The University of Newcastle

Kerrie Mengersen

Introduction to
Bayesian Modelling - 2

Armidale 2004
INTRODUCTION TO BAYESIAN COMPUTATION

- Markov chain Monte Carlo
- Introduction to BUGS
- Convergence diagnostics
Markov chain Monte Carlo

• “Decompose” joint posterior distribution into a sequence of conditional distributions – these are often much simpler (eg, simple univariate normals, etc)

• Simulate from each conditional distribution in turn. We use a simulation method that resembles a Markov chain (so that the new simulated value relies only on the previous value), giving a set of simulated values

\[ \theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(i)}, \ldots \]

which converges to the required conditional, The resulting simulations will come from the required joint distribution

• We can use Markov chain theory to make statements about behaviour and convergence of the chain
MCMC Algorithms

- **Gibbs sampling**: sample from full conditionals

- **Metropolis-Hastings**: sample from an “easy” distribution and accept only some of the values

- Lots of variations: reversible jump, slice sampling, particle filters, perfect sampling, adaptive rejection sampling, etc

- Need to ensure conditions, eg *detailed balance*, *reversibility*
Gibbs sampling

Joint posterior $p(\theta_1, \theta_2, ..., \theta_k | y)$

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

At $i$th iteration $(i+1)$

2. Sample $\theta_1^{(i+1)}$ from $p(\theta_1^{(i)} | \theta_2^{(i)}, \theta_3^{(i)}, ..., \theta_k^{(i)}, y)$
   
   Sample $\theta_2^{(i+1)}$ from $p(\theta_2^{(i)} | \theta_1^{(i+1)}, \theta_3^{(i)}, ..., \theta_k^{(i)}, y)$
   
   ...  

   Sample $\theta_k^{(i+1)}$ from $p(\theta_k^{(i)} | \theta_1^{(i+1)}, \theta_2^{(i+1)}, ..., \theta_{k-1}^{(i+1)}, y)$

3. Repeat step 2 many times
Estimation using MCMC

Have simulations:

\[ \begin{array}{cccc}
\theta_1^{(0)} & \theta_2^{(0)} & \ldots & \theta_k^{(0)} \\
\theta_1^{(1)} & \theta_2^{(1)} & \ldots & \theta_k^{(1)} \\
\theta_1^{(2)} & \theta_2^{(2)} & \ldots & \theta_k^{(2)} \\
\vdots & \vdots & \ddots & \vdots \\
\theta_1^{(l)} & \theta_2^{(l)} & \ldots & \theta_k^{(l)} \\
\end{array} \]

Easy to estimate expected values:

\[ E_{\theta | Y} \left( \frac{\theta_1}{\theta_2} \right) \approx \frac{1}{n} \sum_{i=1}^{n} \left( \frac{\theta_1^{(i)}}{\theta_2^{(i)}} \right) \]

Easy to estimate quantiles (credible intervals)
Easy to estimate densities.
Metropolis sampling

- Often we can’t simulate from conditional dist’n.
- Instead, simulate from “easy” (proposal) distribution and accept only some of the values.
  - **Conditional distribution** \( p(\theta |...) \)
  - **Proposal distribution** \( q(\theta) \)
  - Suppose we have \( \theta^{(i-1)} \) and we want \( \theta^{(i)} \)
  - Simulate possible \( \theta^{(i)} \) (\( \theta^* \) say) from \( q(\theta) \) centred on \( \theta^{(i)} \)
  - Accept \( \theta^* \) with probability:
    \[
    \alpha = \min \{ 1, \frac{p(\theta^* |...)}{p(\theta^{(i-1)} |...)} \}
    \]
  - If \( \theta^* \) is accepted, \( \theta^{(i)} = \theta^* \); otherwise \( \theta^{(i)} = \theta^{(i-1)} \)
Hastings sampler

• If the proposal $q(\theta)$ is not symmetric, the acceptance probability becomes:
  
  \[
  \alpha = \min(1, \frac{q(\theta^{(i-1)})}{q(\theta^*)} \frac{p(\theta^* | \ldots)}{p(\theta^{(i-1)} | \ldots)})
  \]
  
  – Accept $\theta^*$ with probability:
Graphical Representation (conditional independence graphs)

- Concentrate on **structural** relationships
- Directed, undirected and chain graphs
  - nodes represent random quantities
  - links represent relationships
  - missing links represent conditional independence

- Use graphs to:
  - break complex models into simple components
  - communicate essential structure
  - provide basis for computation
Example: Binomial model

- Model

\[ y_i \sim \text{Binomial} \left( \theta_i, n_i \right) \]
\[ \theta_i \sim \text{Beta} \left( a, b \right) \]
Explanation of Graph

3 types of node:

- Constants: double edged boxes
  no parents

- Stochastic: circles
  variables (data or parameters)
  given a probability distribution
  have solid arrows pointing to them

- Deterministic: circles
  logical functions of other nodes
  have dashed arrows pointing to them
Example: Logistic model

• Model:

\[ y_i \sim \text{Binomial} \left( p_i, n_i \right) \]
\[ \text{logit}(p_i) = b_i \]
\[ b_i \sim \text{Normal} \left( \mu, \tau \right) , \tau = 1/ \sigma^2 \]

• Priors

\[ \mu \sim \text{Normal} \left( 0, 1E-6 \right) \]
\[ \tau \sim \text{Gamma} \left( 1E-3, 1E-3 \right) \]
DAG for logistic model

\[ b_i \]

\[ p_i \]

\[ y_i \]

Group \( i \)

\[ n_i \]

\[ \tau \]

\[ \mu \]
Example: More rats

• **Explanation**
  
  30 rats, weighed weekly for 5 weeks.
  
  Model as random effects linear growth curve.

<table>
<thead>
<tr>
<th>Weights $Y_{ij}$ of rat $i$ on day $x_j$</th>
<th>$x_j$ = 8</th>
<th>15</th>
<th>22</th>
<th>29</th>
<th>36</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rat 1</td>
<td>151</td>
<td>199</td>
<td>246</td>
<td>283</td>
<td>320</td>
</tr>
<tr>
<td>Rat 2</td>
<td>145</td>
<td>199</td>
<td>249</td>
<td>293</td>
<td>354</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rat 30</td>
<td>153</td>
<td>200</td>
<td>244</td>
<td>286</td>
<td>324</td>
</tr>
</tbody>
</table>
Model for Rats

• Model

\[ y_{ij} \sim \text{Normal} (\alpha_i + \beta_i (x_j - x), \tau_c) \]

• Priors

\[ \alpha_i \sim \text{Normal} (\alpha_c, r_\alpha) \]

\[ \beta_i \sim \text{Normal} (\beta_c, r_\beta) \]

\[ \alpha_c \sim \text{Normal} (0, 1E-4) \]

\[ \beta_c \sim \text{Normal} (0, 1E-4) \]

\[ \tau_c \sim \text{Gamma} (1E-3, 1E-3) \]

\[ \tau_\alpha \sim \text{Gamma} (1E-3, 1E-3) \]

\[ \tau_\beta \sim \text{Gamma} (1E-3, 1E-3) \]
DAG for rats Logistic model

\[ Y_{ij} = \alpha_i + \beta_j + \mu_{ij} + x_j - x \]

\[ \alpha_c \leftarrow \tau_c \]

\[ \tau_{\alpha} \leftarrow \alpha_i \]

\[ \beta_c \leftarrow \tau_{\beta} \]

\[ \alpha_i \leftarrow \beta_j \]

\[ \mu_{ij} \leftarrow \tau_{c} \]

\[ Y_{ij} \leftarrow \mu_{ij} \]

Days \( j \)  
Rats \( i \)
BUGS

Three current trends:
- Complex hierarchical (random-effects) models being analysed using S-plus, SAS etc
- Graphical models used in multivariate analysis
- Markov chain Monte Carlo (MCMC) methods turning Bayesian into mainstream statistics

Brought together in BUGS:
Bayesian Inference Using Gibbs Sampling
The BUGS Program

• Language for specifying complex directed graphical models
• Constructs graph by identifying parents and children
• Simulates via Gibbs and Metropolis-Hastings algorithms
• Currently restricted to particular distributions (discrete, conjugate, log-concave)
Example: probability of (death) after cardiac surgery?

- 12 hospitals
- Sample size (n) , deaths (y):

<table>
<thead>
<tr>
<th>n</th>
<th>47</th>
<th>148</th>
<th>119</th>
<th>810</th>
<th>211</th>
<th>196</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>0</td>
<td>18</td>
<td>8</td>
<td>46</td>
<td>8</td>
<td>13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n</th>
<th>148</th>
<th>215</th>
<th>207</th>
<th>97</th>
<th>256</th>
<th>360</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>9</td>
<td>31</td>
<td>14</td>
<td>8</td>
<td>29</td>
<td>24</td>
</tr>
</tbody>
</table>
PGM for Hospitals

\[ \theta_i \xrightarrow{i=1,...,k} x_i \]

stochastic
constant

plate
BUGS code

model surgical;
const
   N = 12 ;       # number of hospitals
var
   r[N], p[N], n[N], b[N], mu, tau, sigma, pop.mean;
data r, n in "surgical.dat"
inits in "surgical.in"
{
   for (i in 1:N) {
      r[i] ~ dbin( p[i], n[i] );
      logit(p[i]) <- b[i];
      b[i] ~ dnorm( mu, tau);
   }
   # Priors:
   mu ~ dnorm(0.0, 1.0E-6)
   pop.mean <- exp(mu) / (1 + exp(mu));      # population mean
   tau ~ dgamma (1.0E-3, 1.0E-3);    # 1/sigma^2
   sigma <- 1/sqrt(tau)
}

Related BUGS files

- “surgical.dat” (BUGS data file)
  
  \[
  \begin{array}{ll}
  r & n \\
  0 & 47 \\
  \cdots \\
  24 & 360 \\
  \end{array}
  \]

- “surgical.init” (BUGS initial value file)
  
  \[
  \text{list ( } \tau = 1, \mu = 0 \text{)}
  \]
Running BUGS: log file

Bugs> compile ("surgical.bug")
Bugs> update(500)
   500 updates took 00:00:04
Bugs> monitor (p)
Bugs> monitor (pop.mean)
Bugs> monitor (sigma)
Bugs> update (1000)
   1000 updates took 00:00:08
Bugs> stats (p)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
<th>2.5% : 97.5% CI</th>
<th>median</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>5.17E-2</td>
<td>2.08E-2</td>
<td>1.50E-2 9.42E-2</td>
<td>5.01E-2</td>
<td>1000</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bugs> stats (pop.mean)

<table>
<thead>
<tr>
<th></th>
<th>mean</th>
<th>sd</th>
<th>2.5% : 97.5% CI</th>
<th>median</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.30E-2</td>
<td>1.07E-2</td>
<td>5.17E-2 9.49E-2</td>
<td>7.29E-2</td>
<td>1000</td>
</tr>
</tbody>
</table>

Bugs> q( )
Example: BUGS code for rats

model rats;
const
    N = 30 ; # number of rats
    T = 5; # number of time points
var
    tau.c, alpha0, alpha.c, beta.c, x[T], mu[N,T], Y[N,T], alpha[N], beta[N], tau.alpha, tau.beta, x.bar;
data Y in “rats_y.dat”, x in “rats_x.dat”;
inits in “rats.in”
{  
    for (i in 1:N)  {
        for (j in 1:T) {
            mu[i,j] <- alpha[i] + beta[i] * (x[j] - x.bar);
            Y[i,j] ~ dnorm(mu[i,j], tau.c)
        }
        alpha[i] ~ dnorm(alpha.c, tau.alpha);
        beta[i] ~ dnorm(beta.c, tau.beta);
    }
    alpha.c ~ dnorm(0, 1.0E-4);
    beta.c ~ dnorm(0, 1.0E-4);
    tau.c ~ dgamma(1.0E-3, 1.0E-3);
    tau.alpha ~ dgamma(1.0E-3, 1.0E-3);
    tau.beta ~ dgamma(1.0E-3, 1.0E-3);
    sigma <- 1.0 / sqrt(tau.c);
    x.bar <- mean( x[] );
    alpha0 <- alpha.c - beta.c * x.bar;
}
<table>
<thead>
<tr>
<th>node</th>
<th>mean</th>
<th>sd</th>
<th>MC error</th>
<th>2.5% median</th>
<th>97.5% start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>beta[1]</td>
<td>6.063</td>
<td>0.2411</td>
<td>0.004325</td>
<td>5.595</td>
<td>6.065</td>
<td>6.521</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[2]</td>
<td>7.048</td>
<td>0.257</td>
<td>0.005173</td>
<td>6.563</td>
<td>7.049</td>
<td>7.548</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[3]</td>
<td>6.48</td>
<td>0.2471</td>
<td>0.004511</td>
<td>5.994</td>
<td>6.48</td>
<td>6.968</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[4]</td>
<td>5.345</td>
<td>0.2576</td>
<td>0.005856</td>
<td>4.851</td>
<td>5.345</td>
<td>5.864</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[6]</td>
<td>6.178</td>
<td>0.2384</td>
<td>0.003631</td>
<td>5.72</td>
<td>6.174</td>
<td>6.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[7]</td>
<td>5.972</td>
<td>0.2469</td>
<td>0.005217</td>
<td>5.484</td>
<td>5.971</td>
<td>6.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[8]</td>
<td>6.413</td>
<td>0.2452</td>
<td>0.004439</td>
<td>5.919</td>
<td>6.414</td>
<td>6.889</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[9]</td>
<td>7.055</td>
<td>0.2542</td>
<td>0.005396</td>
<td>6.564</td>
<td>7.051</td>
<td>7.553</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta[10]</td>
<td>5.848</td>
<td>0.2464</td>
<td>0.004784</td>
<td>5.353</td>
<td>5.85</td>
<td>6.34</td>
</tr>
</tbody>
</table>
Example: regression

Consider a set of 5 observed \((x, Y)\) pairs \((1, 1), (2, 3), (3, 3), (4, 3), (5, 5)\). We shall fit a simple linear regression of \(Y\) on \(x\), using the notation

\[
Y_i \sim \text{Normal}(\mu_i, \tau)
\]

\[
\mu_i = \alpha + \beta(x_i - x.\text{bar})
\]

where \(x.\text{bar}\) represents the mean of the \(x\)'s. Note that we parameterise the normal distribution in terms of its precision \(\tau\), which is \(1/\text{variance}\).
What about convergence?

• Theoretical

\[
\sup_{x \in C} |P^n(x, C) - P^\infty(C)| \leq M \rho^n_C \\
\int P(x, dy) V(y) \leq (1 - \beta) V(x) + I_C(x)
\]

• Diagnostics
CODA

• Output processor for BUGS
• Menu-driven set of S-Plus functions for:
  • Convergence diagnosis
    - specific methods
    - autocorrelations and cross-correlations
  • Summary statistics
    - empirical mean, sd, quantiles
    - standard error of the mean
  • Graphical
    - sample trace for each variable
    - kernel density
    - plots of some convergence diagnostics
Convergence: Geweke (1992)

• Look at a single long run
• Test for equal mean for “early” part (1st quarter) and “late” part (second half) of the chain.
• Test statistic is $Z \sim N(0,1)$ if the sample is all from the same distribution.
• Careful: this is only a test of “non-convergence” and can be misleading.
Convergence: Gelman & Rubin (1992)

- Many long runs
- Widely different starting points
- Convergence assessed via an “analysis of variance” between and within the chains.

- **Monitor convergence by R:** a conservative estimate of how much extra information about the variable that we could expect to gain by running the chains indefinitely
  - $R$ tends to 1 as $n$ tends to infinity
  - $R$ is subject to sampling variation so monitor $R$ and is upper 97.5% confidence limit

- Works best when posterior is approx. normal
  (may need to transform some variables, eg probs, variances)
Convergence: Raftery & Lewis (1992)

- Look at a single long run
- Diagnostic estimates:
  - \( n_0 \): length of burnin
  - \( N \): no. additional iterations needed to estimate a posterior quantile adequately
- Chain must be run for at least \( N_{\text{min}} \) iterations before computing diagnostic
- Can give radically different estimates depending on starting values and required accuracy of estimation
- Can under-estimate \( n_0 \) for extreme quantiles
- Must re-diagnose convergence for each quantile.
- Based on 2-state Markov chain theory.
Convergence: Heidelberger & Welch (1983)

- Look at a single long run
- Hypothesis test based on Brownian bridge theory and spectral density estimation
- Iterative procedure:
  - test $H_0$: entire sample of values for a given variable form a stationary process
  - if $H_0$ rejected, discard first 10% and repeat test
  - continue discarding until $H_0$ accepted or 50% samples are discarded (need to run chain for longer)
- Also estimates numerical S.E. of mean and tests size of C.I.
- Test has very low power to detect lack of convergence for small sample size.
CODA Menus

• CODA Main Menu:
  – Output Analysis  - Diagnostics
  – List/Change Defaults  - Quit

• CODA Output Analysis Menu
  – Plots  - Statistics
  – List/Change Defaults  - Return to Main

• CODA Diagnostics Menu
  – Geweke, Gelman and Rubin, Raftery and Lewis, Heidelberger and Welch, Autocorrelations, Cross-Correlations
  – List/Change Defaults  - Return to Main
CODA Output: Surgical Eg

- Trace plot, Kernel density plot
- Summary statistics
- Quantiles for each variable
- Autocorrelations
- Cross-correlations
Geweke Z-score

Iterations used = 501:1500
Thinning interval = 1
Sample size per chain = 1000
Fraction in 1\textsuperscript{st} window = 0.1
Fraction in 2\textsuperscript{nd} window = 0.5
Z: 0.372 1.650 -2.550 -1.150
Geweke z-plots
Gelman and Rubin 50% and 97.5% shrink factors

<table>
<thead>
<tr>
<th>Variable</th>
<th>Point est.</th>
<th>97.5% quantile</th>
</tr>
</thead>
<tbody>
<tr>
<td>mu</td>
<td>1.0</td>
<td>1.01</td>
</tr>
<tr>
<td>p[1]</td>
<td>1.02</td>
<td>1.10</td>
</tr>
<tr>
<td>p[2]</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>sigma</td>
<td>1.02</td>
<td>1.10</td>
</tr>
</tbody>
</table>

Trace plots, shrink factor plots
Bayesian Output Analysis Program (BOA)
Version 0.4.2 for Windows S-PLUS
Copywrite (c) 1999 Brian J. Smith <brian-j-smith@uiowa.edu>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For a copy of the GNU General Public License write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA, or visit their web site at http://www.gnu.org/copyleft/gpl.html

NOTE: if the menu unexpectedly terminates, type "boa.menu(recover = TRUE)" to restart and recover your work

BOA MAIN MENU

1: File >>
2: Data >>
3: Analysis >>
4: Plot >>
5: Options >>
6: Window >>
Selection: 1
A new diagnostic: phase randomisation

- Run a single chain
- Take Fourier transform
- Randomise phase
- Backtransform
  (Phase scrambling, Fourier bootstrap)

Modality and scale of 3\textsuperscript{rd} Cumulant tells us about:

- Linearity
- Stationarity

of the original series
Imagine

Imagine you're a Bayesian
It's easy if you try,
You just adopt a prior,
And the data updates $\pi$.
Statistics is so simple
With subjective probabilityyyyy -- ah-ah! ah ah...

Now imagine you're a frequentist,
Worrying about what might have been,
Spending your whole lifetime
Analyzing data you've never seen.
And if you want an interval,
You'll need a pivotal quantityyyyy -- ah-ah! ah ah...

You may say I sound like Nozer --
But I'm not the only one:
Every four years we all get together,
To talk, drink beer, and lie in the sun.