean</td>
<td>4.333</td>
<td>4.333</td>
</tr>
<tr>
<td>MCMC Mean</td>
<td>4.331</td>
<td>4.333</td>
</tr>
<tr>
<td>Theoretical SD</td>
<td>( \sqrt{\frac{2}{3}} = .8165 )</td>
<td>( \sqrt{\frac{2}{3}} + 1 = 1.291 )</td>
</tr>
<tr>
<td>MCMC SD</td>
<td>.8144</td>
<td>1.289</td>
</tr>
</tbody>
</table>

This first example is a very "tame" example, the effect of the starting value for the Gibbs sampling is not important, and the samplers very easily produce the right posteriors.

One of the real powers of simulation as a way of approximating a posterior is that it is absolutely straightforward to approximate the posterior distribution of any function of the parameter vector \( \theta \), say \( h(\theta) \). One simply plugs simulated values of \( \theta \) into the function observes the resulting relative frequency distribution.

Continuing the normal-normal example, a function of \( \theta \) that could potentially be of interest is the fraction of the \( X \) distribution below some fixed value, say 3.0. (This kind of thing might be of interest if \( X \) were some part dimension and 3.0 were a lower specification for that dimension.) In this situation, the parametric function of interest is

\[
h(\theta) = \Phi\left(\frac{3 - \theta}{1}\right)
\]

and by simply adding the line of code

\[
\text{prob} \leftarrow \Phi(3.0 - \theta)
\]

This figure shows that while the posterior mean for this fraction of the \( X \) distribution is about 15%, very little is actually known about the quantity. In fact, if one wanted to have 95% posterior probability of bracketing \( \text{prob} = \Phi(3.0 - \theta) \), the so-called (approximate) 95% Bayes credible interval (running from the lower 2.5% point to the upper 2.5% point of the approximate posterior distribution)

\[
(.001726, .6062)
\]

would be used.

Example 2 As a second simple well-behaved example, consider a fraction non-conforming context, where one is interested in

\[
X = \text{the number non-conforming in a sample of } n = 50
\]

and believes a priori that

\[
p = \text{the process non-conforming rate}
\]
producing $X$ might be appropriately described as having mean .04 and standard deviation .04.

The model for the observable data here will be

$$X \sim \text{Binomial}(50, p)$$

Figure 5 shows a convenient prior density for $p$ that has the desired mean and standard deviation. This is the so-called Beta distribution with parameters $\alpha = .92$ and $\beta = 22.08$.

The general Beta density is

$$g(p) \propto p^{\alpha-1} (1-p)^{\beta-1}$$

and it is common in Bayes contexts to think of such a prior as contributing to an analysis information roughly equivalent to $\alpha$ "successes" (non-conforming items in the present context) and $\beta$ "failures" (conforming items in the present context). So employing a Beta(.92, 22.08) prior here is roughly equivalent to assuming prior information that a single non-conforming item has been seen in 23 inspected items.

The code in the file

BayesASQEx2.odc

can be used to find a $X = 4$ posterior distribution for $p$ and posterior predictive distribution for

$$X_{\text{new}} = \text{the number non-conforming in the next 1000 produced}$$

(of course assuming the stability of the process at the current $p$). The code is

```plaintext
model {
  X~dbin(p,50)
p~dbeta(.92,22.08)
Xnew~dbin(p,1000)
}
#here is the data for the problem
list(X=4)
#here are 4 possible initializations for Gibbs samplers
```
This is a "tame" problem (that could actually be solved completely by pencil and paper) and the 4 initializations all yield the same view of the posterior. Figure 6 provides an approximate view of the posterior distribution of \( p \) and the posterior predictive distribution of the number of non-conforming items among the next 1000 produced. (Note that in retrospect, it is no surprise that these distributions have essentially the same shape. \( n = 1000 \) is big enough that we should expect the sample fraction non-conforming among the 1000 to be about \( p \), whatever that number might be. Thus \( X_{\text{new}} \) should have a posterior predictive distribution much like a posterior for \( 1000p \).)