

WinBUGS Demo

Saghir A. Bashir

Amgen Ltd, Cambridge, U.K.
Email: saghir_bashir@hotmail.com

4th January 2001

Outline

- Introduction
- BUGS and WinBUGS
- Graphical Models
- DoodleBUGS
- Example - Simulation
- Power calculation
- Summary

Introduction

- Bayesian Inference Using Gibbs Sampling
 - BUGS
- Analysis of Complex Models
- Bayesian Methods
- Markov Chain Monte Carlo Integration
 - Useful when no closed form exists

Classic BUGS

- Declarative Language
 - Similar to Splus
- Complex Statistical Models
 - Missing data
 - Measurement Error
 - No closed form for Likelihood
- Graphical Modelling
- Flexible compared to approximations

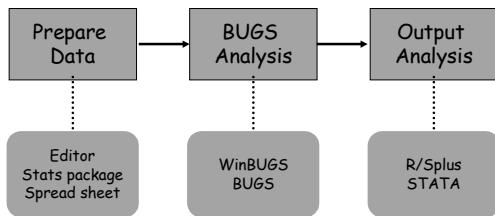
WinBUGS

- Similar to Classic BUGS
 - Plus new methodological developments
- Graphical representation of model
 - DoodleBUGS
- Menu Control of session
- Cut and paste to other packages

BUGS and WinBUGS

- No data management facility
 - Why reinvent the wheel?
- "Easy" interface with other packages
 - R and Splus
 - Stata (S. Bashir)
- Simple analysis of output

Working with BUGS



Graphical Models

- Complex multivariate probability models
 - Representation
 - Visualisation
- Graphs...
 - simplify complex models
 - communicate structure of the problem
 - provide basis for computation

WinBUGS

- BUGS language
- DoodleBUGS
 - Used for the purposed of this Demo.
- WinBUGS is currently free from:
 - <http://www.mrc-bsu.cam.ac.uk/bugs/>
 - Register to get full version access

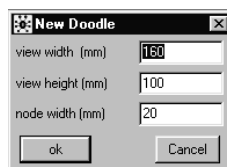
DoodleBUGS

- Start WinBUGS
- Select "Doodle" from menu bar



DoodleBUGS - Basics

- Select "New..."



- Press "ok"
- You have a window to "Doodle" in.

Nodes

- Creating a node
 - Mouse click in Doodle Window

name: | type: stochastic density: dnom
mean: 0.0 precision: 1.0E-6 lower bound upper bound



- Deleting a node: CTRL + Del

Node Types

- Nodes can be
 - Stochastic

name: rate type: stochastic density: dbin

mean: 0.25 lower bound: 0 upper bound: 250

rate

- Logical

name: prediction type: logical link: identity

value: alpha * beta * gamma

prediction

- Constant (rectangle)

name: observed type: constant

observed

Example - Simulation

- Let
 - $r1 \sim \text{Bin}(0.25, 250)$
 - $r2 \sim \text{Bin}(0.35, 150)$
- Calculate p: common proportion for r1 & r2
- $p = (r1+r2)/400$
- Classical $p = 0.2875$

DoodleBUGS

- Start with $r1 \sim \text{Bin}(0.25, 250)$ (stochastic node)

name: r1 type: stochastic density: dbin

proportion: 0.25 order: 250 lower bound: 0 upper bound: 250

r1

DoodleBUGS

- Add $r2 \sim \text{Bin}(0.35, 150)$ (stochastic node)

name: r2 type: stochastic density: dbin

proportion: 0.35 order: 150 lower bound: 0 upper bound: 150

r1

r2

Logical Nodes

- Add p as a logical node

name: p type: logical link: identity

value: (r1+r2)/400

r1

r2

p

- To define a logical node click on "type" for choices.

Logical Functions

- Add "edges" for the logical relationship

name: p type: logical link: identity

value: (r1+r2)/400

r1

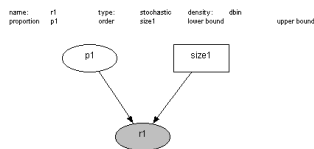
r2

p

- Whilst node p is highlighted, CTRL + click in "parent nodes" r1 and r2 (hollow arrows \Rightarrow logical function)

Stochastic Nodes

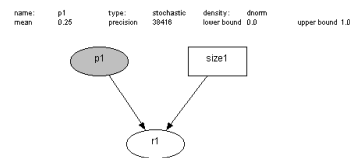
- Stochastic dependence
 - $p1 \sim N(0.25, 0.000026)$ (i.e., $p1 \sim [0.24, 0.26]$)
 - $size1 = 250$ (constant)



- Single arrows for stochastic dependencies

Normal Distribution

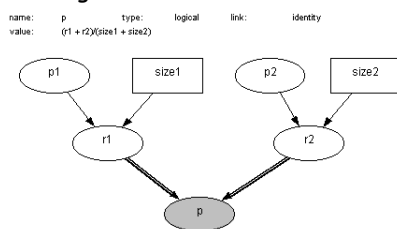
- Note the Normal distribution in BUGS is defined as $N(\text{mean}, \text{precision})$ where $\text{precision} = 1/\text{variance}$



- Note that we can define upper and lower bounds so that the proportion is between 0 and 1.

DoodleBUGS Model

- Let us add these stochastic dependencies to our "logical" model



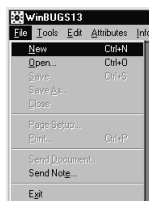
DoodleBUGS Model

- What is our model?
 - $r1 \sim \text{Bin}(p1, size1)$
 - $p1 \sim N(0.25, 0.000026)$
 - $size1 = 250$
 - $r2 \sim \text{Bin}(p2, size2)$
 - $p2 \sim N(0.35, 0.000026)$
 - $size2 = 150$

WinBUGS Modelling

- Running our model in WinBUGS

- Create a New document
 - Menu bar - File - New



- A New document window will appear

WinBUGS Document

- Select your Doodle from your Doodle Window

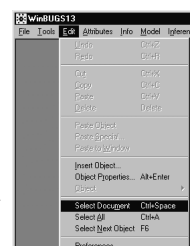
- Menu bar - Edit - Select Document

- Copy your Doodle

- Menu bar - Edit - Copy

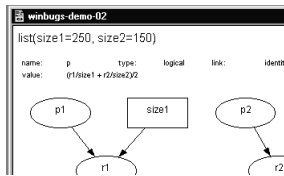
- Paste it into your New Document

- Menu bar - Edit - Paste



Model Data

- Before running we need to give BUGS some data
 - Type `list(size1=250, size2=150)` at the top (or the bottom) of your new document.

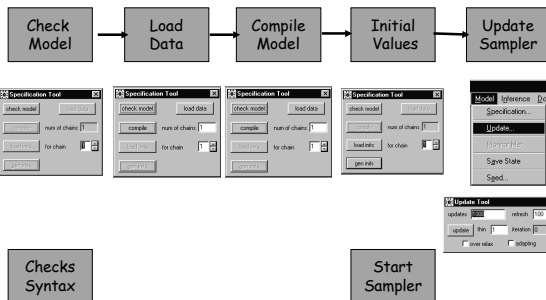


Running BUGS

- Use "Specification..." from the "Model" option on Menu Bar to run BUGS

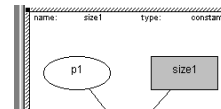


Running BUGS



Check Model

- Select the Doodle (note the hairy border)

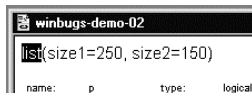


- Menu bar - Model - Check model
- Note the message in bottom left hand corner

model is syntactically correct

Load Data

- Highlight the word "list"



- Menu bar - Model - Data
- Bottom left hand corner

data loaded

Compiling the Model

- Menu bar - Model - Compile
- Bottom left hand corner

model compiled

Load Initial Values

- Menu bar - Model - Gen inits
- Bottom left hand side

initial values generated

Update the Model

- Menu bar - Model - Update



- 1000 MCMC updates to be carried out.

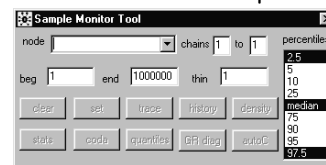


Burn In

- Model has been updated
- MCMC run did not store any data.
 - Used for the "burn in"
- Store values by "monitoring" them to
 - Draw inferences
 - Monitor MCMC run

Monitoring Nodes

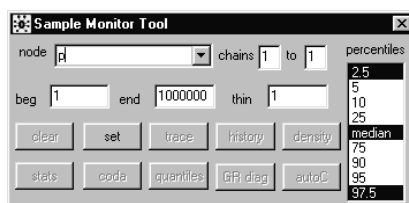
- Monitoring p our parameter of interest
- Menu bar - Inference - Samples...



- Sample Monitor Tool

Monitoring Nodes

- Type name of node "p" to monitor
- Press "set"



Update & Monitor

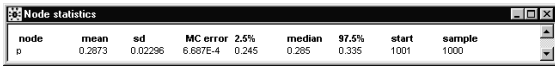
- Update model again



- 1000 values "monitored" of the MCMC run for p

Summary Statistics

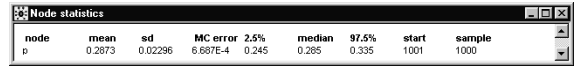
- Summary statistics
- Select "p" from the Sample Monitor Tool
- Press "stats" (Sample Monitor Tool)



node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p	0.2873	0.02296	6.687E-4	0.245	0.285	0.335	1001	1000

- Node statistics window

Summary Statistics

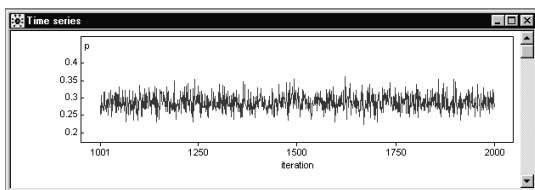


node	mean	sd	MC error	2.5%	median	97.5%	start	sample
p	0.2873	0.02296	6.687E-4	0.245	0.285	0.335	1001	1000

- Mean = 0.2873
- Median = 0.285 (usually more stable)
- 95% credible interval (0.245, 0.335)
- MCMC run size 1000

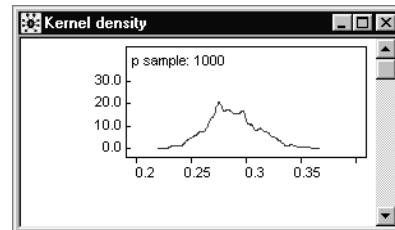
MCMC Time Series

- Press "History" in Sample Monitor Tool



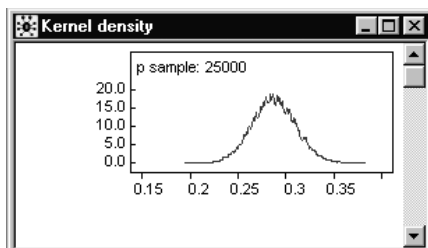
Kernel Density

- Press "Density" in the Sample Monitor Tool



Kernel Density

- Increase monitored values to 25,000



Plates

- Creating a plate
 - CTRL + mouse click in Doodle Window



- Deleting a plate: CTRL + Del

Plates

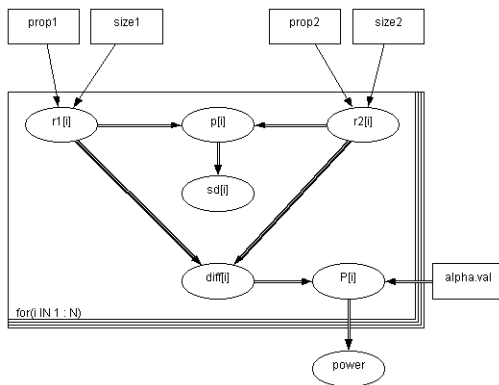
- Allow more complex structure, e.g.,
 - Repeated measures
 - Hierarchical models
- Extend our example to calculate power
 - r_1 and r_2 from Binomial distribution
 - Simulate r_1 and r_2 100 times per "update"
 - Calculate test statistic
 - Count number of times it falls in critical region

Test Statistic

- $H_0: p_1 = p_2 = p$ vs $H_1: p_1 < p_2$
 - $p_1 = r_1/\text{size}_1$ & $p_2 = r_2/\text{size}_2$

- Test statistic $\frac{p_2 - p_1}{\text{s.d.}(p)}$

$$\text{s.d.}(p) = \sqrt{p(1-p)(1/\text{size}_1 + 1/\text{size}_2)}$$



Power

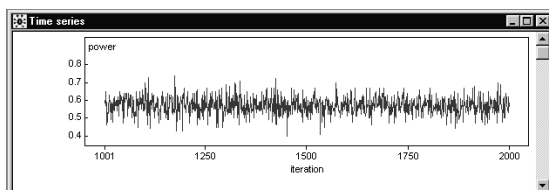
- Data
 - `list(prop1=.25, prop2=.35, size1=250, size2=150, N=100, alpha.val=1.96)`

- Results

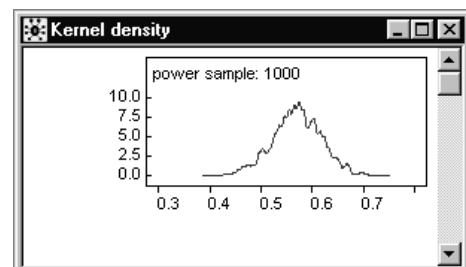
Node statistics								
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
power	0.5705	0.04699	0.001392	0.47	0.57	0.67	1001	1000

- Power = 57% (47%, 67%)

Power - History



Power - Density



Updates

- Updating - Bottom left hand corner

updating

- After updates finish

updates took 69 s

Summary

- BUGS is a power tool
 - Bayesian Analysis
 - Simulation Tool
- Graphical Models
 - Doodle BUGS
 - Simple representation of model
- Easy to use!