Introduction to Just Another Gibbs Sampler (JAGS)

Rebecca Steorts

Department of Statistics
University of Florida

Graduate Student Seminar

October 13, 2009

Outline

What is BUGS?

Definition

BUGS stands for **B**ayesian Inference **U**sing **G**ibbs **S**ampling.

- Examples used are WinBugs, OpenBUGS, and JAGS.
- Often Bayesian inference requires computing intractable integrals.
- BUGS uses Markov Chain Monte Carlo methods (specifically Gibbs sampling) to solve these intractable integrals.

MCMC and Gibbs Sampling

- A Markov Chain is a sequence of random variables, $X_1, X_2, \cdots, X_n$, that satisfy the property

  $P(X_{n+1} \in \mathrm{A} \mid X_1, X_2, \cdots, X_n) = P(X_{n+1} \in \mathrm{A} \mid X_n)$ for all sets A.

- A Gibbs sampler is an MCMC algorithm that generates a sequence of samples from the joint distribution of two or more random variables using the conditional distributions. It does this by cycling back and forth through the conditionals.

WinBUGS

- Classic Bugs was developed in 1995 and ran on all platforms.
- WinBugs was developed in 1997 and only ran on Windows machines.
- WinBugs is a point and click type program and your results will dump into coda files (G.S. output) which you can then read into R.
- WinBugs now runs on Unix with Intel processors (need to install Darwine).
- WinBugs basically emulates Windows on Unix (may be problems with Mac OS 10.5.8 and up).

OpenBUGS

- OpenBUGS is an open source version of WinBUGS.
- Runs on all platforms.
- Can be either menu or command line driven. Running from the command line requires BRugs package in R (not compatible with Unix or Linux).
- Can be synced with R using BRugs.
- Since BRugs currently does not work with all platforms, I prefer JAGS.

JAGS

- Developed by Martyn Plummer in 2003.
- Created to make more similar to Classic Bugs as well as make improvements.
- Runs on all platforms.
- Can run either in JAGS and read coda into R or run JAGS directly from R.
- Utilizes the Adaptive Rejection Metropolis sampler.

JAGS

To run a model in JAGS and then analyze your results in R, you need the following 5 files:

- A file containing your model (model.txt)
- A file containing your data (data.txt)
- A file containing your initial values (initial.txt)
- A jags script containing what you would like JAGS to run (script.txt)
- An R script reading in the coda files and then manipulating the Gibbs sampler results

Then we run the script in the terminal using the command "jags script" and the coda files go into our directory. We can now read the coda files into R and compute summary statistics and run diagnostics.

PIA2 Example

Setup:

- Twelve studies were run to investigate the potential link between presence of a certain genetic trait and risk of heart attack.
- Each study was case-control and considered a group of individuals with coronary heart disease and another group with no coronary heart disease.
- For each study i ($i = 1, \cdots, 12$) the proportion having the genetic trait in each group was recorded.
- For each study, a log odds ratio, $\hat{\psi}_i$, and standard error, σ_i, were calculated.

PIA2 Example

PIA2 data (Burr et al. 2003):

i	1	2	3	4	5	6
$\hat{\psi}_i$	1.06	-0.10	0.62	0.02	1.07	-0.02
σ_i	0.37	0.11	0.22	0.11	0.12	0.12

i	7	8	9	10	11	12
$\hat{\psi}_i$	-0.12	-0.38	0.51	0.00	0.38	0.40
σ_i	0.22	0.23	0.18	0.32	0.20	0.25

PIA2 Example

Our hierarchical model is

$$\hat{\psi}_i \mid \psi_i \overset{ind}{\sim} N(\psi_i, \sigma_i^2) \quad i = 1, \cdots, 12$$

$$\psi_i \mid \mu, \tau^2 \overset{ind}{\sim} N(\mu, \tau^2) \quad i = 1, \cdots, 12$$

$$\mu \mid \tau^2 \sim N(0, 1000\tau^2)$$

$$\gamma = 1/\tau^2 \sim \text{Gamma}(0.1, 0.1)$$

PIA2 Example

The file model.txt contains

```
model {
for (i in 1:N) {
psihat[i] ~ dnorm(psi[i],1/(sigma[i])^2)
psi[i] ~ dnorm(mu,1/tau^2)
}
mu ~ dnorm(0,1/(1000*tau^2))
tau <- 1/sqrt(gam)
gam ~ dgamma(0.1,0.1)
}
```

Note: In BUGS, use dnorm(mean,precision), where precision = 1/variance.

PIA2 Example

The file data.txt contains

```
"N" <- 12
"psihat" <-  c(1.055, -0.097, 0.626, 0.017, 1.068,
-0.025, -0.117, -0.381, 0.507, 0, 0.385, 0.405)
"sigma" <- c(0.373, 0.116, 0.229, 0.117, 0.471,
0.120, 0.220, 0.239, 0.186, 0.328, 0.206, 0.254)
```

The file initial_1.txt contains

```
".RNG.name" <- "base::Super-Duper"
".RNG.seed" <- 12
"psi" <- c(0,0,0,0,0,0,0,0,0,0,0,0)
"mu" <- 0
"gam" <- 1
```

PIA2 Example

The file script.txt contains

```
model clear
data clear
model in "model.txt"
data in "data.txt"
compile
inits in "initial_1.txt"
initialize
update 10000
monitor mu, thin(10)
monitor psi, thin(10)
monitor gam, thin(10)
update 100000
coda *
```

Diagnostics

Now, we read in the coda files into R from the current directory and continue our analysis. The first part of our analysis will consist of some diagnostic procedures.

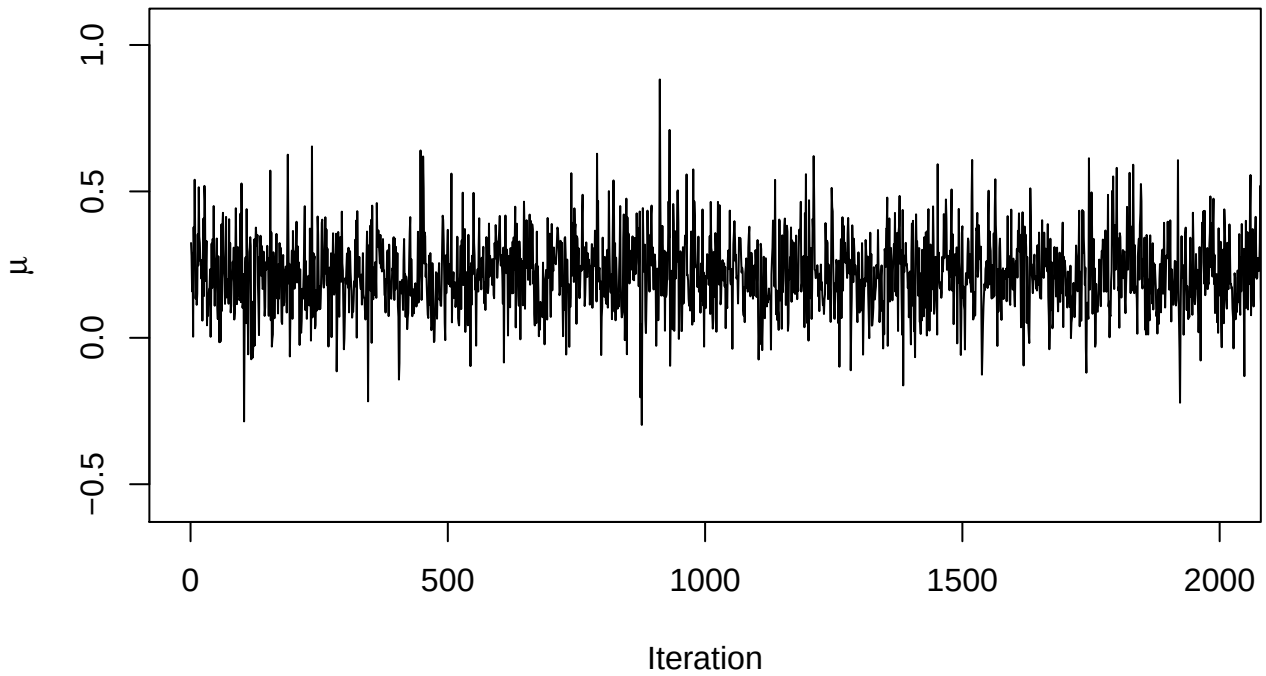
We will consider

- Trace Plots
- Autocorrelation Plots
- Gelman-Rubin Diagnostic
- Geweke Diagnostic

Trace Plots

Definition

A trace plot is a time series plot of the parameter, say μ, that we monitor as the Markov chain proceeds.



Autocorrelation Plot

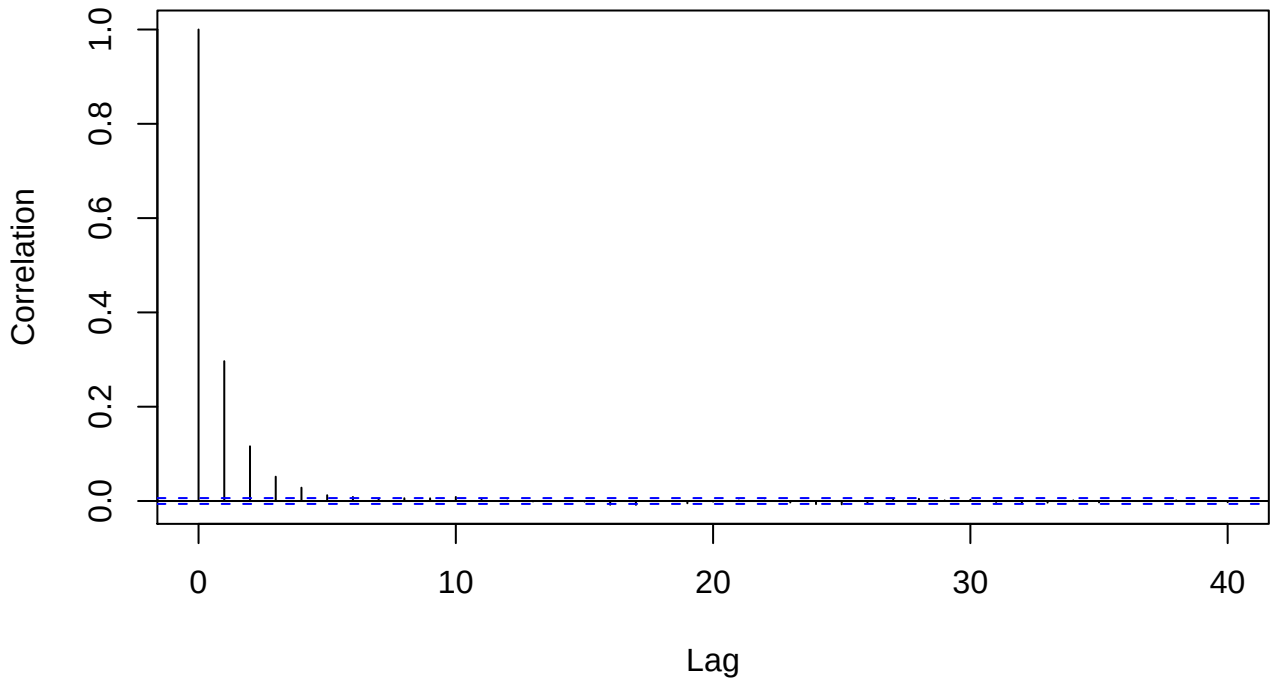
Definition

An autocorrelation plot graphically measures the correlation between X_1 and each X_{k+1} variable in the chain.

- The Lag-k correlation is the $\text{Corr}(X_1, X_{k+1})$.
- By looking at autocorrelation plots of parameters that we are interested in, we can decide how much to thin or subsample our chain by.
- We can rerun our JAGS script using our thin value.

Autocorrelation Plot of μ for PlA2 Data

We take the thin value to be the first lag whose correlation ≤ 0.2. For this plot, we take a thin of 2. We will go back and rerun our JAGS script and skip every other value in each chain.



Gelman-Rubin Diagnostic

Definition

The Gelman-Rubin diagnostic tests that burn-in is adequate and requires that multiple starting points be used.

To compute the G-R statistic, we must

- Run two chains in JAGS using two different sets of initial values (and two different seeds).
- Load coda package in R and run gelman.diag(mcmc.list(chain1,chain2)).

Gelman-Rubin Diagnostic

How do we interpret the Gelman-Rubin Diagnostic?

- If the chain has reached convergence, the G-R test statistic R $\approx$ 1. We conclude that burn-in is adequate.
- Values above 1.05 indicate lack of convergence.

Warning

The distribution of R under the null hypothesis is essentially an F distribution. Recall that the F-test for comparing two variances is not robust to violations of normality. Thus, we want to be cautious in using the G-R diagnostic.

Gelman-Rubin Diagnostic

Doing this in R, for the PIA-2 example, we find

	Point est.	97.5% quantile
mu	1	1
psi[1]	1	1
psi[2]	1	1
...		
psi[11]	1	1
psi[12]	1	1
gam	1	1

Since 1 is in all the 95% CI, we can conclude that we have not failed to converge.

Geweke Diagnostic

Suppose μ is the parameter of interest.

Main Idea

If burn-in is adequate, the mean of the posterior distribution of μ from the first half of the chain should equal the mean from the second half of the chain.

To compute the Geweke statistic, we must

- Run a chain in JAGS along with a set of initial values.
- Load the coda package in R and run geweke.diag(mcmc.list(chain)).

Geweke Diagnostic

- The Geweke statistic asymptotically has a standard normal distribution, so if the values from R are outside -2.5 or 2.5, this indicates nonstationarity of chain and that burn-in is not sufficient.
- Using the Geweke diagnostic on the PIA2 data indicates that burn-in of 10,000 is sufficient (the largest absolute Z-score is 1.65).
- Observe that the Geweke diagnostic does not require multiple starting points as Gelman-Rubin does.
- The Geweke statistic (based on a T-test) is robust against violations of normality so the Geweke test is preferred to Gelman-Rubin.

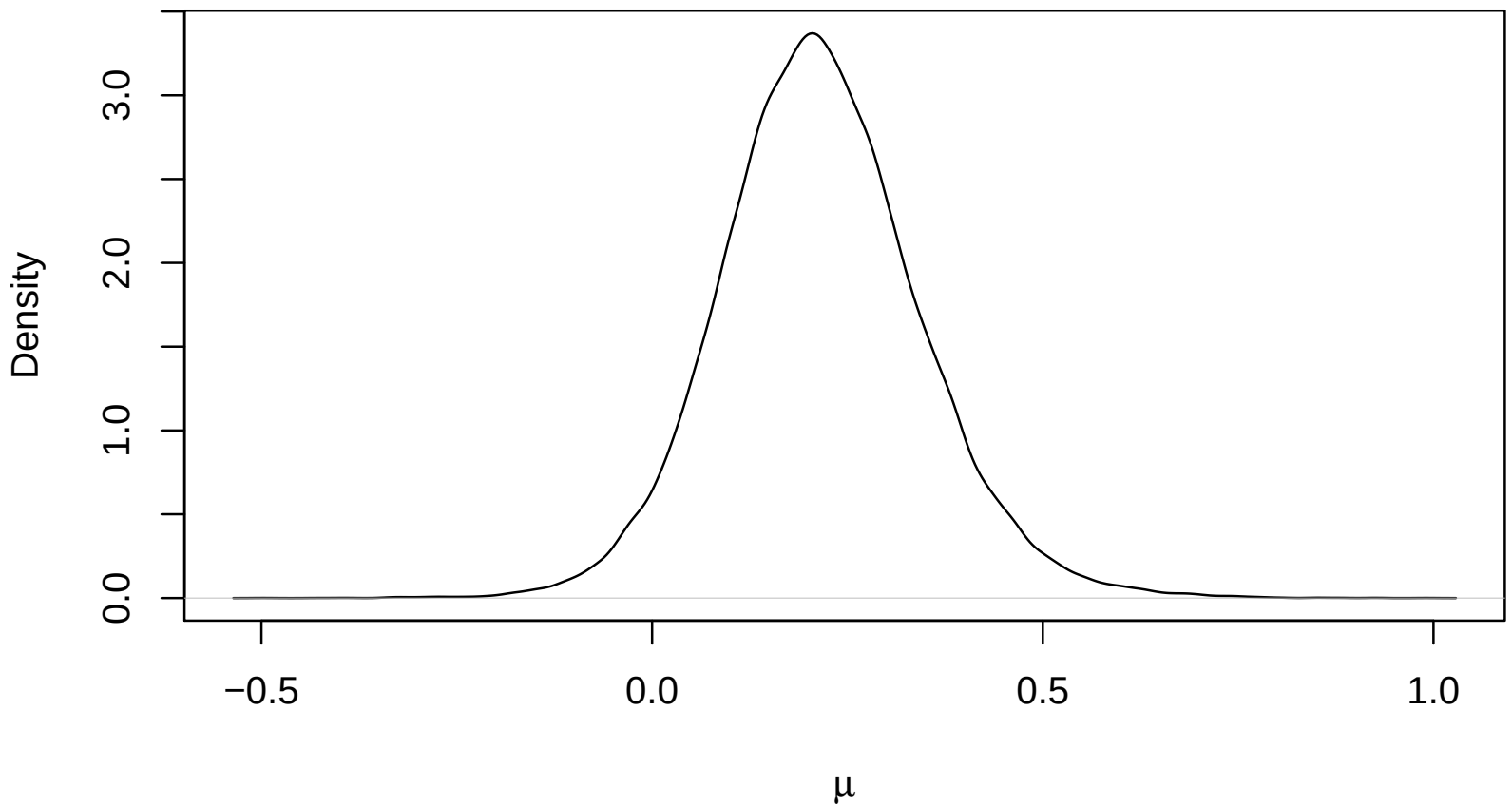
Analysis

Using Gelman-Rubin and Geweke we have shown that burn-in is "sufficient" and using the autocorrelation plot for μ we found a thin of 2. We can rerun our model in JAGS, now thinning.

- We can look at summary statistics such as means, standard errors, and credible intervals using the summary function.
- We can use kernel density functions in R to estimate posterior distributions that we are interested in using the density function.

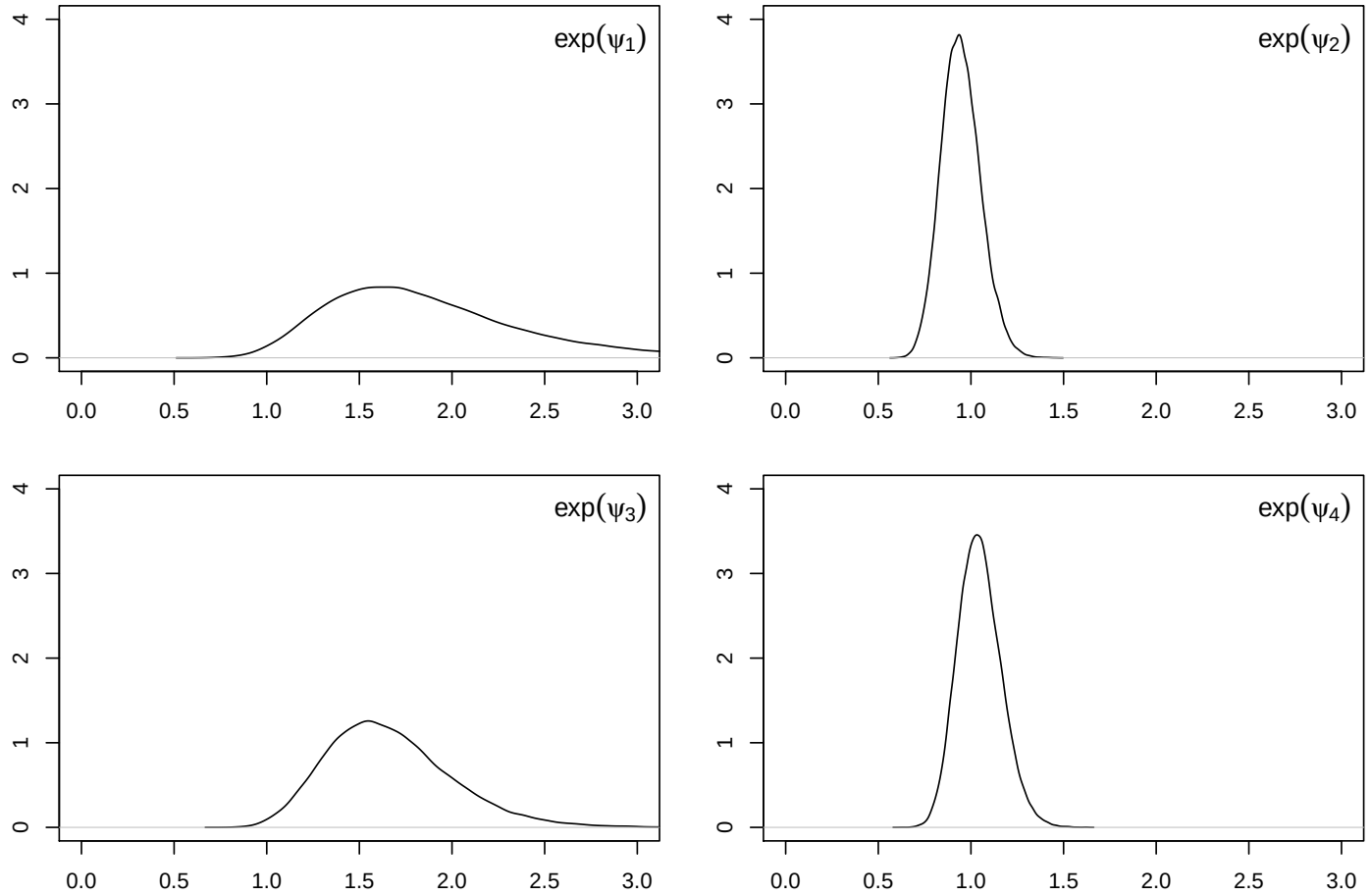
Analysis

The posterior of $\mu \mid$ data is



Analysis

Alternatively, we can estimate the conditional distributions of $\exp(\psi_i)$'s given the data. A few are shown below.



Running JAGS on Your Own

- To install JAGS, download Martyn Plummer's guide to JAGS from http://www-fis.iarc.fr/∼martyn/software/jags/ and follow downloading instructions depending on the platform you are running.
- I have posted a tutorial on my webpage on how to run the PLA2 data. All the files I ran in my presentation today are on my webpage along with today's presentation.

Summary

- BUGS is a useful tool for Bayesian inference when the models become very complicated
- We should be familiar with diagnostic procedures available to us, but beware that they are not foolproof and only tell us when our models have not converged. We cannot prove a model has reached convergence by a diagnostic procedure.
- JAGS runs on all platforms and works very well with R so this may be the most convenient version of BUGS to use. You'll want to make this decision on your own.