Introduction to Bayesian Analysis using WinBUGS

Nicky Best, Alexina Mason and Philip Li

(Thanks to Sylvia Richardson, David Spiegelhalter)

Short Course, Feb 16, 2011

http://www.bias-project.org.uk

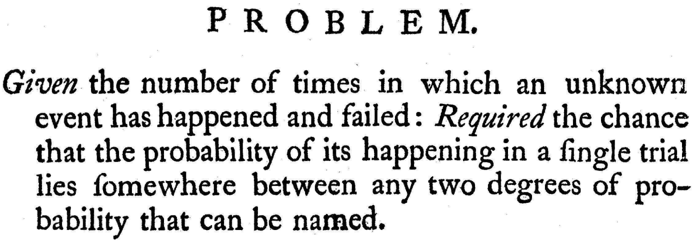
Lecture 1: Introduction to Bayesian Monte Carlo methods in WINBUGS

Outline

- Probability as a means of representing uncertainty
- Bayesian direct probability statements about parameters
- Probability distributions
- Monte Carlo simulation
- Implementation in WinBUGS — Demo
- Graphical representation of probability models
- Examples

How did it all start?

In 1763, Reverend Thomas Bayes of Tunbridge Wells wrote



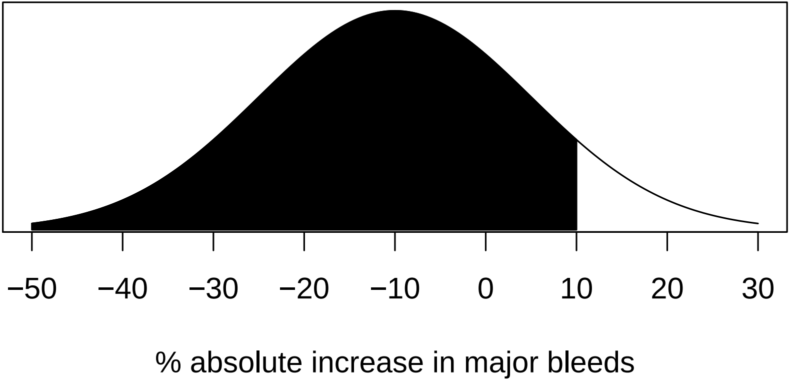
PROBLEM.

Given the number of times in which an unknown event has happened and failed: *Required* the chance that the probability of its happening in a ſingle trial lies ſomewhere between any two degrees of probability that can be named.

In modern language, given $r \sim \mathrm{Binomial}(\theta, n)$, what is $\Pr(\theta_1 < \theta < \theta_2 | r, n)$?

Basic idea: Direct expression of uncertainty about unknown parameters

eg "There is an 89% probability that the absolute increase in major bleeds is less than 10 percent with low-dose PLT transfusions" (Tinmouth et al, Transfusion, 2004)



Why a direct probability distribution?

- Tells us what we want: what are plausible values for the parameter of interest?
- No *P-values*: just calculate relevant tail areas
- No (difficult to interpret) *confidence intervals*: just report, say, central area that contains 95% of distribution
- Easy to make predictions (see later)
- Fits naturally into decision analysis, cost-effectiveness analysis, project prioritisation
- There is a procedure for adapting the distribution in the light of additional evidence: i.e. *Bayes theorem* allows us to learn from experience

Inference on proportions

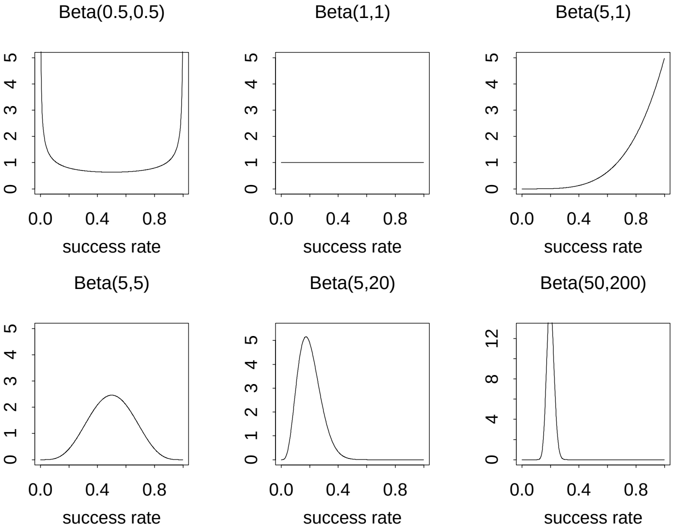
- What is a reasonable form for a probability distribution for a proportion?
- $\theta \sim \text{Beta}(a, b)$ represents a beta distribution with properties:

$$
\begin{aligned}
p(\theta|a,b) &= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\,\theta^{a-1}\,(1-\theta)^{b-1}; \quad \theta \in (0,1) \\
\mathrm{E}(\theta|a,b) &= \frac{a}{a+b} \\
\mathrm{V}(\theta|a,b) &= \frac{ab}{(a+b)^2(a+b+1)}:
\end{aligned}
$$

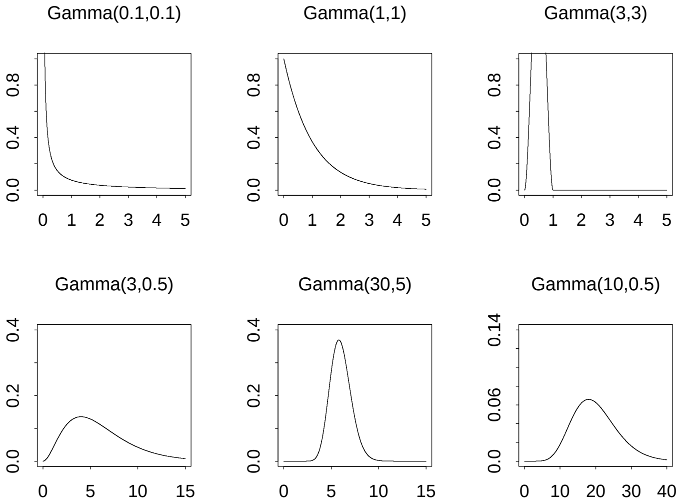
- WinBUGS notation:

```
theta ~ dbeta(a,b)
```

Some Beta distributions



Some Gamma distributions



The Gamma distribution

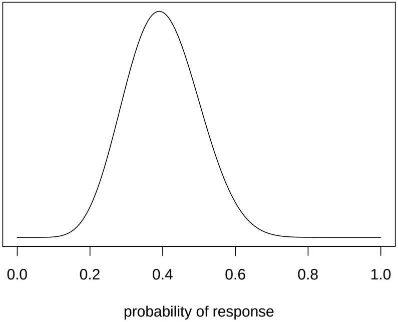
Flexible distribution for positive quantities. If $Y \sim \text{Gamma}[a, b]$

$$
\begin{aligned}
p(y|a,b) &= \frac{b^a}{\Gamma(a)} y^{a-1}\, e^{-by}; \quad y \in (0,\infty) \\
\mathrm{E}(Y|a,b) &= \frac{a}{b}; \quad \mathrm{V}(Y|a,b) = \frac{a}{b^2}.
\end{aligned}
$$

- Gamma(1,b) distribution is exponential with mean $1/b$
- Gamma($\frac{v}{2}, \frac{1}{2}$) is Chi-squared dist on v degrees of freedom
- Used as conjugate prior distribution for inverse variances (precisions)
- Used as sampling distribution for skewed positive valued quantities (alternative to log normal likelihood)
- WinBUGS notation: `y ~ dgamma(a,b)`

Example: Drug

- Consider a drug to be given for relief of chronic pain
- Experience with similar compounds has suggested that annual response rates between 0.2 and 0.6 could be feasible
- Interpret this as a distribution with mean = 0.4, standard deviation 0.1
- A Beta(9.2,13.8) distribution has these properties:



Making predictions

- Before observing a quantity Y, can provide its predictive distribution by integrating out unknown parameter

$$p(Y) = \int p(Y|\theta)p(\theta)d\theta.$$

- Predictions are useful in e.g. cost-effectiveness models, design of studies, checking whether observed data is compatible with expectations, and so on.

Drug example: Predictions

$$\begin{aligned} \theta &\sim \text{Beta}(a, b) \\ Y_n &\sim \text{Binomial}(\theta, n), \end{aligned}$$

The exact predictive distribution for Y_n is known as the **Beta-Binomial**. It has the complex form

$$p(y_n) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \left(\begin{array}{c} n \\ y_n \end{array} \right) \frac{\Gamma(a+y_n)\Gamma(b+n-y_n)}{\Gamma(a+b+n)}.$$

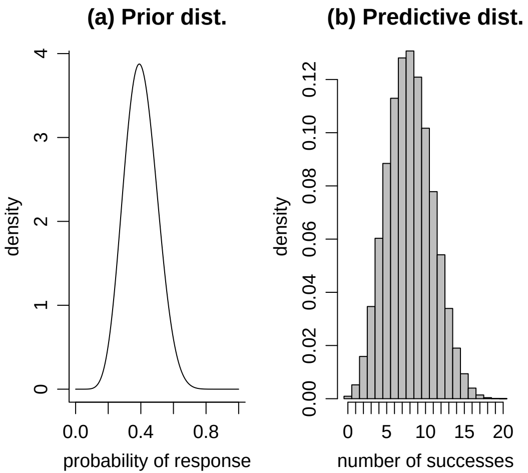
$$\text{mean, } \mathrm{E}(Y_n) = n\frac{a}{a+b}$$

If $a = b = 1$ (Uniform distribution), $p(y_n)$is uniform over 0,1,...,n.

But in WINBUGS we can just write

```
theta ~ dbeta(a,b)
Y     ~ dbin(theta,n)
```

and the integration is automatically carried out and does not require algebraic cleverness.



(a) is the Beta(9.2, 13.8) prior probability distribution for the response rate θ
(b) is the predictive Beta-Binomial distribution of the number of successes Y in the next 20 trials

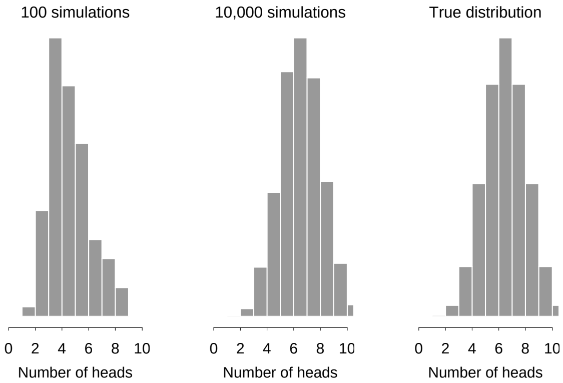
Example: a Monte Carlo approach to estimating tail-areas of distributions

Suppose we want to know the probability of getting 8 or more heads when we toss a fair coin 10 times.
An *algebraic* approach:

$$
\begin{aligned}
\Pr(\geq 8 \text{ heads}) &= \sum_{z=8}^{10} p\left(z|\pi = \frac{1}{2}, n = 10\right) \\
&= \begin{pmatrix} 10 \\ 8 \end{pmatrix} \left(\frac{1}{2}\right)^{8} \left(\frac{1}{2}\right)^{2} + \begin{pmatrix} 10 \\ 9 \end{pmatrix} \left(\frac{1}{2}\right)^{9} \left(\frac{1}{2}\right)^{1} \\
&\quad + \begin{pmatrix} 10 \\ 10 \end{pmatrix} \left(\frac{1}{2}\right)^{10} \left(\frac{1}{2}\right)^{0} \\
&= 0.0547.
\end{aligned}
$$

A *physical* approach would be to repeatedly throw a set of 10 coins and count the proportion of throws that there were 8 or more heads.

A *simulation* approach uses a computer to toss the coins!



Proportion with 8 or more ’heads’ in 10 tosses:

(a) After 100 ’throws’ (0.02); (b) after 10,000 throws (0.0577); (c) the true Binomial distribution (0.0547)

General Monte Carlo analysis - ‘forward sampling’

Used extensively in risk modelling - can think of as ‘adding uncertainty’ to a spreadsheet

- Suppose have logical function f containing uncertain parameters
- Can express our uncertainty as a prior distribution
- Simulate many values from this prior distribution
- Calculate f at the simulated values (‘iterations’)
- Obtain an empirical predictive distribution for f
- Sometimes termed *probabilistic sensitivity analysis*
- Can do in Excel add-ons such as `@RISK` or `Crystal Ball`.

The BUGS program

Bayesian inference Using Gibbs Sampling

- Language for specifying complex Bayesian models
- Constructs object-oriented internal representation of the model
- Simulation from full conditionals using Gibbs sampling
- Current versions:
    - WinBUGS 1.4.3 (runs in Windows)
        - Established, stable version of software
        - Can run in batch mode or be called from other software using scripts
        - Interfaces developed for R, Excel, Splus, SAS, Matlab
        - Freely available from `http://www.mrc-bsu.cam.ac.uk/bugs`
    - OpenBUGS (runs on Windows, Unix/Linux and Macs (via Wine))
        - Open source version on which all future developments will take place
        - Freely available from `http://www.openbugs.info`

In this course, we will be using WinBUGS 1.4.3

Running WinBUGS for Monte Carlo analysis (no data)

1. Open *Specification tool* from *Model* menu.
2. Program responses are shown on bottom-left of screen.
3. Highlight `model` by double-click. Click on *Check model*.
4. Click on *Compile*.
5. Click on *Gen Inits*.
6. Open *Update* from *Model* menu, and *Samples* from *Inference* menu.
7. Type nodes to be monitored into *Sample Monitor*, and click *set* each.
8. Type * into *Sample Monitor*, and click *trace* to see sampled values.
9. Click on *Update* to generate samples.
10. Type * into *Sample Monitor*, and click *stats* etc to see results on all monitored nodes.

Using WinBUGS for Monte Carlo analysis

- The model for the 'coin' example is

  $$Y \sim \text{Binomial}(0.5, 10)$$

  and we want to know $P(Y \geq 8)$.
- This model is represented in the BUGS language as

```
model{
 Y     ~  dbin(0.5,10)
 P8   <-  step(Y-7.5)
 }
```

- P8 is a step function which will take on the value 1 if `Y -7.5` is $\geq$ 0, i.e. Y is 8 or more, 0 if 7 or less.
- Running this simulation for 100, 10000 and 1000000 iterations, and then taking the empirical mean of P8, provided the previous estimated probabilities that Y will be 8 or more.

Some aspects of the BUGS language

- `<-` represents logical dependence, *e.g.* `m <- a + b*x`
- `~` represents stochastic dependence, *e.g.* `r ~ dunif(a,b)`
- Can use arrays and loops

```
for (i in 1:n){
  r[i] ~ dbin(p[i],n[i])
  p[i] ~ dunif(0,1)
}
```

- Some functions can appear on left-hand-side of an expression, *e.g.*

```
logit(p[i])<- a + b*x[i]
log(m[i])  <- c + d*y[i]
```

- `mean(p[])` to take mean of whole array, `mean(p[m:n])` to take mean of elements m to n. Also for `sum(p[])`.
- `dnorm(0,1)I(0,)` means the prior will be restricted to the range $(0, \infty)$.

Functions in the BUGS language

- `p <- step(x-0.7)` = 1 if x $\geq$ 0.7, 0 otherwise. Hence monitoring p and recording its mean will give the probability that x$\geq$ 0.7.
- `p <- equals(x,0.7)` = 1 if x $=$ 0.7, 0 otherwise.
- `tau <- 1/pow(s,2)` sets $\tau = 1/s^2$.
- `s <-   1/ sqrt(tau)` sets $s = 1/\sqrt{\tau}$.
- `p[i,k] <- inprod(pi[], Lambda[i,,k])` sets $p_{ik} = \sum_j \pi_j \Lambda_{ijk}$. `inprod2` may be faster.
- See ‘Model Specification/Logical nodes’ in manual for full syntax.

Some common Distributions in the BUGS language

Expression	Distribution	Usage
`dbin`	binomial	`r ~ dbin(p,n)`
`dnorm`	normal	`x ~ dnorm(mu,tau)`
`dpois`	Poisson	`r ~ dpois(lambda)`
`dunif`	uniform	`x ~ dunif(a,b)`
`dgamma`	gamma	`x ~ dgamma(a,b)`

- The normal is parameterised in terms of its mean and *precision* = 1/ variance = 1/sd^2.
- See ‘Model Specification/The BUGS language: stochastic nodes/Distributions’ in manual for full syntax.
- **Functions cannot be used as arguments in distributions (you need to create new nodes).**

Drug example: Monte Carlo predictions

- Our prior distribution for proportion of responders in one year θ was Beta$(9.2, 13.8)$.
- Consider situation *before* giving 20 patients the treatment. What is the chance if getting 15 or more responders?

$$\begin{array}{cclll} \theta & \sim & \text{Beta}(9.2, 13.8) & & \text{prior distribution} \\ y & \sim & \text{Binomial}(\theta, 20) & & \text{sampling distribution} \\ P_{\text{crit}} & = & P(y \geq 15) & & \text{Probability of exceeding critical threshold} \end{array}$$

```
# In BUGS syntax:
 model{
 theta     ~ dbeta(9.2,13.8) # prior distribution
 y         ~ dbin(theta,20)  # sampling distribution
 P.crit   <- step(y-14.5)    # =1 if y >= 15, 0 otherwise

 }
```

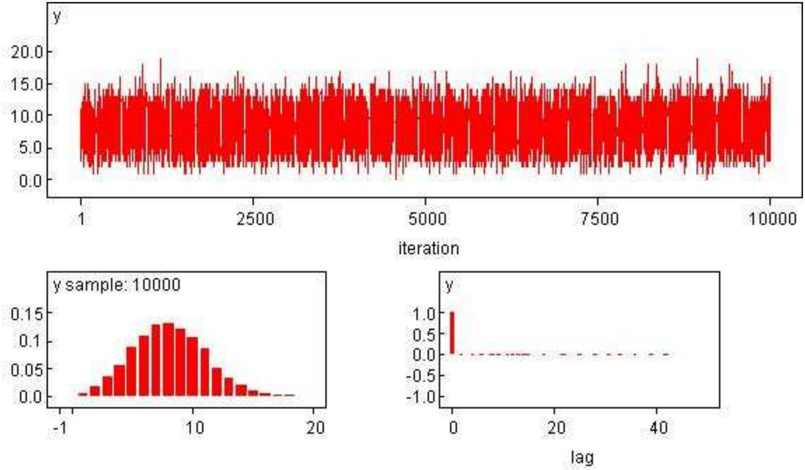
WinBUGS output and exact answers

```
node    mean  sd    MC error 2.5%  median 97.5% start sample

theta   0.400 0.099 9.41E-4  0.217 0.398  0.604 1     10000
y       8.058 2.917 0.03035  3.0   8.0    14.0  1     10000
P.crit  0.015 0.122 0.00128  0.0   0.0    0.0   1     10000
```

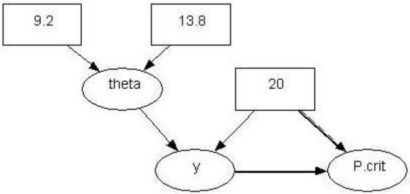
- Note that the mean of the 0-1 indicator `P.crit` provides the estimated tail-area probability.
- Exact answers from closed-form analysis:
  - θ: mean 0.4 and standard deviation 0.1
  - y: mean 8 and standard deviation 2.93.
  - Probability of at least 15: 0.015
- These are independent samples, and so MC error = SD/$\sqrt{\text{Number of iterations}}$.
- Can achieve arbitrary accuracy by running the simulation for longer.

WinBUGS output



Independent samples, and so no auto-correlation and no concern with convergence.

Graphical representation of models



- *Doodle* represents each quantity as a node in directed acyclic graph (DAG).
- Constants are placed in rectangles, random quantities in ovals
- Stochastic dependence is represented by a single arrow, and logical function as double arrow
- WinBUGS allows models to be specified graphically and run directly from the graphical interface
- Can write code from Doodles
- Good for explanation, but can be tricky to set up

Script for running Drug Monte Carlo example

Run from Model/Script menu

```
display(’log’)    # set up log file
check(’c:/drug-MC’)     # check syntax of model
#  data(’c:/drug-data’) # load data file if there is one
compile(1)        # generate code for 1 simulations
# inits(1,’c:/drug-in1’)# load initial values if necessary
gen.inits()       # generate initial values for all unknown
                  # quantities not given initial values
set(theta)        # monitor the true response rate
set(y)            # monitor the predicted number of successes
set(P.crit)       # monitor indicator of critical success rate
trace(*)          # watch some simulated values
update(10000)     # perform 10000 simulations
history(theta)    # Trace plot of samples for theta
stats(*)          # Calculate summary statistics
                  # for all monitored quantities
density(theta)    # Plot distribution of theta
density(y)        # Plot distribution of y
```

Example: Using Monte Carlo methods to allow uncertainty in a power calculation

- a randomised trial planned with n patients in each of two arms
- response with standard deviation $\sigma = 1$
- aimed to have Type 1 error 5% and 80% power
- to detect a true difference of θ = 0.5 in mean response between the groups

Necessary sample size per group is

$$n = \frac{2\sigma^2}{\theta^2}(0.84 + 1.96)^2 = 63$$

Alternatively, for fixed n, the power is

$$\text{Power} = \Phi\left(\sqrt{\frac{n\theta^2}{2\sigma^2}} - 1.96\right).$$

Example: Uncertainty in a power calculation

Suppose we wish to express uncertainty concerning both θ and σ, e.g.

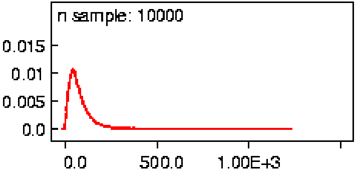
$$\theta \sim \mathrm{N}(0.5, 0.1^2), \quad \sigma \sim \mathrm{N}(1, 0.3^2).$$

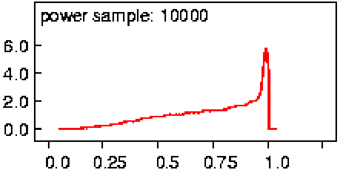
1. Simulate values of θ and σ from their prior distributions
2. Substitute them in the formulae
3. Obtain a predictive distribution over n or Power

```
prec.sigma  <- 1/(0.3*0.3) # transform to precision=1/sd2
prec.theta  <- 1/(0.1*0.1)
sigma      ~ dnorm(1, prec.sigma)I(0,)
theta      ~ dnorm(.5, prec.theta)I(0,)
n      <- 2 * pow( (.84 +1.96) * sigma / theta , 2)
power  <- phi(  sqrt(63/2)* theta /sigma  -1.96  )
prob70 <-step(power-.7)
```

Example: Uncertainty in a power calculation

	Median	95% interval
n	62.5	9.3 to 247.2
Power (%)	80	29 to 100





- For *n*= 63, the median power is 80%, and a trial of 63 patients per group could be seriously underpowered
- There is a 37% chance that the power is less than 70%

WinBUGS Demo

- Getting started, manuals and examples
- Running a model using GUI interface
  - Checking and compiling model code
  - Running simulations (updates)
  - Trace/history plots
  - Obtaining summary statistics and density plots
  - Obtaining sampled values
- Running a model using scripts

Lecture 2.
Introduction to conjugate Bayesian inference

Outline

- What are Bayesian methods?
- Bayes theorem and its link with Bayesian inference
- Prior, likelihood and posterior distributions
- Conjugate Bayesian inference for binomial, Normal and count data

What are Bayesian methods?

- Bayesian methods have been widely applied in many areas:
  - medicine / epidemiology
  - genetics
  - ecology
  - environmental sciences
  - social and political sciences
  - finance
  - archaeology
  - .....
- Motivations for adopting Bayesian approach vary:
  - natural and coherent way of thinking about science and learning
  - pragmatic choice that is suitable for the problem in hand

What are Bayesian methods?

Spiegelhalter et al (2004) define a Bayesian approach as

> *‘the explicit use of external evidence in the design, monitoring, analysis, interpretation and reporting of a [scientific investigation]’*

They argue that a Bayesian approach is:

- more flexible in adapting to each unique situation
- more efficient in using all available evidence
- more useful in providing relevant quantitative summaries

than traditional methods

Example

A clinical trial is carried out to collect evidence about an unknown ‘treatment effect’

Conventional analysis

- p-value for H_0: treatment effect is zero
- Point estimate and CI as summaries of size of treatment effect

Aim is to learn what this trial tells us about the treatment effect

Bayesian analysis

- Inference is based on probability statements summarising the posterior distribution of the treatment effect

Asks: ‘how should this trial change our opinion about the treatment effect?’

Components of a Bayesian analysis

The Bayesian analyst needs to explicitly state

- a reasonable opinion concerning the plausibility of different values of the treatment effect *excluding* the evidence from the trial (the **prior distribution**)
- the support for different values of the treatment effect based *solely* on data from the trial (the **likelihood**),

and to combine these two sources to produce

- a final opinion about the treatment effect (the **posterior distribution**)

The final combination is done using Bayes theorem (and only simple rules of probability), which essentially weights the likelihood from the trial with the relative plausibilities defined by the prior distribution

One can view the Bayesian approach as a formalisation of the process of learning from experience

Bayesian inference: the posterior distribution

Posterior distribution forms basis for all inference — can be summarised to provide

- point and interval estimates of Quantities of Interest (QOI), e.g. treatment effect, small area estimates, ...
- point and interval estimates of any function of the parameters
- probability that QOI (e.g. treatment effect) exceeds a critical threshold
- prediction of QOI in a new unit
- prior information for future experiments, trials, surveys, ...
- inputs for decision making
- ....

Bayes theorem and its link with Bayesian inference

Bayes' theorem

- Provable from probability axioms
- Let A and B be events, then

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}.$$

- If A_i is a set of mutually exclusive and exhaustive events (*i.e.* $p(\bigcup_i A_i) = \sum_i p(A_i) = 1$), then

$$p(A_i|B) = \frac{p(B|A_i)p(A_i)}{\sum_j p(B|A_j)p(A_j)}.$$

Example: use of Bayes theorem in diagnostic testing

A new HIV test is claimed to have “95% sensitivity and 98% specificity".

In a population with an HIV prevalence of 1/1000, what is the chance that patient testing positive actually has HIV?

- Let A be the event that patient is truly HIV positive, $\overline{A}$ be the event that they are truly HIV negative.
- Let B be the event that they test positive.
- We want $p(A|B)$.
- “95% sensitivity” means that $p(B|A) = .95$.
- “98% specificity” means that $p(B|\overline{A}) = .02$.
- Now Bayes theorem says

$$p(A|B) = \frac{p(B|A)p(A)}{p(B|A)p(A) + p(B|\overline{A})p(\overline{A})}.$$

- Hence $p(A|B) = \frac{.95 \times .001}{.95 \times .001 + .02 \times .999} = .045$.
- Thus over 95% of those testing positive will, in fact, not have HIV.

Comments

- Our intuition is poor when processing probabilistic evidence
- The vital issue is *how should this test result change our belief that patient is HIV positive?*
- The disease prevalence can be thought of as a *'prior'* probability ($p = 0.001$)
- Observing a positive result causes us to modify this probability to $p = 0.045$. This is our *'posterior'* probability that patient is HIV positive.
- Bayes theorem applied to *observables* (as in diagnostic testing) is uncontroversial and established
- More controversial is the use of Bayes theorem in general statistical analyses, where *parameters* are the unknown quantities, and their prior distribution needs to be specified — this is **Bayesian inference**

Bayesian inference

Makes fundamental distinction between

- Observable quantities x, i.e. the data
- Unknown quantities θ

θ can be statistical parameters, missing data, mismeasured data ...
$\rightarrow$ parameters are treated as random variables

$\rightarrow$ in the Bayesian framework, we make probability statements about model parameters

! in the frequentist framework, parameters are fixed non-random quantities and the probability statements concern the data

Bayesian inference (continued)

As with any analysis, we start by positing a model, $p(x \mid \theta)$

This is the **likelihood**, which relates all variables into a **'full probability model'**

From a Bayesian point of view

- θ is unknown so should have a **probability distribution** reflecting our uncertainty about it before seeing the data
  $\rightarrow$ need to specify a **prior distribution** $p(\theta)$
- x is known so we should condition on it
  $\rightarrow$ use Bayes theorem to obtain conditional probability distribution for unobserved quantities of interest given the data:

$$p(\theta \mid x) = \frac{p(\theta)\, p(x \mid \theta)}{\int p(\theta)\, p(x \mid \theta)\, d\theta} \propto p(\theta)\, p(x \mid \theta)$$

This is the **posterior distribution**

Bayesian inference (continued)

- The prior distribution $p(\theta)$, expresses our uncertainty about θ **before** seeing the data.
- The posterior distribution $p(\theta \mid x)$, expresses our uncertainty about θ **after** seeing the data.

Example: Inference on proportions

- Suppose we observe r positive responses out of n patients.
- Assuming patients are independent, with common unknown response rate θ, leads to a binomial likelihood

$$p(r|n,\theta) \quad = \quad \left(\begin{array}{c} n \\ r \end{array}\right) \theta^r (1-\theta)^{n-r} \ \propto \ \theta^r (1-\theta)^{n-r}$$

- θ needs to be given a continuous prior distribution.
- Suppose that, before taking account of the evidence from our study, we believe all values for θ are equally likely (is this plausible?) $\Rightarrow \theta \sim \text{Unif}(0,1)$ i.e. $p(\theta) = \frac{1}{1-0} = 1$
- Posterior is then

$$p(\theta|r,n) \propto \theta^r (1-\theta)^{(n-r)} \times 1$$

- This has form of the *kernel* of a Beta(r+1, n-r+1) distribution, where

$$\theta \quad \sim \quad \text{Beta}(a,b) \ \equiv \ \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{a-1}(1-\theta)^{b-1}$$

Example: Inference on proportions (continued)

To represent external evidence that some response rates are more plausible than others, it is mathematically convenient to use a Beta(a, b) prior distribution for θ

$$p(\theta) \propto \theta^{a-1}(1-\theta)^{b-1}$$

Combining this with the binomial likelihood gives a posterior distribution

$$\begin{aligned} p(\theta \mid r, n) &\propto p(r \mid \theta, n) p(\theta) \\ &\propto \theta^{r}(1-\theta)^{n-r}\theta^{a-1}(1-\theta)^{b-1} \\ &= \theta^{r+a-1}(1-\theta)^{n-r+b-1} \\ &\propto \text{Beta}(r+a,\ n-r+b) \end{aligned}$$

Comments

- When the prior and posterior come from the same family of distributions the prior is said to be **conjugate** to the likelihood
  - Occurs when prior and likelihood have the same 'kernel'
- Recall from lecture 1 that a Beta(a, b) distribution has

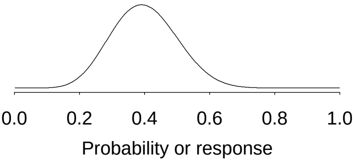
$$\begin{aligned} \text{mean} &= a/(a+b), \\ \text{variance} &= ab/\left[(a+b)^2(a+b+1)\right] \end{aligned}$$

  Hence posterior mean is $E(\theta|r,n) = (r+a)/(n+a+b)$
- a and b are equivalent to observing a priori $a-1$ successes in $a+b-2$ trials → can be elicited
- With fixed a and b, as r and n increase, $E(\theta|r,n) \to r/n$ (the MLE), and the variance tends to zero
  - This is a general phenomenon: as n increases, posterior distribution gets more concentrated and the likelihood dominates the prior
- A Beta(1, 1) is equivalent to Uniform(0, 1)

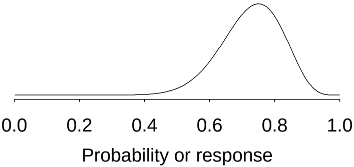
Example: Drug

- Recall example from lecture 1, where we consider early investigation of a new drug
- Experience with similar compounds has suggested that response rates between 0.2 and 0.6 could be feasible
- We interpreted this as a distribution with mean = 0.4, standard deviation 0.1 and showed that a Beta(9.2,13.8) distribution has these properties
- Suppose we now treat $n = 20$ volunteers with the compound and observe $y = 15$ positive responses

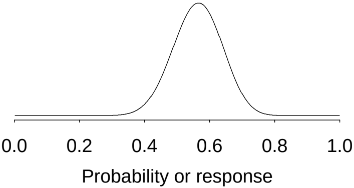
Prior, likelihood and posterior for Drug example



Beta(9.2, 13.8) prior distribution supporting response rates between 0.2 and 0.6

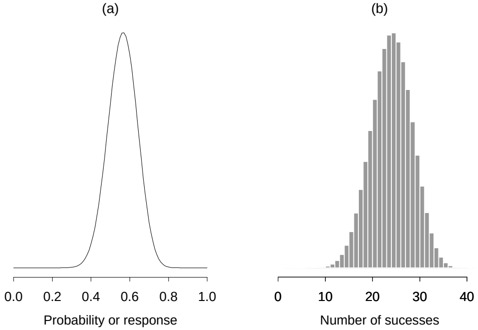


Likelihood arising from a Binomial observation of 15 successes out of 20 cases



Parameters of the Beta distribution are updated to (a+15, b+20-15) = (24.2, 18.8): mean 24.2/(24.2+18.8) = 0.56

Posterior and predictive distributions for Drug example



(a) Beta posterior after having observed 15 successes in 20 trials
(b) predictive Beta-Binomial distribution of the number of successes $\tilde{y}_{40}$ in the next 40 trials with mean 22.5 and standard deviation 4.3

- Suppose we would consider continuing a development program if the drug managed to achieve at least a further 25 successes out of these 40 future trials

Drug (continued): learning about parameters from data using Markov chain Monte-Carlo (MCMC) methods

- In the Drug example so far, we have calculated the posterior (and predictive) distributions in closed form
  - this is possible because we are using conjugate priors
  - means that we can make use of known properties of the closed-form posterior distribution to make inference, e.g. expected value (posterior mean), tail-area probabilities are known analytically
- Using MCMC (e.g. in WinBUGS), no need to explicitly specify posterior
- Can just specify the prior and likelihood separately
- WinBUGS contains algorithms to evaluate (and summarise) the posterior given (almost) arbitrary specification of prior and likelihood
  - posterior doesn't need to be closed form
  - but can (usually) recognise conjugacy when it exists

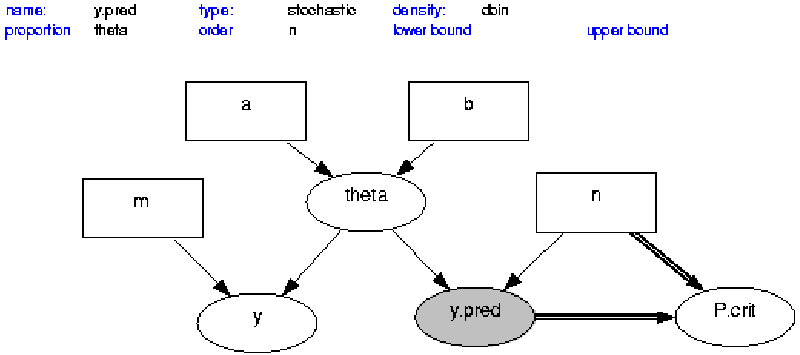
The drug model can be written

$$\begin{array}{rcll} \theta & \sim & \text{Beta}[a, b] & \text{prior distribution} \\ y & \sim & \text{Binomial}[\theta, m] & \text{sampling distribution} \\ y_{\text{pred}} & \sim & \text{Binomial}[\theta, n] & \text{predictions} \\ P_{\text{crit}} & = & P(y_{\text{pred}} \geq n_{\text{crit}}) & \text{Probability of exceeding critical threshold} \end{array}$$

```
# In BUGS syntax:

# Model description '
model {
 theta   ~ dbeta(a,b)              # prior distribution
 y       ~ dbin(theta,m)           # sampling dist
 y.pred  ~ dbin(theta,n)           # predictions
 P.crit <- step(y.pred-ncrit+0.5) # =1 if y.pred>=ncrit,
                                   # =0 otherwise
}
```

Graphical representation of Drug model



name: y.pred type: stochastic density: dbin
proportion theta order n lower bound upper bound

Note that adding data to a model is simply extending the graph.

WinBUGS output and exact answers

```
node     mean    sd     MC error   2.5% median 97.5% start sample
theta     0.56  0.074 4.292E-4  0.41   0.56   0.70 1001  30000
y.pred  22.52  4.278 0.02356  14.00 23.00  31.0  1001  30000
P.crit    0.32  0.469 0.002631  0.00   0.00   1.0  1001  30000
```

Exact answers from conjugate analysis

- θ: mean 0.563 and standard deviation 0.075
- Y^{pred}: mean 22.51 and standard deviation 4.31.
- Probability of at least 25: 0.329

MCMC results are within Monte Carlo (sampling) error of the true values

Bayesian inference using the Normal distribution

Known variance, unknown mean

- Suppose we have a sample of Normal data $x_i \sim \mathrm{N}(\theta, \sigma^2) \quad (i = 1, ..., n)$.
- For now assume σ^2 is known and θ has a Normal prior $\theta \sim \mathrm{N}(\mu, \sigma^2/n_0)$
  - Same standard deviation σ is used in the likelihood and the prior.
  - Prior variance is based on an 'implicit' sample size n_0
- Then straightforward to show that the posterior distribution is

$$\theta|\boldsymbol{x} \sim \mathrm{N}\left(\frac{n_0\mu + n\overline{x}}{n_0 + n}, \frac{\sigma^2}{n_0 + n}\right)$$

Bayesian inference using the Normal distribution

- As n_0 tends to 0, the prior variance becomes larger and the distribution becomes ‘flatter’, and in the limit the prior distribution becomes essentially uniform over $-\infty, \infty$
- Posterior mean $(n_0\mu + n\overline{x})/(n_0 + n)$ is a weighted average of the prior mean μ and parameter estimate $\overline{x}$, weighted by their precisions (relative ‘sample sizes’), and so is always a compromise between the two
- Posterior variance is based on an implicit sample size equivalent to the sum of the prior ‘sample size’ n_0 and the sample size of the data n
- As $n \rightarrow \infty$, $p(\theta|\boldsymbol{x}) \rightarrow \mathrm{N}(\overline{x}, \sigma^2/n)$ which does not depend on the prior
- Compare with frequentist setting, the MLE is $\hat{\theta} = \bar{x}$ with $\mathrm{SE}(\hat{\theta}) = \sigma/\sqrt{n}$, and sampling distribution

$$p(\hat{\theta} \mid \theta) = p(\bar{x}|\theta) = \mathrm{N}(\theta, \, \sigma^2/n)$$

Example: THM concentrations

- Regional water companies in the UK are required to take routine measurements of trihalomethane (THM) concentrations in tap water samples for regulatory purposes
- Samples tested throughout year in each water supply zone
- Suppose we want to estimate the average THM concentration in a particular water zone, z
- Two independent measurements, x_{z1} and x_{z2} are taken and their mean, $\overline{x}_z$ is 130 $\mu g/l$
- Suppose we know that the assay measurement error has a standard deviation $\sigma = 5 \mu g/l$
- What should we estimate the mean THM concentration to be in this water zone?

Let the mean THM concentration be denoted θ_z.

Standard analysis would use sample mean $\overline{x}_z$ = $130 \mu g/l$ as an estimate of θ_z, with standard error $\sigma/\sqrt{n} = 5/\sqrt{2} = 3.5 \mu g/l$

95% CI: $\overline{x}_z \pm 1.96 \times \sigma/\sqrt{n}$, i.e. 123.1 to 136.9 $\mu g/l$.

THM example (continued)

Suppose historical data on THM levels in other zones supplied from the same source showed that the mean THM concentration was 120 $\mu g/l$ with standard deviation 10 $\mu g/l$

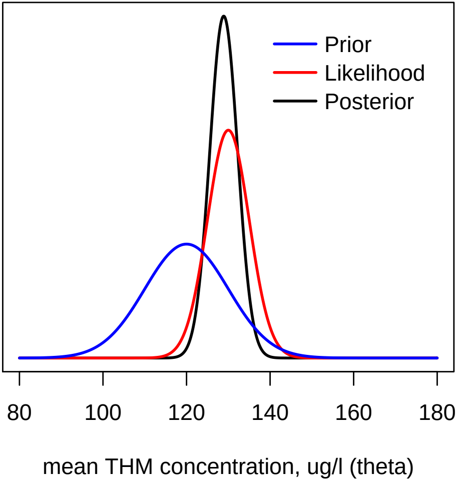
- suggests Normal$(120, 10^2)$ prior for θ_z
- if we express the prior standard deviation as $\sigma/\sqrt{n_0}$, we can solve to find $n_0 = (\sigma/10)^2$ = 0.25
- so our prior can be written as $\theta_z \sim$ Normal$(120, \sigma^2/0.25)$

Posterior for θ_z is then

$$\begin{aligned} p(\theta_z|\boldsymbol{x}) &= \text{Normal}\left(\frac{0.25 \times 120 + 2 \times 130}{0.25 + 2}, \ \frac{5^2}{0.25 + 2}\right) \\ &= \text{Normal}(128.9, \ \ 3.33^2) \end{aligned}$$

giving 95% interval for θ_z of 122.4 to 135.4$\mu g/l$

Prior, likelihood and posterior for THM example



Prediction

Denoting the posterior mean and variance as $\mu_n = (n_0\mu + n\overline{x})/(n_0 + n)$ and $\sigma_n^2 = \sigma^2/(n_0 + n)$, the *predictive distribution* for a new observation $\tilde{x}$ is

$$p(\tilde{x}|\mathbf{x}) = \int p(\tilde{x}|\mathbf{x}, \theta)p(\theta|\mathbf{x})d\theta$$

which generally simplifies to

$$p(\tilde{x}|\mathbf{x}) = \int p(\tilde{x}|\theta)p(\theta|\mathbf{x})d\theta$$

which can be shown to give

$$p(\tilde{x}|\mathbf{x}) \sim \mathrm{N}\left(\mu_n, \sigma_n^2 + \sigma^2\right)$$

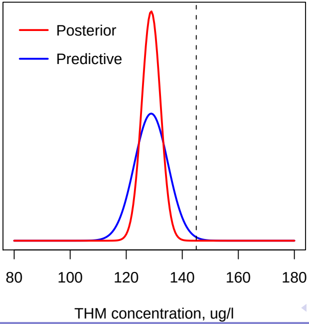
So the predictive distribution is centred around the posterior mean with variance equal to sum of the posterior variance and the sample variance of $\tilde{x}$

Example: THM concentration (continued)

- Suppose the water company will be fined if THM levels in the water supply exceed $145 \mu g/l$
- Predictive distribution for THM concentration in a future sample taken from the water zone is

$$N(128.9, 3.33^2 + 5^2) = N(128.9, 36.1)$$

- Probability that THM concentration in future sample exceeds $145 \mu g/l$ is $1 - \Phi[(145 - 128.9)/\sqrt(36.1)] = 0.004$



Bayesian inference using count data

Suppose we have an independent sample of counts $x_1, ..., x_n$ which can be assumed to follow a Poisson distribution with unknown mean μ:

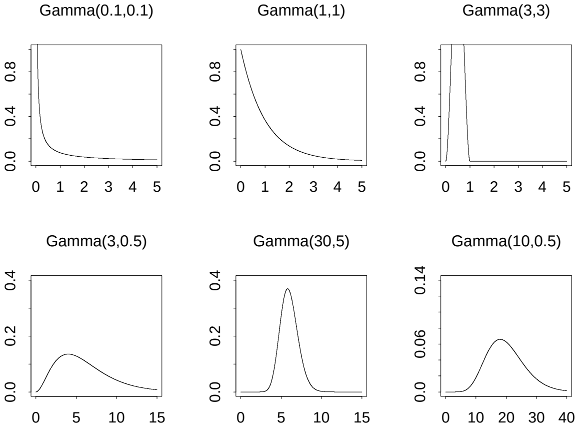
$$p(\boldsymbol{x}|\mu) \;=\; \prod_i \frac{\mu^{x_i} e^{-\mu}}{x_i!}$$

The conjugate prior for the mean of a Poisson distribution is a Gamma distribution:

$$p(\mu) = Gamma(a, b) \;=\; \frac{b^a}{\Gamma(a)} \mu^{a-1} e^{-b\mu}$$

Recall from lecture 1 that a Gamma(a, b) density has mean a/b and variance a/b^2

Some Gamma distributions



Bayesian inference using count data (continued)

This implies the following posterior

$$\begin{aligned} p(\mu \mid \boldsymbol{x}) &\propto p(\mu)\, p(\boldsymbol{x} \mid \mu) \\ &= \frac{b^a}{\Gamma(a)} \mu^{a-1} e^{-b\mu} \prod_{i=1}^{n} e^{-\mu} \frac{\mu^{x_i}}{x_i!} \\ &\propto \mu^{a+n\overline{x}-1}\, e^{-(b+n)\mu} \\ &= \mathrm{Gamma}(a+n\overline{x},\, b+n). \end{aligned}$$

The posterior is another (different) Gamma distribution.

$$E(\mu \mid \boldsymbol{x}) = \frac{a+n\overline{x}}{b+n} = \overline{x}\left(\frac{n}{n+b}\right) + \frac{a}{b}\left(1 - \frac{n}{n+b}\right)$$

So posterior mean is a compromise between the prior mean a/b and the MLE $\overline{x}$

Example: Estimation of disease risk in a single area

Often interested in estimating the **rate** or **relative risk** rather than the **mean** for Poisson data:

- Suppose we observe $x = 5$ cases of leukaemia in one region, with age-sex-standardised expected number of cases E = 2.8
- Assume Poisson likelihood for x with mean $\mu = \lambda \times E$, where λ is the unknown relative risk:

$$p(x|\lambda, E) = \frac{(\lambda E)^x e^{-\lambda E}}{x!}$$

- Assume Gamma(a, b) prior for the relative risk λ:

$$p(\lambda) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}$$

- Posterior for λ is then

$$\begin{aligned} p(\lambda|x, E) &\propto \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda} \frac{(\lambda E)^x e^{-\lambda E}}{x!} \\ &\propto \lambda^{a+x-1} e^{-(b+E)\lambda} \propto \text{Gamma}(a+x, b+E) \end{aligned}$$

Disease risk example: Vague prior

Suppose we wish to express vague prior information about λ

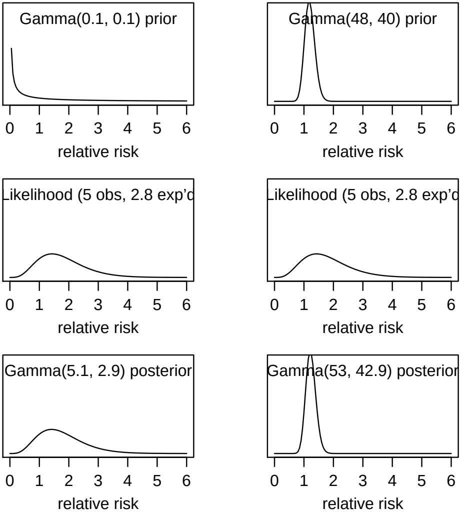
- A Gamma(0.1, 0.1) distribution represents a prior for the relative risk λ with
  - mean 0.1/0.1 = 1
  - variance $0.1/0.1^2 = 10$
  - 95^{th} percentile = 5.8
- This gives a posterior $p(\lambda|x) = \text{Gamma}(5.1, 2.9)$
- This has posterior mean $= 5.1/2.9 = 1.76$
  (cf MLE = x/E = 5/2.8 = 1.78)

Disease risk example: Informative prior

Alternatively, we may have strong prior information to suggest that the relative risk in the region is probably around 1.2, and has only a 5% probability of being higher than 1.5

- A Gamma(48, 40) distribution represents a prior for the relative risk λ with
  - mean 48/40 = 1.2
  - 95th percentile = 1.5
- This gives a posterior $p(\lambda|x) = \text{Gamma}(53, 42.8)$
- This has posterior mean $= 53/42.9 = 1.24$

Prior, likelihood and posterior for disease risk example: Vague (left) and Informative (right) priors



Comments

For all these examples, we see that

- the posterior mean is a compromise between the prior mean and the MLE
- the posterior s.d. is less than each of the prior s.d. and the s.e.(MLE)

> *'A Bayesian is one who, vaguely expecting a horse and catching a glimpse of a donkey, strongly concludes he has seen a mule' (Senn, 1997)*

As $n \to \infty$,

- the posterior mean $\to$ the MLE
- the posterior s.d. $\to$ the s.e.(MLE)
- the posterior does not depend on the prior.

These observations are generally true, when the MLE exists and is unique

Choosing prior distributions

When the posterior is in the same family as the prior then we have what is known as *conjugacy*. This has the advantage that prior parameters can usually be interpreted as a *prior sample*. Examples include:

Likelihood	Parameter	Prior	Posterior
Normal	mean	Normal	Normal
Normal	precision	Gamma	Gamma
Binomial	success prob.	Beta	Beta
Poisson	rate or mean	Gamma	Gamma

- Conjugate prior distributions are mathematically convenient, but do not exist for all likelihoods, and can be restrictive
- Computations for non-conjugate priors are harder, but possible using MCMC (see next lecture)

Lecture 3.
Introduction to Markov Chain Monte Carlo methods

Outline

- Why do we need simulation methods for Bayesian inference?
- Sampling from posterior distributions using Markov chains
- Gibbs sampling
- Checking convergence of the MCMC simulations
- Checking efficiency of the MCMC simulations
- Making inference using samples from the posterior distribution
- WINBUGS demo

Why is computation important?

- Bayesian inference centres around the posterior distribution

$$p(\boldsymbol{\theta}|x) \propto p(x|\boldsymbol{\theta}) \times p(\boldsymbol{\theta})$$

  where $\boldsymbol{\theta}$ is typically a large vector of parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, ..., \theta_k\}$

- $p(x|\boldsymbol{\theta})$ and $p(\boldsymbol{\theta})$ will often be available in closed form, but $p(\boldsymbol{\theta}|x)$ is usually not analytically tractable, and we want to
  - obtain marginal posterior $p(\theta_i|x) = \int\int ... \int p(\boldsymbol{\theta}|x)\, d\boldsymbol{\theta}_{(-i)}$ where $\boldsymbol{\theta}_{(-i)}$ denotes the vector of θ's excluding θ_i
  - calculate properties of $p(\theta_i|x)$, such as mean $(= \int \theta_i p(\theta_i|x) d\theta_i)$, tail areas $(= \int_T^\infty p(\theta_i|x) d\theta_i)$ etc.

→ numerical integration becomes vital

Monte Carlo integration

- We have already seen that Monte Carlo methods can be used to simulate values from prior distributions and from **closed form** posterior distributions
- If we had algorithms for sampling from arbitrary (typically high-dimensional) posterior distributions, we could use Monte Carlo methods for Bayesian estimation, as follows

Monte Carlo integration (continued)

- Suppose we can draw samples from the joint posterior distribution for $\boldsymbol{\theta}$, *i.e.*

$$(\theta_1^{(1)}, ..., \theta_k^{(1)}), (\theta_1^{(2)}, ..., \theta_k^{(2)}), ..., (\theta_1^{(N)}, ..., \theta_k^{(N)}) \quad \sim \quad p(\boldsymbol{\theta}|x)$$

- Then
  - $\theta_1^{(1)}, ..., \theta_1^{(N)}$ are a sample from the marginal posterior $p(\theta_1|x)$
  - $E(g(\theta_1)) \; = \; \int g(\theta_1)p(\theta_1|x)d\theta_1 \; \approx \; \frac{1}{N}\sum_{i=1}^{N} g(\theta_1^{(i)})$

→ this is Monte Carlo integration

→ theorems exist which prove convergence in limit as $N \rightarrow \infty$ even if the sample is dependent (crucial to the success of MCMC)

How do we sample from the posterior?

- We want samples from joint posterior distribution $p(\boldsymbol{\theta}|x)$
- *Independent* sampling from $p(\boldsymbol{\theta}|x)$ may be difficult
- **BUT** *dependent* sampling from a *Markov chain* with $p(\boldsymbol{\theta}|x)$ as its stationary (equilibrium) distribution is easier
- A sequence of random variables $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, ...$ forms a Markov chain if $\theta^{(i+1)} \sim p(\theta|\theta^{(i)})$
  i.e. conditional on the value of $\theta^{(i)}$, $\theta^{(i+1)}$ is independent of $\theta^{(i-1)}, ..., \theta^{(0)}$

Sampling from the posterior using Markov chains

Several standard ‘recipes’ available for designing Markov chains with required stationary distribution $p(\theta|x)$

- Metropolis *et al.* (1953); generalised by Hastings (1970)
- **Gibbs Sampling** (see Geman and Geman (1984), Gelfand and Smith (1990), Casella and George (1992)) is a special case of the Metropolis-Hastings algorithm which generates a Markov chain by sampling from **full conditional distributions**
- See Gilks, Richardson and Spiegelhalter (1996) for a full introduction and many worked examples

Gibbs sampling

Let our vector of unknowns $\boldsymbol{\theta}$ consist of k sub-components $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_k)$

1) Choose starting values $\theta_1^{(0)}$, $\theta_2^{(0)}$, ..., , $\theta_k^{(0)}$

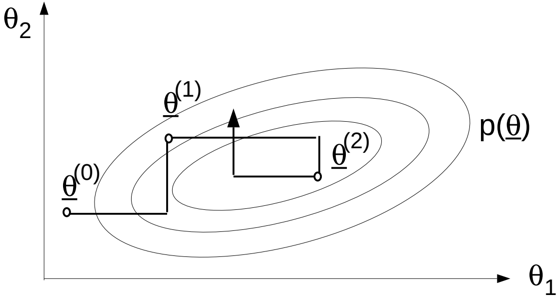
2) Sample $\theta_1^{(1)}$ from $p(\theta_1 | \theta_2^{(0)}, \theta_3^{(0)}, ..., , \theta_k^{(0)}, x)$
   Sample $\theta_2^{(1)}$ from $p(\theta_2 | \theta_1^{(1)}, \theta_3^{(0)}, ..., , \theta_k^{(0)}, x)$
   .....
   Sample $\theta_k^{(1)}$ from $p(\theta_k | \theta_1^{(1)}, \theta_2^{(1)}, ..., , \theta_{k-1}^{(1)}, x)$

3) Repeat step 2 many 1000s of times
   - eventually obtain sample from $p(\boldsymbol{\theta}|x)$

The conditional distributions are called ‘full conditionals’ as they condition on all other parameters

Gibbs sampling continued

Example with $k = 2$



- Sample $\theta_1^{(1)}$ from $p(\theta_1|\theta_2^{(0)}, x)$
- Sample $\theta_2^{(1)}$ from $p(\theta_2|\theta_1^{(1)}, x)$
- Sample $\theta_1^{(2)}$ from $p(\theta_1|\theta_2^{(1)}, x)$
- ......

$\boldsymbol{\theta}^{(n)}$ forms a Markov chain with (*eventually*) a stationary distribution $p(\boldsymbol{\theta}|x)$.

Running WinBUGS on the Drug example

Recall how the drug model is written

$$\begin{array}{rcll}
\theta & \sim & \text{Beta}[a, b] & \text{prior distribution} \\
y & \sim & \text{Binomial}[\theta, m] & \text{sampling distribution} \\
y_{\text{pred}} & \sim & \text{Binomial}[\theta, n] & \text{predictive distribution} \\
P_{\text{crit}} & = & P(y_{\text{pred}} \geq n_{\text{crit}}) & \text{Probability of exceeding critical threshold}
\end{array}$$

```
# In BUGS syntax:

model {
 theta  ~ dbeta(a,b)               # prior distribution
 y      ~ dbin(theta,m)            # sampling dist
 y.pred ~ dbin(theta,n)            # predictive dist
 P.crit <- step(y.pred-ncrit+0.5) # =1 if y.pred>=ncrit,
                                  # =0 otherwise
}
```

Data files

Data can be written after the model description, or held in a separate .txt or .odc file

```
list( a = 9.2,    # parameters of prior distribution
 b = 13.8,
 y = 15,     # number of successes
 m = 20,     # number of trials
 n = 40,     # future number of trials
ncrit = 25)  # critical value of future successes
```

Alternatively, in this simple example, we could have put all data and constants into model description:

```
model{
 theta  ~ dbeta(9.2,13.8)    # prior distribution
 y      ~ dbin(theta,20)     # sampling dist
 y.pred ~ dbin(theta,40)     # predictive dist
 P.crit <- step(y.pred-24.5) # =1 if y.pred>=24.5,
                             # =0 otherwise
 y      <- 15
}
```

The WINBUGS data formats

WINBUGS accepts data files in:

1. Rectangular format (easy to cut and paste from spreadsheets)

```
 n[] r[]
 47  0
148 18
 ...
360 24
END
```

2. S-Plus format:

```
list(N=12,n = c(47,148,119,810,211,196,
                148,215,207,97,256,360),
          r = c(0,18,8,46,8,13,9,31,14,8,29,24))
```

Generally need a ‘list’ to specify values of scalar quantities like the size of dataset (N) etc.

Initial values

- WinBUGS can automatically generate initial values for the MCMC analysis using *gen inits*
- Fine if have informative prior information
- If have fairly ‘vague’ priors, better to provide reasonable values in an initial-values list

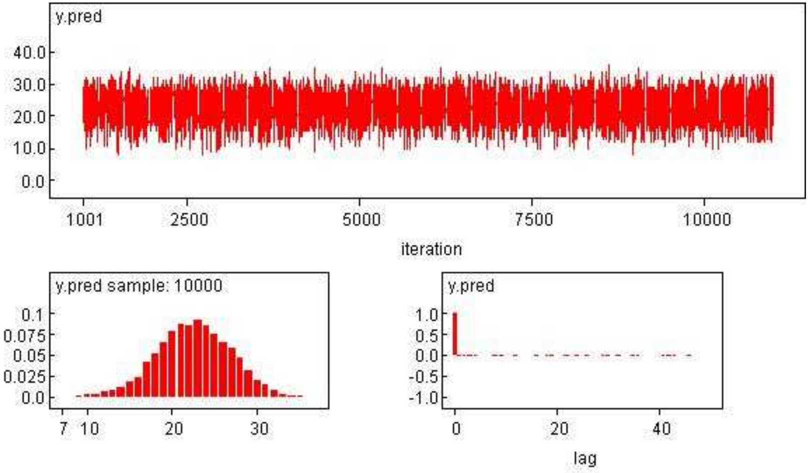
Initial values list can be after model description or in a separate file

```
list(theta=0.1)
```

Running WinBUGS for MCMC analysis (single chain)

1. Open *Specification tool* from *Model* menu.
2. Program responses are shown on bottom-left of screen.
3. Highlight `model` by double-click. Click on *Check model*.
4. Highlight start of data. Click on *Load data*.
5. Click on *Compile*.
6. Highlight start of initial values. Click on *Load inits*.
7. Click on *Gen Inits* if more initial values needed.
8. Open *Samples* from *Inference* menu.
9. Type names of nodes to be monitored into *Sample Monitor*, and click *set* after each.
10. Open *Update* from *Model* menu, enter required number of updates then click on *Update*
11. Check convergence and perform more updates if necessary
12. Type * into *Sample Monitor*, discard burn-in and click *stats* etc. to see results on all monitored nodes.

WinBUGS output



Using MCMC methods

There are two main issues to consider

1. Convergence
   - how quickly does the distribution of $\theta^{(t)}$ approach $p(\theta|x)$?
2. Efficiency
   - how well are functionals of $p(\theta|x)$ estimated from $\{\theta^{(t)}\}$?

Checking convergence

This is the users responsibility!

- Note: Convergence is to target **distribution** (the required posterior), not to a single value.
- Once convergence reached, samples should look like a random scatter about a stable mean value

Convergence diagnosis

- How do we know we have reached convergence?
- i.e. How do we the know number of ‘burn-in’ iterations?
- Many ‘convergence diagnostics’ exist, but none foolproof
- CODA and BOA software contain large number of diagnostics

Brooks-Gelman-Rubin (bgr) diagnostic

- Multiple ($\geq$ 2) runs
- Widely differing starting points
- Convergence assessed by quantifying whether sequences are much further apart than expected based on their internal variability
- Diagnostic uses components of variance of the multiple sequences

Example of checking convergence

Consider the following response rates for different doses of a drug

dose x_i	No. subjects n_i	No. responses r_i
1.69	59	6
1.72	60	13
1.75	62	18
1.78	56	28
1.81	63	52
1.83	59	53
1.86	62	61
1.88	60	60

Fit a logistic regression with ‘centred’ covariate $(x_i - \overline{x})$:

$$\begin{aligned} r_i &\sim \text{Binomial}(p_i, n_i) \\ \text{logit } p_i &= \alpha + \beta(x_i - \overline{x}) \\ \alpha &\sim \mathrm{N}(0, 10000) \\ \beta &\sim \mathrm{N}(0, 10000) \end{aligned}$$

Checking convergence with multiple runs

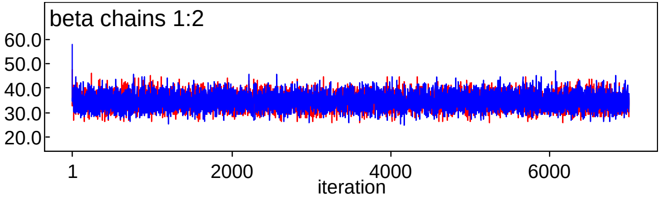
- Set up multiple initial value lists, e.g.

```
list(alpha=-100, beta=100)
list(alpha=100, beta=-100)
```

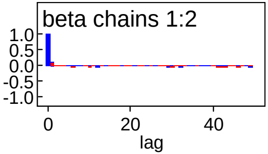
- Before clicking *compile*, set *num of chains* to 2
- Load both sets of initial values
- Monitor from the start of sampling
- Visually inspect trace/history plots to see if chains are overlapping
- Assess how much burn-in needed using the *bgr* statistic
- Check autocorrelation, as high autocorrelation is symptom of slow convergence

Output for ‘centred’ analysis

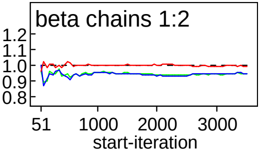
history



autocorrelation



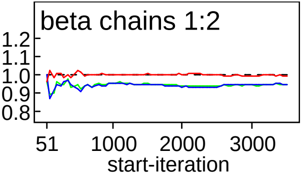
bgr diagnostic

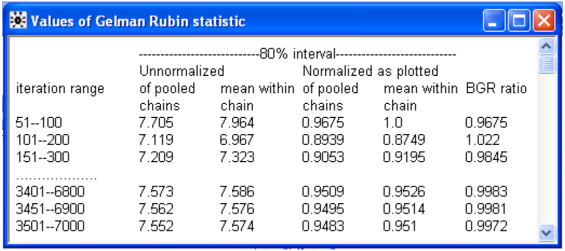


Discard first 1,000 iterations as burn-in

```
node   mean   sd     MC error 2.5%    median   97.5%   start  sample
beta   34.6   2.93   0.0298   29.17   34.54    40.6    1001   12000
```

BGR convergence diagnostic



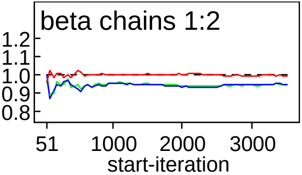


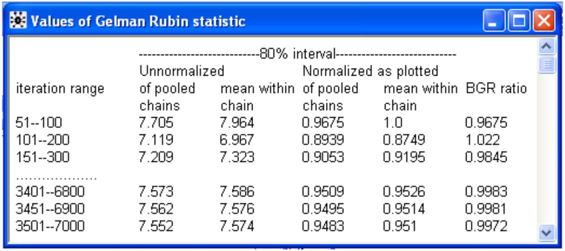
	----------------------------80% interval----------------------------				
	Unnormalized		Normalized as plotted		
iteration range	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
51--100	7.705	7.964	0.9675	1.0	0.9675
101--200	7.119	6.967	0.8939	0.8749	1.022
151--300	7.209	7.323	0.9053	0.9195	0.9845
....					
3401--6800	7.573	7.586	0.9509	0.9526	0.9983
3451--6900	7.562	7.576	0.9495	0.9514	0.9981
3501--7000	7.552	7.574	0.9483	0.951	0.9972

Interpreting the *bgr* statistics

- *Green*: width of 80% intervals of pooled chains: should be stable
- *Blue*: average width of 80% intervals for chains: should be stable
- *Red*: ratio of pooled/within: should be near 1

BGR convergence diagnostic





	---------------------------80% interval---------------------------				
	Unnormalized		Normalized as plotted		
iteration range	of pooled chains	mean within chain	of pooled chains	mean within chain	BGR ratio
51--100	7.705	7.964	0.9675	1.0	0.9675
101--200	7.119	6.967	0.8939	0.8749	1.022
151--300	7.209	7.323	0.9053	0.9195	0.9845
....					
3401--6800	7.573	7.586	0.9509	0.9526	0.9983
3451--6900	7.562	7.576	0.9495	0.9514	0.9981
3501--7000	7.552	7.574	0.9483	0.951	0.9972

Interpreting the *bgr* statistics (continued)

- WinBUGS 1.4.3 splits iterations into multiple overlapping intervals, calculates *bgr* statistics for each interval, and plots them against starting iteration of interval
  - approximate convergence can be ‘read off’ plot as iteration after which red *bgr* ratio line stablizes around 1, and blue and green 80% interval lines stablize to approximately constant value (not necessarily 1)
- Double-click on plot, then *ctrl* + right click gives values of statistics

WinBUGS Demo

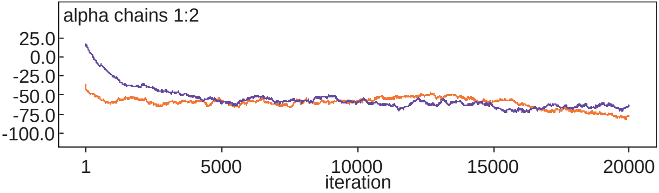
- Loading data files
- Loading multiple initial values files
- Visually inspecting trace plots
- bgr diagnostics
- Autocorrelation plots
- Discarding burn-in samples

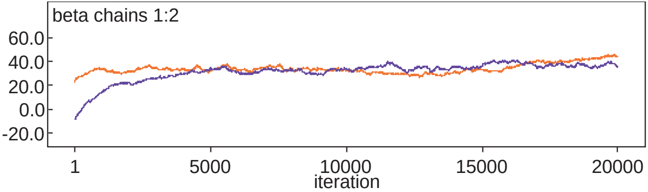
Problems with convergence

Fit a logistic curve with ‘un-centred’ covariate x:

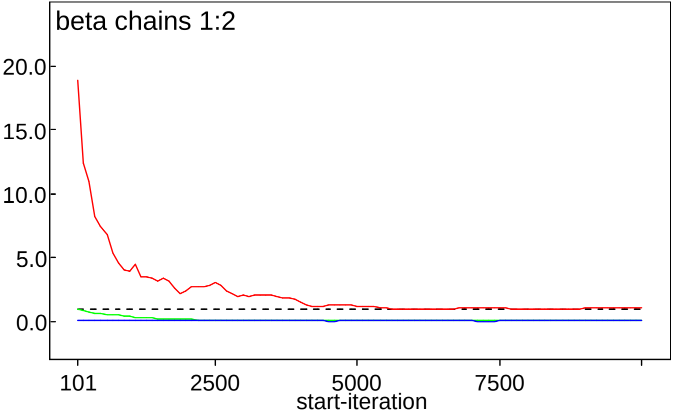
$$\begin{aligned} r_i &\sim \text{Binomial}(p_i, n_i) \\ \text{logit } p_i &= \alpha + \beta x_i \\ \alpha &\sim \mathrm{N}(0, 10000) \\ \beta &\sim \mathrm{N}(0, 10000) \end{aligned}$$

History plots for ‘un-centred’ analysis





bgr plot for uncentered analysis

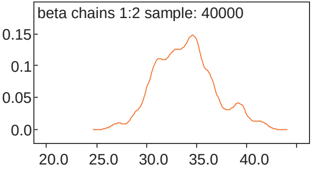


Discard first 10,000 iterations as burn-in

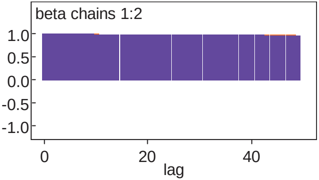
```
node   mean    sd     MC error 2.5%    median   97.5%   start sample
beta   33.36   3.00   0.2117   28.18   33.5     38.33   10001 20000
```

Output for ‘un-centred’ analysis

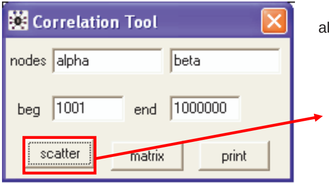
posterior density



autocorrelation



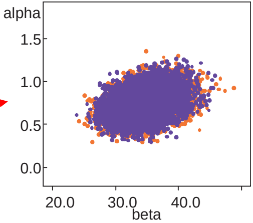
bivariate posteriors



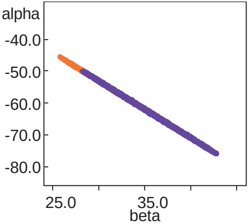
nodes alpha beta

beg 1001 end 1000000

centred



un-centred



How many iterations after convergence?

- After convergence, further iterations are needed to obtain samples for posterior inference.
- More iterations = more accurate posterior estimates.
- Efficiency of sample mean of θ as estimate of theoretical posterior expectation $E(\theta)$ usually assessed by calculating Monte Carlo standard error (MC error)
- MC error = standard error of posterior sample mean as estimate of theoretical expectation for given parameter
- MC error depends on
  - true variance of posterior distribution
  - posterior sample size (number of MCMC iterations)
  - autocorrelation in MCMC sample
- Rule of thumb: want MC error $< 1 - 5\%$ of posterior SD

Inference using posterior samples from MCMC runs

A powerful feature of the Bayesian approach is that all inference is based on the joint posterior distribution
$\Rightarrow$ can address wide range of substantive questions by appropriate summaries of the posterior

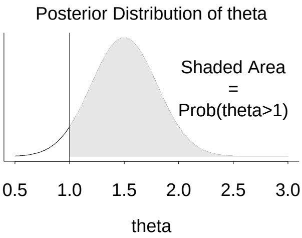
- Typically report either mean or median of the posterior samples for each parameter of interest as a point estimate
- 2.5% and 97.5% percentiles of the posterior samples for each parameter give a 95% posterior credible interval (interval within which the parameter lies with probability 0.95)

```
node   mean    sd     MC error 2.5%   median   97.5%  start  sample
beta   34.60   2.92   0.0239   29.11  34.53    40.51  1001   14000
```

So point estimate of `beta` would be 34.60, with 95% credible interval (29.11, 40.51)

Probability statements about parameters

- Classical inference cannot provide probability statements about parameters (e.g. p-value is not Pr(H_0 true), but probability of observing data as or more extreme than we obtained, given that H_0 is true)
- In Bayesian inference, it is simple to calculate e.g. Pr($\theta > 1$):
  - = Area under posterior distribution curve to the right of 1
  - = Proportion of values in posterior sample of θ which are > 1



- In `WinBUGS` use the `step` function:
  `p.theta <- step(theta - 1)`
- For discrete parameters, may also be interested in Pr($\delta = \delta_0$):
  `p.delta <- equals(delta, delta0)`
- Posterior means of `p.theta` and `p.delta` give the required probabilities

Complex functions of parameters

- Classical inference about a function of the parameters $g(\theta)$ requires construction of a specific estimator of $g(\theta)$. Obtaining appropriate error can be difficult.
- Easy using MCMC: just calculate required function $g(\theta)$ as a logical node at each iteration and summarise posterior samples of $g(\theta)$

In dose-response example, suppose we want to estimate the $ED95$: that is the dose that will provide 95% of maximum efficacy, i.e.

$$\begin{aligned} \text{logit}\, 0.95 &= \alpha + \beta(ED95 - \bar{x}) \\ ED95 &= (\text{logit }\, 0.95 - \alpha)/\beta + \bar{x} \end{aligned}$$

Simply add following line into BUGS model code:

```
ED95 <- (logit(0.95) - alpha)/beta + mean(x[])
```

Set monitor on `ED95`, update, and obtain summary statistics:

```
node   mean    sd      MC error  2.5%   median  97.5%  start  sample
ED95   1.857   0.007   8.514E-5  1.84   1.857   1.874  1001   10000
```

Functions of parameters: Ranks

- Recent trend in UK towards ranking ‘institutional’ performance e.g. schools, hospitals
- Might also want to rank treatments, answer ‘which is the best‘ etc
- Rank of a point estimate is a highly unreliable summary statistic

⇒ Would like measure of uncertainty about rank

- Bayesian methods provide *posterior interval estimates* for ranks
- WinBUGS contains ‘built-in’ options for ranks:
  - Rank option of `Inference` menu monitors the rank of the elements of a specified vector
  - `rank(x[], i)` returns the rank of the i^{th} element of x
  - `equals(rank(x[],i),1)` =1 if i^{th} element is ranked lowest, 0 otherwise. Mean is probability that i^{th} element is ’best’ (if counting adverse events)
  - `ranked(x[], i)` returns the value of the i^{th}-ranked element of x

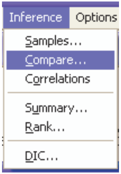
Example of ranking and posterior probabilities: ‘Blocker’ trials

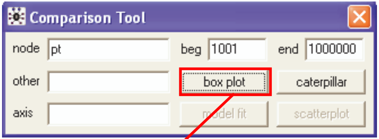
- 22 trials of beta-blockers used in WinBUGS manual to illustrate random-effects meta-analysis.
- Just consider treatment arm: which trial has the lowest mortality rate?
- For illustration, no random effects — just assume non-informative independent beta[0.5, 0.5] prior for each response rate.

```
for( i in 1 : Num) {
 rt[i] ~ dbin(pt[i],nt[i])
 pt[i] ~ dbeta(0.5,0.5)           # independent priors
 rnk[i] <-  rank(pt[], i)         # rank of i’th trial
 prob.lowest[i]<-equals(rnk[i],1) # prob i’th trial lowest
 N[i]<-i                          # used for indexing plot
}
```

Displaying posterior distribution of ranks

Mortality rates and ranks




Comparison Tool

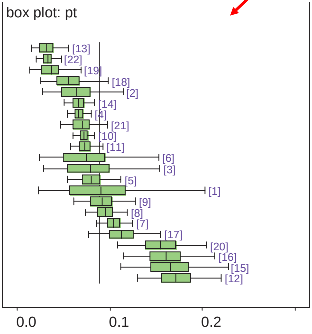
node pt

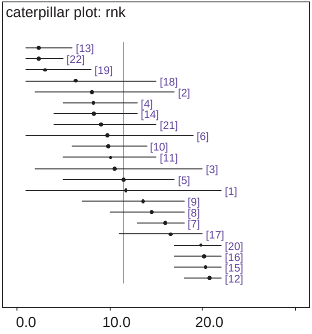
beg 1001

end 1000000

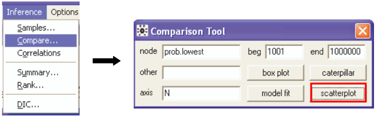
other

axis

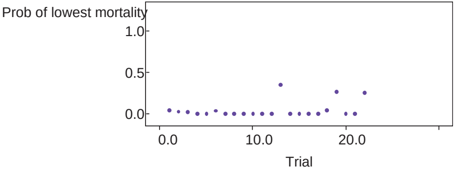

box plot: pt


caterpillar plot: rnk

Posterior probability that trial *i* has lowest mortality



node: prob.lowest
beg: 1001
end: 1000000
other:
axis: N



```
node          mean  sd   MC er 2.5% med 97.5%
prob.low[1]   0.036 0.18 0.002 0.0  0.0 1.0
prob.low[2]   0.024 0.15 0.002 0.0  0.0 0.0
prob.low[3]   0.017 0.12 0.001 0.0  0.0 0.0
prob.low[4]   0.0   0.0  0.0   0.0  0.0 0.0
prob.low[5]   0.0   0.0  0.0   0.0  0.0 0.0
prob.low[6]   0.032 0.17 0.002 0.0  0.0 1.0
prob.low[7]   0.0   0.0  0.0   0.0  0.0 0.0
.....
prob.low[13] 0.34   0.47 0.007 0.0  0.0 1.0
prob.low[14] 0.0    0.0  0.0   0.0  0.0 0.0
.....
prob.low[18] 0.04   0.18 0.002 0.0  0.0 1.0
prob.low[19] 0.25   0.43 0.005 0.0  0.0 1.0
prob.low[20] 0.0    0.0  0.0   0.0  0.0 0.0
prob.low[21] 0.0    0.0  0.0   0.0  0.0 0.0
prob.low[22] 0.25   0.43 0.006 0.0  0.0 1.0
```

Ranking methods may be useful when

- comparing alternative treatments/interventions
- comparing subsets
- comparing response-rates, cost-effectiveness or any summary measure

Lecture 4.
Bayesian regression models

Outline

- Bayesian formulation of linear regression model
  - Implementation in WINBUGS
- Choosing prior distributions for model parameters
- Generalised linear regression models, non-linear regression models, models for categorical data
  - Implementation in WINBUGS
- Making predictions
- Bayesian model comparison

Bayesian regression models

Standard (and non standard) regression models can be easily formulated within a Bayesian framework.

- Specify probability distribution (likelihood) for the data
- Specify form of relationship between response and explanatory variables
- Specify prior distributions for regression coefficients and any other unknown (nuisance) parameters

Bayesian regression models

Some advantages of a Bayesian formulation in regression modelling include:

- Easy to include parameter restrictions and other relevant prior knowledge
- Easily extended to non-linear regression
- Easily ‘robustified’
- Easy to make inference about functions of regression parameters and/or predictions
- Easily extended to handle missing data and covariate measurement error

Linear regression

Consider a simple linear regression with univariate Normal outcome y_i and a vector of covariates $x_{1i}, ..., x_{pi}$, $i = 1, ..., n$

$$y_i = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ki} + \epsilon_i; \qquad \epsilon_i \sim \text{Normal}(0, \sigma^2)$$

An equivalent Bayesian formulation would typically specify

$$\begin{aligned} y_i &\sim \text{Normal}(\mu_i, \sigma^2) \\ \mu_i &= \beta_0 + \sum_{k=1}^{p} \beta_k x_{ki} \\ (\beta_0, \beta_1, ..., \beta_p, \sigma^2) &\sim \text{Prior distributions} \end{aligned}$$

A typical choice of 'vague' prior distribution (see later for more details) that will give numerical results similar to OLS or MLE is:

$$\begin{aligned} \beta_k &\sim \text{Normal}(0, 100000) \quad k = 0, ..., p \\ 1/\sigma^2 &\sim \text{Gamma}(0.001, 0.001) \end{aligned}$$

Example: Small Area Estimation of Average Income

- 284 municipalities in Sweden
- Response = INCOME: average annual household income in area (based on survey data)

$$INCOME_i = \sum_{j=1}^{n_i} \frac{w_{ij} y_{ij}}{\sum w_{ij}};\ w_{ij} \text{ sampling weights};$$

- Predictors = AVAGE: Average age of heads of household; RURAL: Rurality level of the area (1 urban, 2 mixed, 3 rural)
- Model specification:

$$\begin{aligned}
\text{INCOME}_i &\sim \text{Normal}(\mu_i, \sigma^2) \quad i = 1, ..., 284 \\
\mu_i &= \alpha + \beta \times \text{AVAGE}_i + \text{<effect of RURAL>} \\
1/\sigma^2 &\sim \text{Gamma}(0.001, 0.001) \\
\alpha &\sim \text{Normal}(0, 100000) \\
\beta &\sim \text{Normal}(0, 100000) \\
\text{Prior} & \text{ on } \text{ coefficients for RURAL effect}
\end{aligned}$$

Specifying categorical covariates in BUGS language

RURAL$_i$ is a 3-level categorical explanatory variable
Two alternative ways of specifying model in BUGS language

1. Create usual 'design matrix' in data file:

```
  INCOME[]   AVAGE[]   RURAL2[] RURAL3[]
  900.00      43.33       0       0  # Rurality 1
  879.23      50.12       0       0
  890.42      39.37       0       0
 .......
 1011.69      33.09       0       0
 1019.71      40.73       1       0  # Rurality 2
 1032.60      41.32       1       0
 1006.82      55.70       1       0
 .......
 1188.50      29.25       1       0
 1166.29      34.17       0       1  # Rurality 3
 1121.47      52.88       0       1
 .......
 1103.11      47.27       0       1
END
```

BUGS model code

```
for (i in 1:N) {
  INCOME[i] ~ dnorm(mu[i], tau)
  mu[i] <- alpha + beta*(AVAGE[i]-mean(AVAGE[])) +
           delta2*RURAL2[i] + delta3*RURAL3[i]
}
alpha ~ dnorm(0, 0.00001)
beta ~ dnorm(0, 0.00001)
delta2 ~ dnorm(0, 0.00001)
delta3 ~ dnorm(0, 0.00001)
tau ~ dgamma(0.001, 0.001); sigma2 <- 1/tau
```

Note: BUGS parameterises normal in terms of mean and **precision** (1/variance)!!

Initial values file would be something like

```
list(alpha = 1, beta = -2, delta2 = -2,
     delta3 = 4, tau = 2)
```

Specifying categorical covariates in BUGS (cont.)

2. Alternatively, input explanatory variable as single vector coded by its level:

```
  INCOME[]   AVAGE[]   RURAL[]
  900.00      43.33      1
  879.23      50.12      1
  890.42      39.37      1
.......
 1011.69      33.09      1
 1019.71      40.73      2
 1032.60      41.32      2
 1006.82      55.70      2
.......
 1188.50      29.25      2
 1166.29      34.17      3
 1121.47      52.88      3
......
 1103.11      47.27      3
END
```

BUGS model code

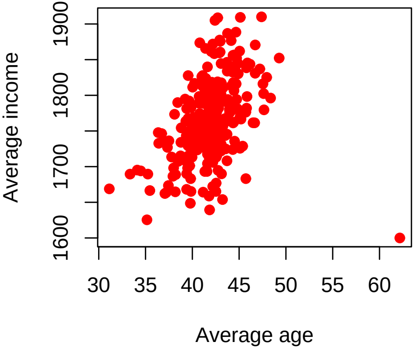
Then use ’double indexing’ feature of BUGS language

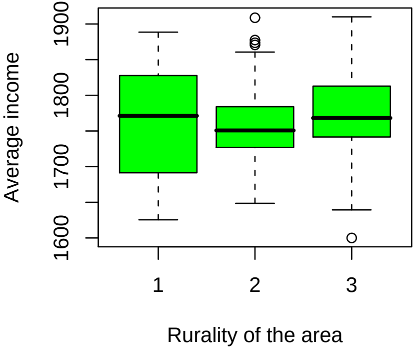
```
for (i in 1:N) {
  INCOME[i] ~ dnorm(mu[i], tau)
  mu[i] <- alpha + beta*(AVAGE[i]-mean(AVAGE[]))
            + delta[RURAL[i]]
}
alpha ~ dnorm(0, 0.00001)
beta ~ dnorm(0, 0.00001)
delta[1] <- 0  # coefficient for reference category
delta[2] ~ dnorm(0, 0.00001)
delta[3] ~ dnorm(0, 0.00001)
tau ~ dgamma(0.001, 0.001);  sigma2 <- 1/tau
```

In initial values file, need to specify initial values for `delta[2]` and `delta[3]` but not `delta[1]`. Use following syntax:

```
list(alpha = 1, beta = -2, delta = c(NA, -2, 4),
     tau = 2)
```

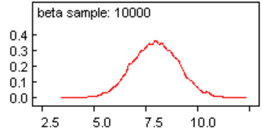
Raw data


Average income
1600
1700
1800
1900
30
35
40
45
50
55
60
Average age


Average income
1600
1700
1800
1900
1
2
3
Rurality of the area

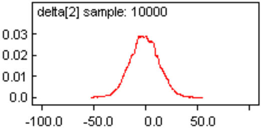
Posterior distributions of regression coefficients

Change in average income per year increase in average age



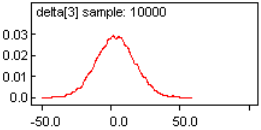
Posterior mean 7.207
95% interval (4.99, 9.47)

Change in average income in Mixed vs Rural



Posterior mean -0.534
95% interval (-27.4, 25.06)

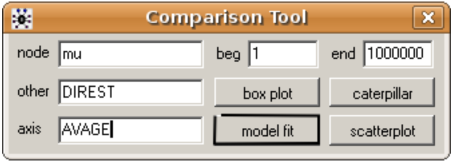
Change in average income in Urban vs Rural



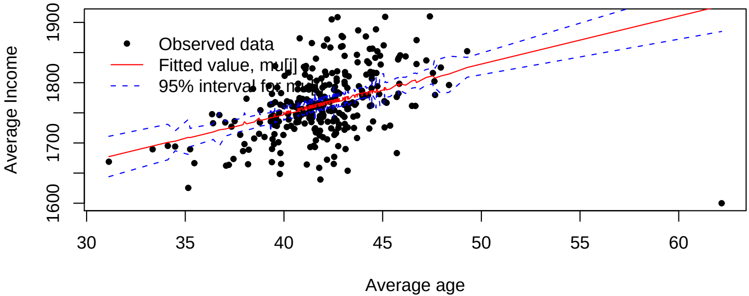
Posterior mean 5.319
95% interval (-22.54, 32.34)

95% intervals for RURAL effect both include zero → drop RURAL

Fitted regression line versus covariate (AVAGE)



node: mu
beg: 1
end: 1000000
other: DIREST
axis: AVAGE



- Influential point corresponds to area 257 (area with the highest AVAGE)

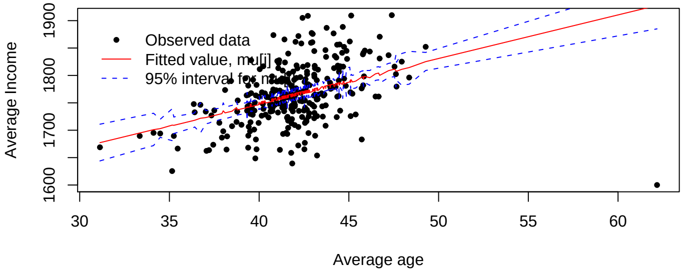
  → Robustify model assuming t-distributed errors

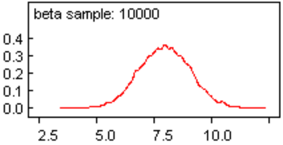
Model with t-errors

```
for (i in 1:N) {
  INCOME[i] ~ dt(mu[i], tau, 4) # robust likelihood
                                # (t on 4 df)
  mu[i] <- alpha + beta*(AVAGE[i]-mean(AVAGE[]))
}
alpha ~ dnorm(0, 0.00001)
beta ~ dnorm(0, 0.00001)
tau ~ dgamma(0.001, 0.001)
sigma2 <- 1/tau

dummy <- RURAL[1]  # ensures all variables in data
                   # file appear in model code
```

Fitted model: Normal errors

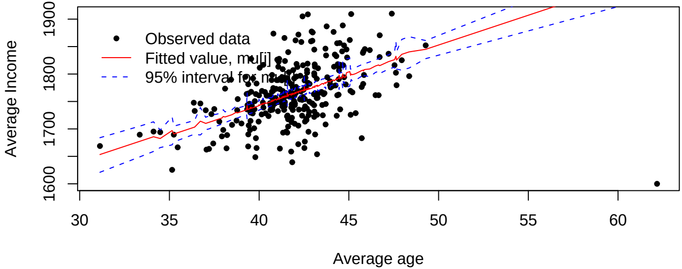


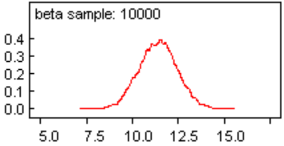


Posterior mean 7.957

95% interval (5.79, 10.07)

Fitted model: t errors





Posterior mean 10.85
95% interval (8.75, 12.99)

Specifying prior distributions

Why did we choose a Normal$(0, 100000)$ prior for each regression coefficient and a Gamma$(0.001, 0.001)$ prior for the inverse of the error variance?

Choice of prior is, in principle, subjective

- it might be elicited from experts (see Spiegelhalter et al (2004), sections 5.2, 5.3)
- it might be more convincing to be based on historical data, *e.g.* a previous study
    - assumed relevance is still a subjective judgement (see Spiegelhalter et al (2004), section 5.4)
- there has been a long and complex search for various 'non-informative', 'reference' or 'objective' priors (Kass and Wasserman, 1996)

‘Non-informative’ priors

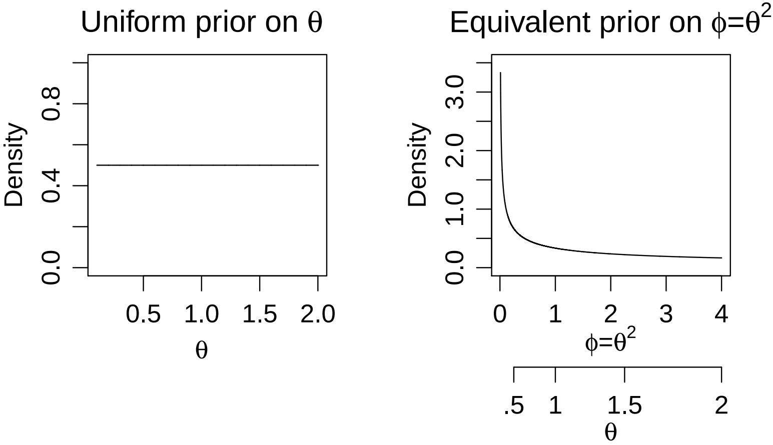
- Better to refer to as ‘vague’, ‘diffuse’ or ‘minimally informative’ priors
- Prior is vague with respect to the likelihood
  - prior mass is diffusely spread over range of parameter values that are plausible, i.e. supported by the data (likelihood)

Uniform priors

Set $p(\theta) \;\propto\; 1$

- This is improper ($\int p(\theta)d\theta \neq 1$)
- The posterior will still usually be proper
- Inference is based on the likelihood $p(x \mid \theta)$
- It is not really objective, since a flat prior $p(\theta) \;\propto\; 1$ on θ does not correspond to a flat prior on $\phi = g(\theta)$, but to $p(\phi) \;\propto\; \left|\frac{d\theta}{d\phi}\right|$ where $\left|\frac{d\theta}{d\phi}\right|$ is the Jacobian
  - Note: Jacobian ensures area under curve (probability) in a specified interval (θ_1, θ_2) is preserved under the transformation $\rightarrow$ same area in interval $(\phi_1 = g(\theta_1), \phi_2 = g(\theta_2))$
- Example: Suppose $p(\theta) \propto 1$ and $\phi = g(\theta) = \theta^2$
  Then $\theta = \sqrt{\phi}$ and $\left|\frac{d\theta}{d\phi}\right| = \frac{1}{2\sqrt{\phi}}$
  So a uniform prior on θ is equivalent to a prior on ϕ such that $p(\phi) \propto \frac{1}{\sqrt{\phi}}$

Priors on transformations



Proper approximations to Uniform($-\infty, \infty$) prior

- $p(\theta) = \text{Uniform}(a, b)$ where a and b specify an appropriately wide range, e.g. Uniform($-1000, 1000$)
  - Remember that if typical values of θ are expected to be around 1–10 (say) then a Uniform($-1000, 1000$) distributions represents a vague prior
  - But if typical values of θ are expected to be around 500–1000 (say) then you would need correspondingly wider bounds, e.g. Uniform($-100000, 1000000$)
- $p(\theta) = \text{Normal}(0, V)$ where V is an appropriately large value for the variance, e.g. Normal($0, 100000$)
  - See comment above re: expected magnitude of θ and implications for choice of V
  - Recall that WinBUGS parameterises Normal in terms of mean and precision, so a normal prior with variance $V = 100000$ will be

```
theta ~ dnorm(0, 0.00001)
```

Jeffreys' invariance priors

- Jeffreys rule for specifying non-informative priors is motivated by the desire that inference should not depend on how the model is parameterised
  - For example, when modelling binomial data, some researchers may model the proportion p whereas others may model the odds (or log odds) $p/(1-p)$
- **Jeffreys rule**: The prior is obtained as the square root of the determinant of the information matrix for the model
  - In mathematical terms, Jeffreys prior for a parameter θ is $p(\theta) \propto I(\theta)^{1/2}$ where $I(\theta)$ is Fisher information for θ

$$I(\theta) = -\mathbb{E}_{X|\theta}\left[\frac{\partial^2 \log p(X|\theta)}{\partial\theta^2}\right] = \mathbb{E}_{X|\theta}\left[\left(\frac{\partial \log p(X|\theta)}{\partial\theta}\right)^2\right]$$

Jeffreys' priors (continued)

- Fisher Information measures curvature of log likelihood
- High curvature occurs wherever small changes in parameter values are associated with large changes in the likelihood
  - Jeffreys' prior gives more weight to these parameter values
  - data provide strong information about parameter values in this region
  - ensures data dominate prior everywhere
- Jeffreys' prior is invariant to reparameterisation because

$$I(\phi)^{1/2} = I(\theta)^{1/2} \left| \frac{d\theta}{d\phi} \right|$$

  That is, using Jeffreys' rule to specify a prior for θ will give a prior that is equivalent to the transformed prior obtained using Jeffreys' rule to specify a prior for $\phi = g(\theta)$

In general, Jeffreys' rule yields

- The flat (uniform) prior for 'location' parameters
- The inverse prior (see later) for 'scale' parameters

Some recommendations for specifying priors

- Distinguish
  - *primary* parameters of interest in which one may want minimal influence of priors
  - *secondary* structure used for smoothing *etc.* in which informative priors may be more acceptable
- Prior best placed on interpretable parameters
- Great caution needed in complex models that an apparently innocuous uniform prior is not introducing substantial information
- *'There is no such thing as a 'noninformative' prior. Even improper priors give information: all possible values are equally likely'* (Fisher, 1996)

Priors for location parameters

‘Location’ parameters are quantities such as means, regression coefficients,...

- Uniform prior on a wide range, or a Normal prior with a large variance can be used, e.g.

$$\begin{aligned} \theta &\sim \text{Unif}(-100, 100) \\ \theta &\sim \text{Normal}(0, 100000) \end{aligned}$$

```
theta ~ dunif(-100, 100)
theta ~ dnorm(0, 0.00001)
```

  Prior will be locally uniform over the region supported by the likelihood

  - ! remember that `WinBUGS` parameterises the Normal in terms of mean and *precision* so a vague Normal prior will have a *small* precision
  - ! ‘wide’ range and ‘small’ precision depend on the scale of measurement of θ

Priors for scale parameters

- Sample variance σ^2: standard 'reference' (Jeffreys') prior is the 'inverse' prior

$$p(\sigma^2) \;\propto\; \frac{1}{\sigma^2} \propto \text{Gamma}(0,0)$$

- This is equivalent to a flat (uniform) prior on the log scale:

$$p(\log(\sigma^2)) \;\propto\; \text{Uniform}(-\infty, \infty)$$

- This prior makes intuitive sense: if totally ignorant about the scale (order of magnitude) of a parameter, then it is equally likely to lie in the interval 1–10 as it is to lie in the interval 10–100, etc.

Priors for scale parameters (continued)

- Jeffreys' prior on the inverse variance (precision, $\tau = \sigma^{-2}$) is

  $$p(\tau) \propto \frac{1}{\tau} \propto \text{Gamma}(0, 0)$$

  which may be approximated by a 'just proper' prior

  $$\tau \sim \text{Gamma}(\epsilon, \epsilon) \qquad (\epsilon\text{small})$$

- This is also the conjugate prior and so is widely used as a 'vague' proper prior for the precision of a Normal likelihood
- In BUGS language: `tau ~ dgamma(0.001, 0.001)`

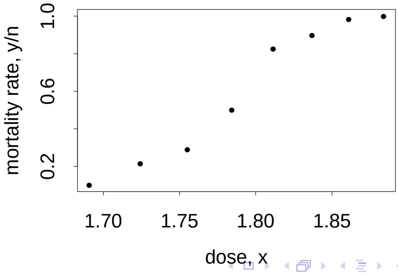
Sensitivity analysis plays a crucial role in assessing the impact of particular prior distributions, whether elicited, derived from evidence, or reference, on the conclusions of an analysis.

Generalised Linear Regression Models

- Specification of Bayesian GLMs follows straightforwardly from previous discussion of linear models
- No closed form solution available, but straightforward to obtain samples from posterior using MCMC

Example: Beetles

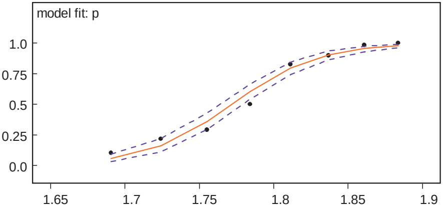
Dobson (1983) analyses binary dose-response data from a bioassay experiment recording numbers of beetles killed after 5 hour exposure to carbon disulphide at N=8 different concentrations



- We start by fitting a logistic regression model

$$
\begin{aligned}
y_i &\sim \text{Binomial}(p_i, n_i) \\
\text{logit}\, p_i &= \alpha + \beta(x_i - \overline{x}) \\
\alpha &\sim \text{Normal}(0, 10000) \\
\beta &\sim \text{Normal}(0, 10000)
\end{aligned}
$$

Beetles: logistic regression model fit (red = posterior mean of p_i; blue = 95% interval; black dots = observed rate y_i/n_i)



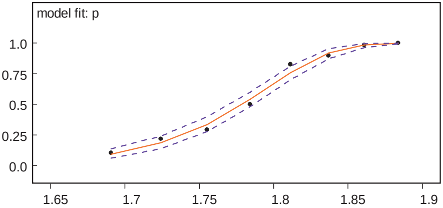
dose level i	obs. rate y_i/n_i	posterior mean of p_i	95% interval
1	0.10	0.06	(0.03, 0.09)
2	0.22	0.16	(0.11, 0.22)
3	0.29	0.36	(0.29, 0.43)
4	0.50	0.61	(0.54, 0.67)
5	0.83	0.80	(0.74, 0.85)
6	0.90	0.90	(0.86, 0.94)
7	0.98	0.96	(0.93, 0.97)
8	1.00	0.98	(0.96, 0.99)

- Some evidence of lack of fit for highest doses, so try alternative complementary log-log (cloglog) link function
  - cloglog function is the inverse of the log Weibull CDF
  - similar shape to logit function, except higher categories are more probable under cloglog than logit

$$\begin{aligned} y_i &\sim \text{Binomial}(p_i, n_i) \\ \text{cloglog}\, p_i &= \alpha + \beta(x_i - \overline{x}) \end{aligned}$$

$$\alpha \sim \text{Normal}(0, 10000); \qquad \beta \sim \text{Normal}(0, 10000)$$

Beetles: cloglog regression model fit (red = posterior mean of p_i; blue = 95% interval; black dots = observed rate y_i/n_i)



dose level i	obs. rate y_i/n_i	posterior mean of p_i	95% interval
1	0.10	0.09	(0.06, 0.14)
2	0.22	0.19	(0.14, 0.24)
3	0.29	0.34	(0.28, 0.40)
4	0.50	0.54	(0.48, 0.60)
5	0.83	0.76	(0.70, 0.81)
6	0.90	0.92	(0.87, 0.95)
7	0.98	0.98	(0.96, 0.99)
8	1.00	1.00	(0.99, 1.00)

Could also try probit (inverse Normal CDF) link function

- Can write probit model in two different ways

$$\text{probit } p_i = \alpha + \beta(x_i - \overline{x})$$

  or

$$p_i = \Phi(\alpha + \beta(x_i - \overline{x}))$$

- In WINBUGS , either

```
probit(p[i]) <- alpha + beta*(x[i]-mean(x[]))
```

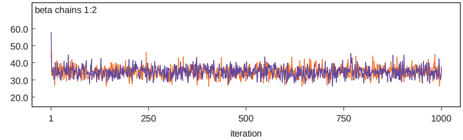
  or

```
p[i] <- phi(alpha + beta*(x[i]-mean(x[])))
```

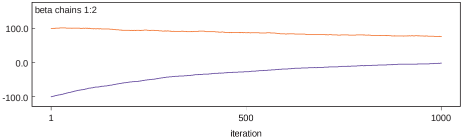
- The second way is *slower*, but can be *more robust* to numerical problems.

Centering covariates to improve convergence

History plot for slope, β: Centred covariate

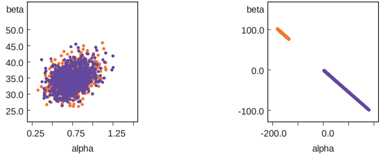


History plot for slope, β: Uncentred covariate



Bivariate scatter plot showing correlation between sampled values of α and β

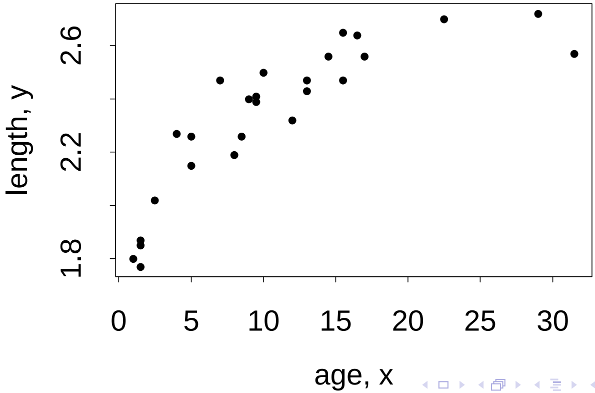
Centered covariate **Uncentred covariate**



Non linear regression models

Example: Dugongs

Carlin and Gelfand (1991) consider data on length (y_i) and age (x_i) measurements for 27 dugongs (sea cows) captured off the coast of Queensland



Dugongs: non linear regression

- A frequently used nonlinear growth curve with no inflection point and an asymptote as x_i tends to infinity is

$$\begin{aligned} y_i &\sim \text{Normal}(\mu_i, \sigma^2) \\ \mu_i &= \alpha - \beta\gamma^{x_i} \end{aligned}$$

  where $\alpha, \beta > 0$ and $\gamma \in (0, 1)$

- Vague prior distributions with suitable constraints may be specified

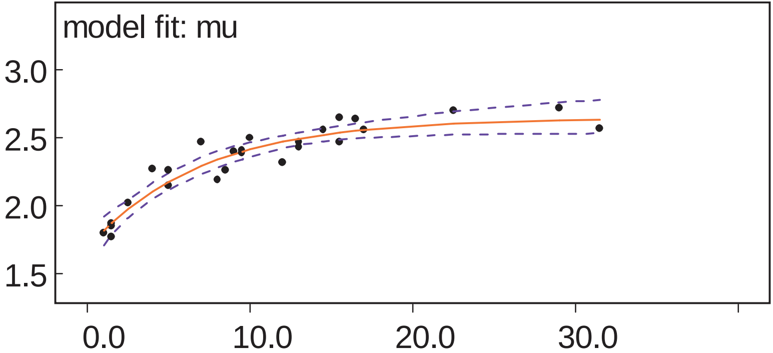
$$\begin{aligned} \alpha &\sim \text{Uniform}(0, 100) \quad \text{or} \quad \alpha \sim \text{Normal}(0, 10000)I(0,) \\ \beta &\sim \text{Uniform}(0, 100) \quad \text{or} \quad \beta \sim \text{Uniform}(0, 10000)I(0,) \\ \gamma &\sim \text{Uniform}(0, 1) \end{aligned}$$

- For the sampling variance, could specify uniform prior on log variance or log sd scale

$$\log\sigma \sim \text{Uniform}(-10, 10)$$

  or gamma prior on precision scale

Dugongs: model fit



(red = posterior mean of μ_i; blue = 95% interval)

Modelling categorical variables in WINBUGS

1. A single binary variable, $y = 0$ or 1

   ```
   y ~ dbern(p)   # Bernoulli distribution
   ```

   where p represents a probability

2. A single categorical variable, e.g. $y = 1{,}2$ or 3

   ```
   y ~ dcat(p[])
   ```

   where `p[]` is a 3-dimensional vector of probabilities

3. Counts of n independent categorical variables

   ```
   y[1:K] ~ dmulti(p[], n)
   ```

   where `y[1:K]` is a vector of length K giving counts of the number of times category $1, 2, ..., K$ was observed in n independent trials and `p[]` is a K-dimensional vector of probabilities

Example of using the `dcat` distribution in WinBUGS : Modelling unknown denominators

- Suppose we are told that a fair coin has come up heads 10 times
  - how many times (*n*) has it been tossed?
- We want to specify a uniform prior distribution for *n*

Unknown denominators (continued)

1. Could give a continuous prior distribution for *n* and use ‘round’ function

```
model {
   r <- 10
   q <- 0.5
   r ~ dbin(q, n)
   n.cont ~ dunif(10, 100)
   n <- round(n.cont)
}
```

BUGS output:

```
node   mean  sd   MC error 2.5% median 97.5% start sample
n      21.0  4.79 0.0790   13.0 21.0   32.0  1001  5000
n.cont 21.08 4.80 0.07932  13.3 20.6   32.0  1001  5000
```

We can be 95% sure that the coin has been tossed between 13 and 32 times

Unknown denominators (continued)

2. Or a discrete uniform prior on 10 to 100

```
model {
  r <- 10
  q <- 0.5
  r ~ dbin(q, n)
  # discrete prior on 10 to 100
  for(j in 1:9)   {  p[j]<-0   }
  for(j in 10:100){  p[j]<-1/91}
  n ~ dcat(p[])
}
```

BUGS output:

```
node   mean   sd   MC error 2.5% median 97.5% start sample
n      21.07 4.76 0.0392   13.0 21.0   32.0  1001  10000
```

We obtain a similar answer to before, i.e. we can be 95% sure that the coin has been tossed between 13 and 32 times

Making predictions

- Important to be able to predict unobserved quantities for
  - ‘filling-in’ missing or censored data
  - model checking - are predictions ‘similar’ to observed data?
  - making predictions!
- Easy in MCMC/WinBUGS; just specify a stochastic node without a data-value - it will be automatically predicted
- Provides automatic imputation of missing data
- Easiest case is where there is no data at all: just ‘forward sampling’ from prior, *Monte Carlo* methods

Example: Dugongs — prediction

Suppose we want to project beyond current observations, eg at ages 35 and 40

Could explicitly set up predictions

```
for (i in 1:N){
    y[i]   ~   dnorm( mu[i], inv.sigma2 )
    mu[i] <-   alpha - beta * pow(gamma, x[i])
    }
mu35  <-   alpha - beta * pow(gamma, 35)
mu40  <-   alpha - beta * pow(gamma, 40)
y35    ~   dnorm( mu35, inv.sigma2 )
y40    ~   dnorm( mu40, inv.sigma2 )
```

Interval around μ_{40} will reflect uncertainty concerning fitted parameters

Interval around y_{40} will additionally reflect sampling error σ and uncertainty about σ

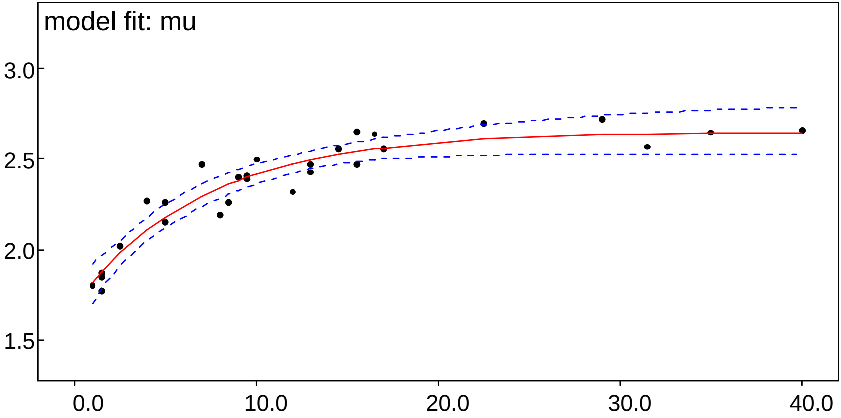
Dugongs: prediction as missing data

Easier to set up as missing data - WinBUGS automatically predicts it

```
list(x = c( 1.0, 1.5, 1.5, 1.5, 2.5, 4.0, 5.0, 5.0, 7.0,
            8.0, 8.5, 9.0, 9.5, 9.5, 10.0, 12.0, 12.0,
            13.0, 13.0, 14.5, 15.5, 15.5, 16.5, 17.0,
            22.5, 29.0, 31.5, 35, 40),
     Y = c(1.80, 1.85, 1.87, 1.77, 2.02, 2.27, 2.15, 2.26,
           2.47, 2.19, 2.26, 2.40, 2.39, 2.41, 2.50, 2.32,
           2.32, 2.43, 2.47, 2.56, 2.65, 2.47, 2.64, 2.56,
           2.70, NA, NA), N = 29)
```

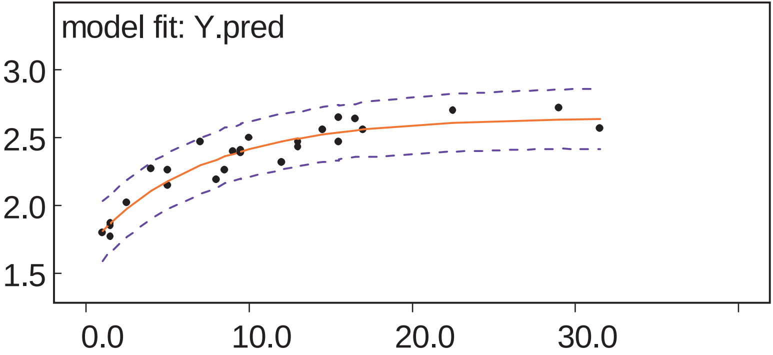
```
node   mean sd      MC error 2.5% median 97.5% start sample
mu[28] 2.65 0.071 0.00423    2.53 2.642  2.815 1001  10000
 Y[28] 2.65 0.122 0.00453    2.41 2.648  2.902 1001  10000
mu[29] 2.65 0.078 0.00477    2.53 2.644  2.837 1001  10000
 Y[29] 2.65 0.127 0.00502    2.41 2.649  2.921 1001  10000
```

Dugongs: projections



Dugongs: prediction as model checking

```
y.pred[i] ~  dnorm( mu[i], inv.sigma2 )
```



Bayesian model comparison using DIC

- Natural way to compare models is to use criterion based on trade-off between the fit of the data to the model and the corresponding complexity of the model
- Spiegelhalter et al (2002) proposed a Bayesian model comparison criterion based on this principle:

Deviance Information Criterion,
DIC = ‘goodness of fit’ + ‘complexity’

DIC (continued)

- They measure fit via the deviance

$$D(\theta) = -2 \log L(\text{data}|\theta)$$

- Complexity measured by estimate of the 'effective number of parameters':

$$\begin{aligned} p_D &= E_{\theta|y}[D] - D(E_{\theta|y}[\theta]) \\ &= \overline{D} - D(\overline{\theta}); \end{aligned}$$

  i.e. posterior mean deviance minus deviance evaluated at the posterior mean of the parameters

- The DIC is then defined analagously to AIC as

$$\begin{aligned} \text{DIC} &= D(\overline{\theta}) + 2p_D \\ &= \overline{D} + p_D \end{aligned}$$

  Models with smaller DIC are better supported by the data

- DIC can be monitored in `WinBUGS` from `Inference/DIC` menu

DIC (continued)

- These quantities are easy to compute in an MCMC run
- Aiming for Akaike-like, cross-validatory, behaviour based on ability to make short-term predictions of a repeat set of similar data.
- p_D is not invariant to reparameterisation.
- p_D can be be negative! (not desirable)
- Care needed when posterior mean is not a good plug-in estimate (e.g. highly skewed posteriors, discrete parameters)
- DIC can be negative (not a problem!)
- Only differences in DIC are of interest — rule of thumb: differences in DIC of 4–7 or more provide evidence of substantial difference in model fit

Beetles: model comparison using DIC

Recall: we fitted several alternative models for the link function: a logistic regression model, a log-log link function and a probit link.

$$
\begin{array}{rcl}
y_i & \sim & \text{Binomial}(p_i, n_i) \\
\text{`link'}\ p_i & = & \alpha + \beta(x_i - \overline{x}) \\
\text{where `link'} & \text{was} & \text{one of `logit', `cloglog' or `probit'} \\
\alpha & \sim & \text{Normal}(0, 10000) \\
\beta & \sim & \text{Normal}(0, 10000)
\end{array}
$$

	$\overline{D}$	$\hat{D}$	p_D	DIC
logit link	39.43	37.45	1.99	41.42
probit link	38.34	36.32	2.02	40.35
cloglog link	31.65	29.65	2.00	33.65

DIC shows clear support for cloglog link model over logit and probit.

Summary and Next Steps

Summary

Key concepts you should be familiar with after this course

- Use of probability distributions to represent uncertainty about unknown quantities
- Prior to posterior updating of probability distributions in light of data (Bayesian inference)
- Idea of summarising samples generated from probability distributions to make inference about uncertain quantities (Monte Carlo and MCMC methods)
- How to choose suitable prior distributions for different types of variables

Key skills you should have learnt during this course

- Using WINBUGS for forward sampling and for fitting simple Bayesian models
- How to check convergence of MCMC runs
- How to summarise and interpret the output from MCMC runs

Next Steps

- Work through some of the examples in the WinBUGS ‘Help’ menu
- Check the links on the WinBUGS web resources page `www.mrc-bsu.cam.ac.uk/bugs/weblinks/webresource.shtml`
    - Includes links to books containing WinBUGS examples/code, Bayesian/WinBUGS teaching materials, and discipline-specific sites with examples of WinBUGS analyses/code
- Join the bugs email discussion list: Send a one line message

  `join bugs firstname(s) lastname`

  to

  `jiscmail@jiscmail.ac.uk`
- Attend a more advanced course on Bayesian methods and WinBUGS

References and Further Reading

Berry, DA (1996). *Statistics: A Bayesian Perspective*, Duxbury, London.

Breslow, N (1990). Biostatistics and Bayes. *Statistical Science*, **5**, 269–298.

Brooks, SP (1998). Markov chain Monte Carlo method and its application. *The Statistician*, **47**, 69-100.

Brooks, SP and Gelman, A (1998). Alternative methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, **7**, 434-455.

Congdon, P. (2001) *Bayesian statistical modelling*. Wiley.

Cowles, MK and Carlin, BP (1996) Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, **91**, 883–904.

Dunson, D (2001). Commentary: Practical advantages of Bayesian analysis in epidemiologic data. *American Journal of Epidemiology*, **153**, 1222–1226.

Gelman, A, Carlin, JC, Stern, H and Rubin, DB (2004). *Bayesian Data Analysis*, 2nd edition, Chapman & Hall, New York.

Hoff, P. (2009). *A first course in Bayesian Statistical Methods*. Springer, New York.

References and Further Reading

Kass, RE and Wasserman, L (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, **91**, 1343–70.

Lee, PM (2004). *Bayesian Statistics: An Introduction*, 3rd edition, Arnold, London.

Lilford, RJ and Braunholtz, D (1996). The statistical basis of public policy: a paradigm shift is overdue. *British Medical Journal*, **313**, 603–607.

Senn, S (1997). Statistical basis of public policy — present remembrance of priors past is not the same as a true prior. *British Medical Journal*, **314**, 73.

Spiegelhalter, DJ (1998). Bayesian graphical modelling: a case-study in monitoring health outcomes. *Journal of the Royal Statistical Society, Series C*, **47**, 115–133.

Spiegelhalter, DJ, Gilks, WR and Richardson, S (1996). *Markov chain Monte Carlo in Practice*, Chapman & Hall, London.

Spiegelhalter, DJ, Abrams, K and Myles, JP (2004). *Bayesian Approaches to Clinical Trials and Health Care Evaluation*, Wiley, Chichester.

Spiegelhalter, DJ, Best, NG, Carlin, BP, and van der Linde, A (2002). Bayesian measures of model complexity and fit (with discussion). *J Roy Statist Soc B*, **64**, 583–639.