Lecture

Introduction to Markov Chain Monte Carlo methods

Learning Objectives

After this session students should be able to:

- Describe MCMC simulation methods
- Compare and contrast MCMC and MC methods learnt in the previous lectures
- Describe the main methods available to check for convergence of MCMC simulations
- Explain the role of MC error in determining the effective sample size of an MCMC simulation

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
- OpenBUGS demo

Why is computation important?

- Bayesian inference centres around the posterior distribution

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

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

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

$\rightarrow$ 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 general Bayesian inference

How do we sample from non-conjugate and high-dimensional posteriors?

- We want samples from joint posterior distribution $p(\theta|y)$
- Independent sampling from $p(\theta|y)$ may be difficult
- **BUT** dependent sampling from a Markov chain with $p(\theta|y)$ as its stationary (equilibrium) distribution is easier
- A sequence of random variables $\theta^{(0)}, \theta^{(1)}, \theta^{(2)}, \ldots$ 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 $\theta^{(i-1)}, \ldots, \theta^{(0)}$

Sampling from the posterior using Markov chains

Several standard 'recipes' available for designing Markov chains with required stationary distribution $p(\theta|y)$

- 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 $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$

1) Choose starting values $\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_k^{(0)}$
2) Sample $\theta_1^{(1)}$ from $p(\theta_1 | \theta_2^{(0)}, \theta_3^{(0)}, \ldots, \theta_k^{(0)}, y)$

   Sample $\theta_2^{(1)}$ from $p(\theta_2 | \theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_k^{(0)}, y)$

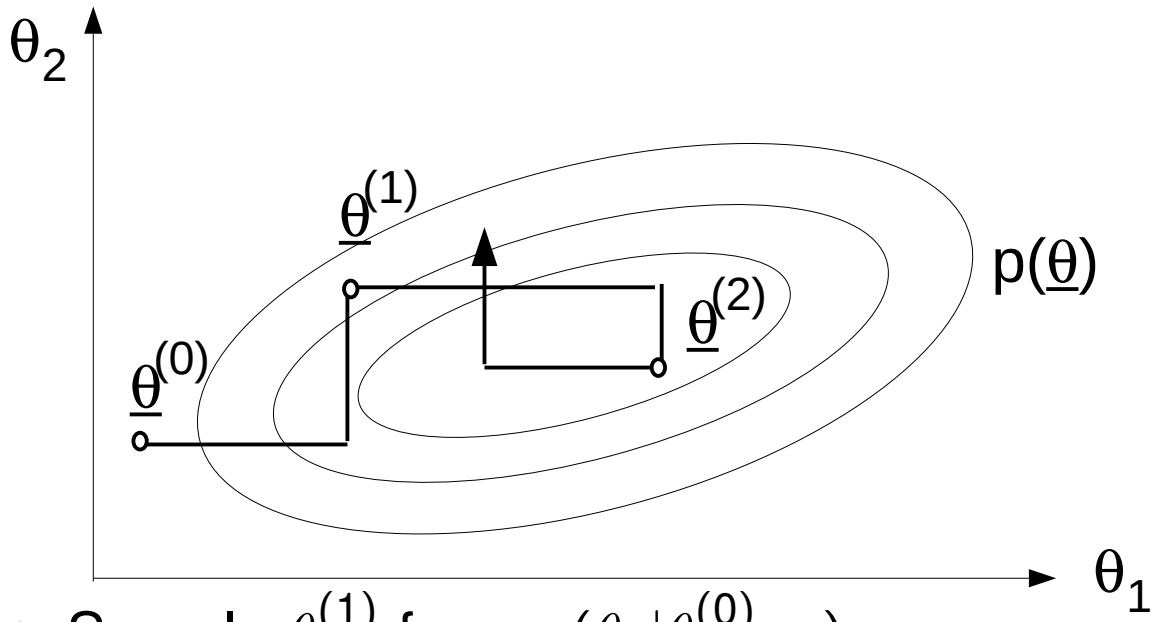
   $\vdots$

   Sample $\theta_k^{(1)}$ from $p(\theta_k | \theta_1^{(1)}, \theta_2^{(1)}, \ldots, \theta_{k-1}^{(1)}, y)$
3) Repeat step 2 many 1000s of times
   - eventually obtain sample from $p(\boldsymbol{\theta}|y)$

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

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

Initial values

- MCMC requires initial (starting) values to be specified for all unknown quantities
- OpenBUGS can automatically generate initial values using *gen inits*
  - these are generated from the prior distribution for each variable
- OK if have informative priors
- If have fairly 'vague' priors, better for user to provide reasonable values in a separate initial values list

Initial values list can be after model description or in a separate file, e.g.

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

Note: initial values are just a starting point for the MCMC simulation, they are not priors

Using MCMC methods

There are two main issues to consider

1. Convergence
   - how quickly does the distribution of $\theta^{(t)}$ approach $p(\theta|y)$?
2. Efficiency
   - how well are functionals of $p(\theta|y)$ 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 know the 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 (≥ 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 (similar to example from Practical 3 last week)

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 (uncentred analysis)

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

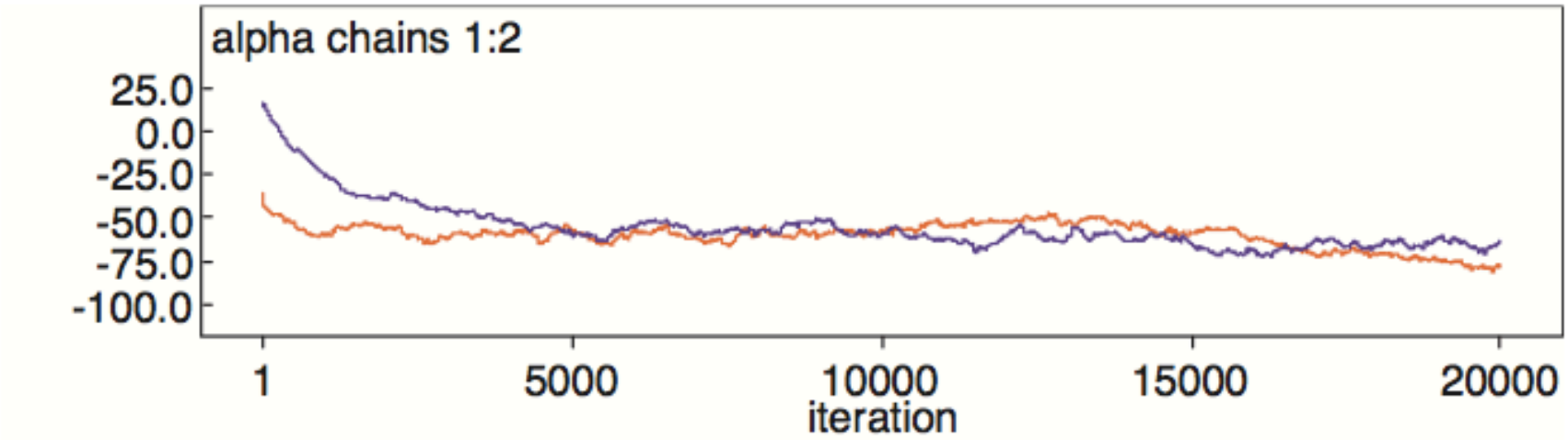
Checking convergence with multiple runs

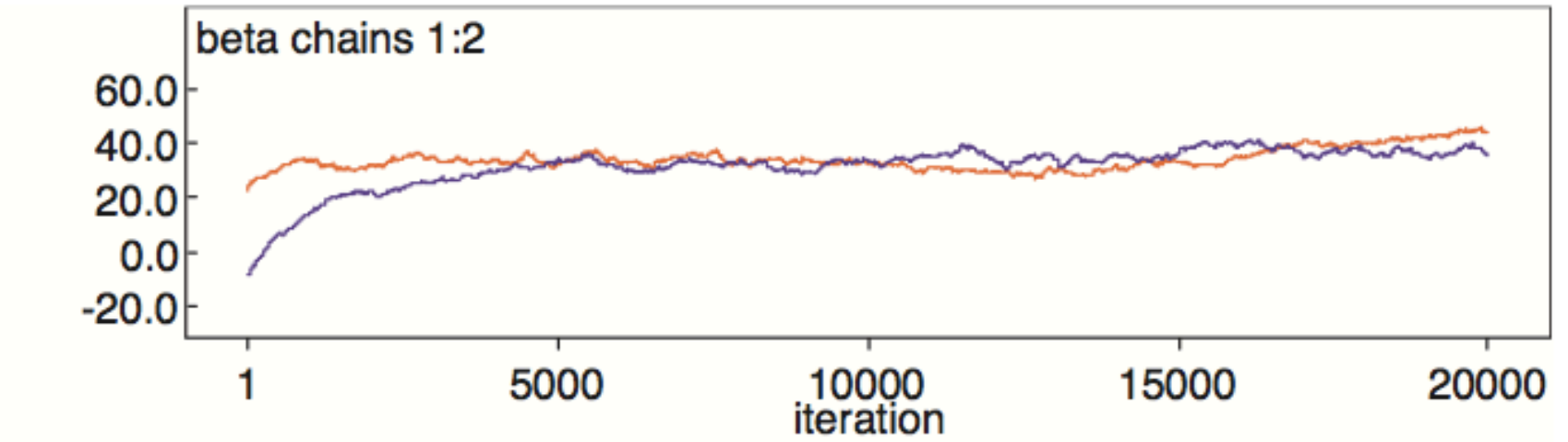
- Set up multiple initial value lists, e.g.

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

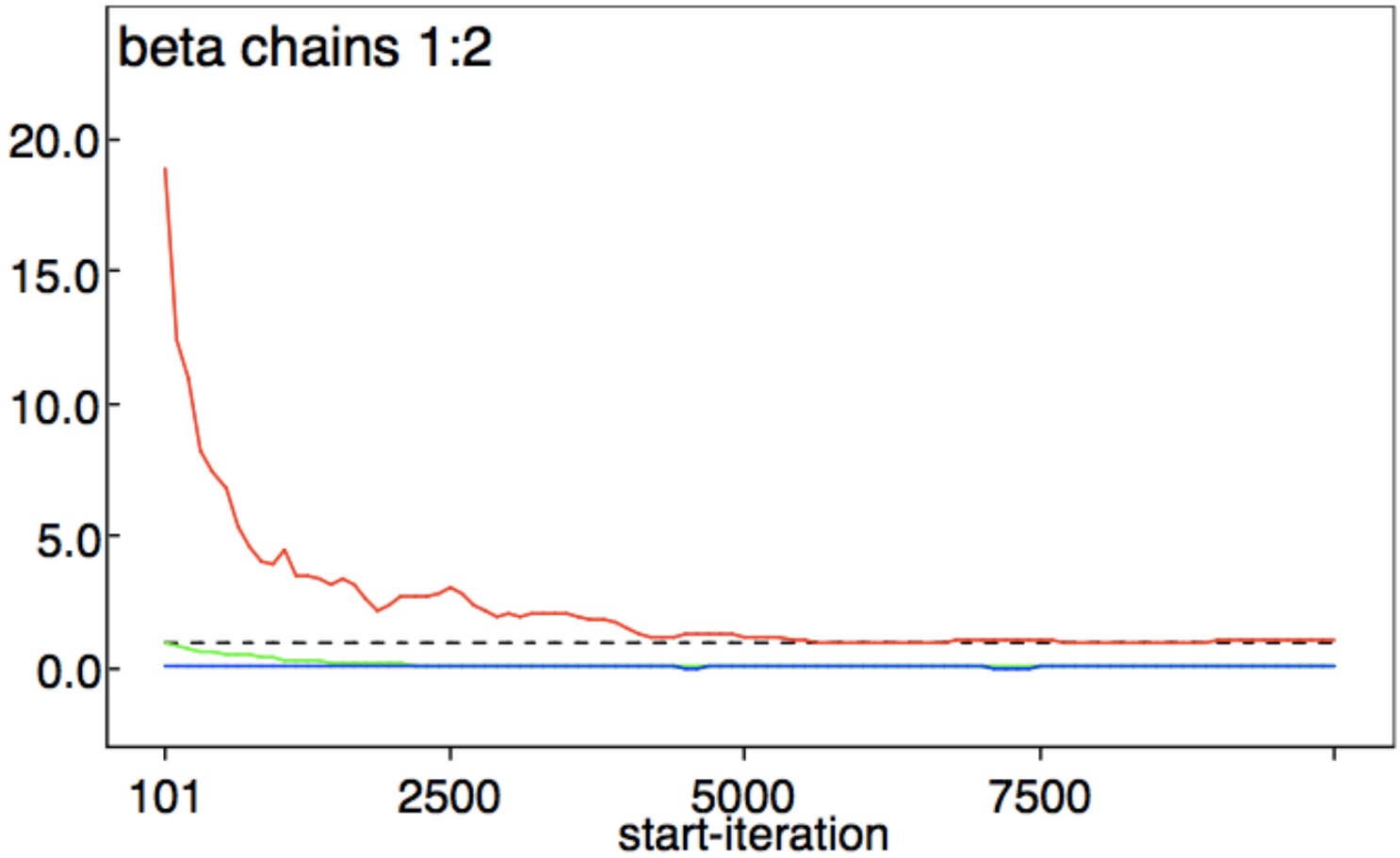
- Before clicking *compile*, set *num of chains* to 2
- Load both sets of initial values
- Monitor from the start of sampling
- Visually inspect trace/history plots to see if chains are overlapping
- Assess how much burn-in needed using the *bgr* statistic
- Check autocorrelation, as high autocorrelation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

History plots for 'un-centred' analysis





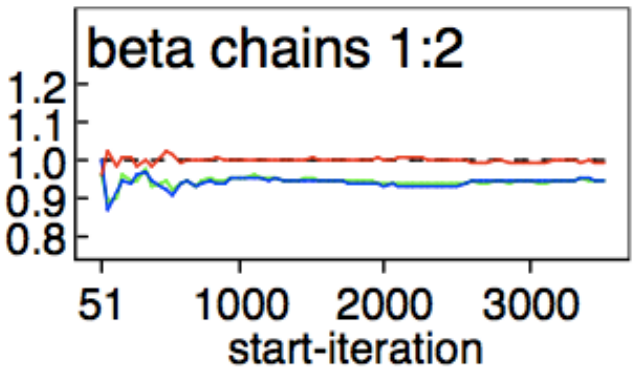
bgr plot for uncentred analysis



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

BGR convergence diagnostic in OpenBUGS



Values of Gelman Rubin statistic

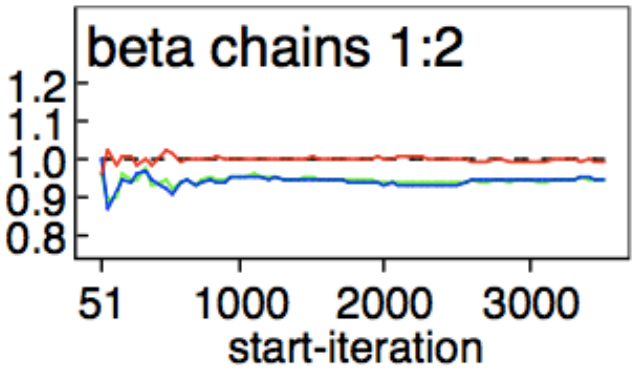
iteration range	80% interval: Unnormalized of pooled chains	Unnormalized mean within chain	Normalized as plotted of pooled chains	Normalized as plotted mean within chain	BGR ratio
51--100	7.705	7.964	0.9675	1.0	0.9675
101--200	7.119	6.967	0.8939	0.8749	1.022
151--300	7.209	7.323	0.9053	0.9195	0.9845
.....					
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

When convergence is reached:

- *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 in OpenBUGS



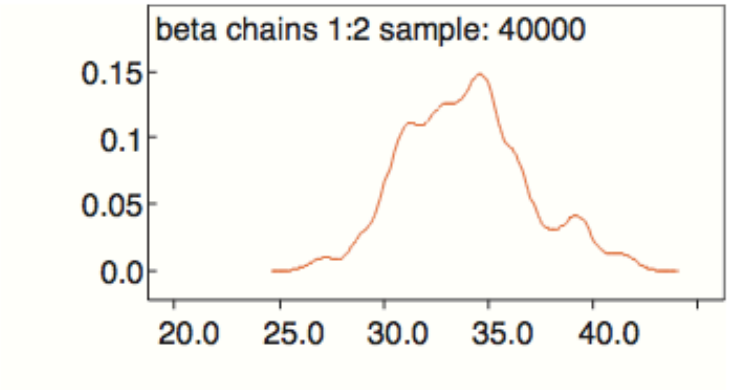
Values of Gelman Rubin statistic

iteration range	80% interval: Unnormalized of pooled chains	Unnormalized mean within chain	Normalized as plotted of pooled chains	Normalized as plotted 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

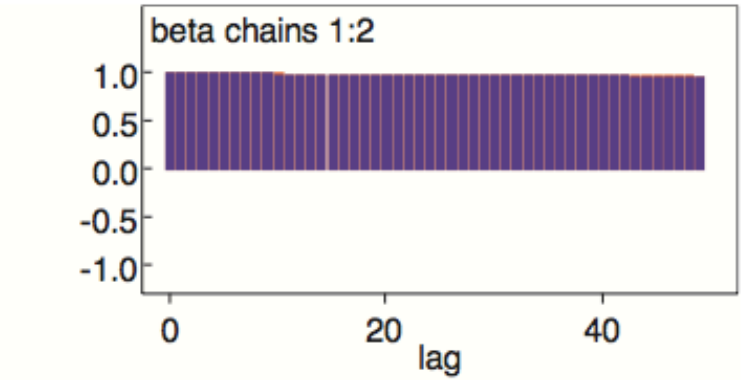
- OpenBUGS splits iterations into multiple overlapping intervals, calculates *bgr* statistics for each interval, and plots them against starting iteration of interval
  - approximate convergence can be 'read off' plot as iteration after which red *bgr* ratio line stabilises around 1, and blue and green 80% interval lines stabilise to approximately constant value (not necessarily 1)
- In OpenBUGS, right-click on the plot, select *Properties*, then click on *Data* gives values of statistics

Output for 'un-centred' analysis

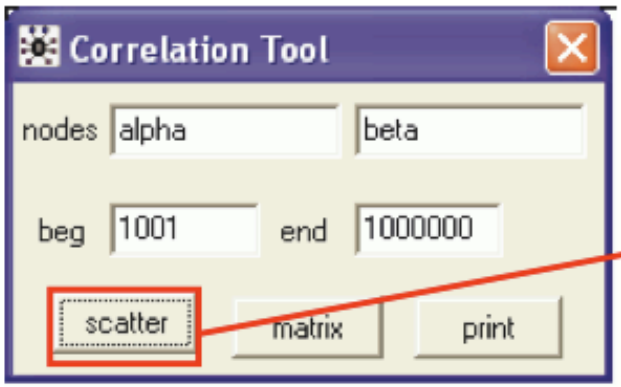
posterior density



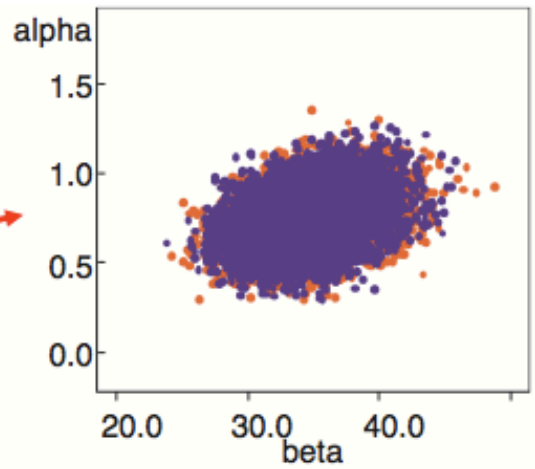
autocorrelation



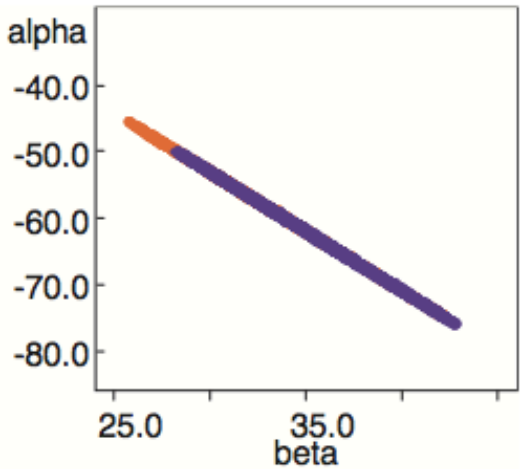
bivariate posteriors



centred



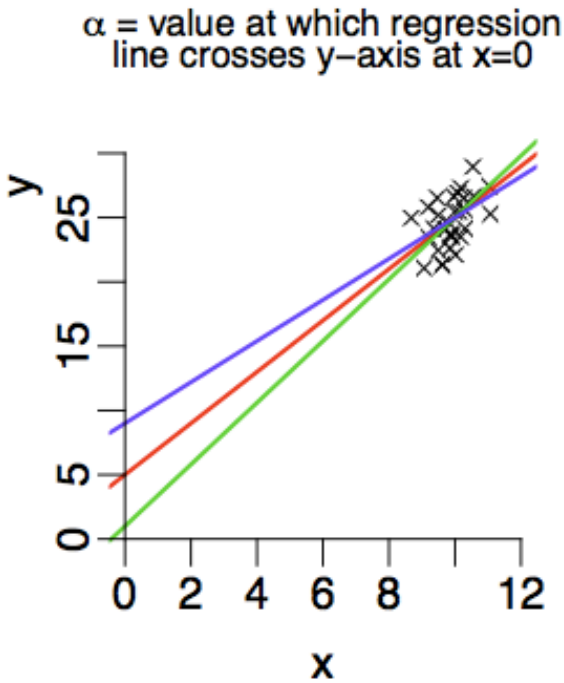
un-centred

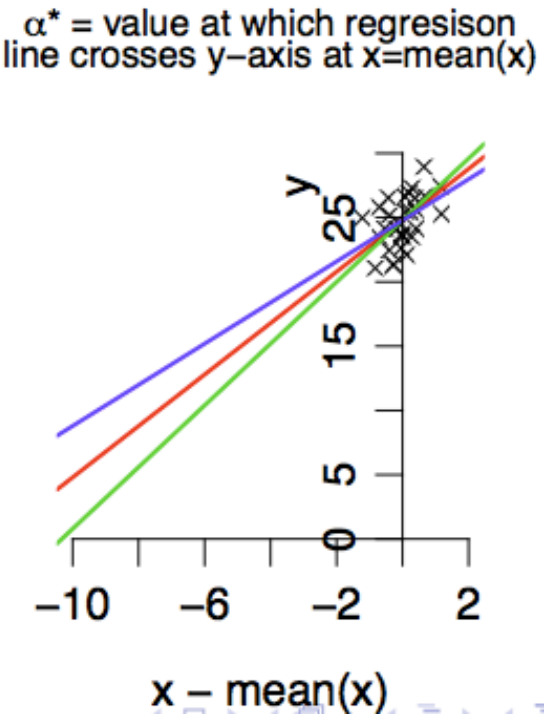


Re-fit same logistic regression, but with centred covariate

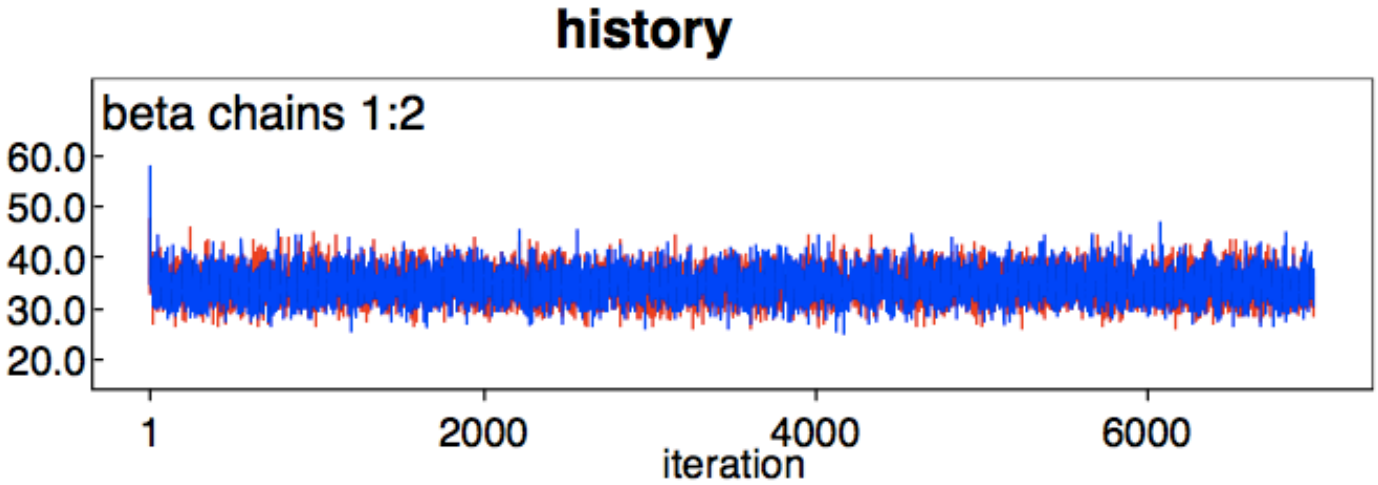
$$
\begin{aligned}
r_i &\sim \text{Binomial}(p_i, n_i) \\
\text{logit } p_i &= \alpha^* + \beta(x_i - \bar{x}) \\
\alpha^* &\sim \text{N}(0, 10000) \\
\beta &\sim \text{N}(0, 10000)
\end{aligned}
$$

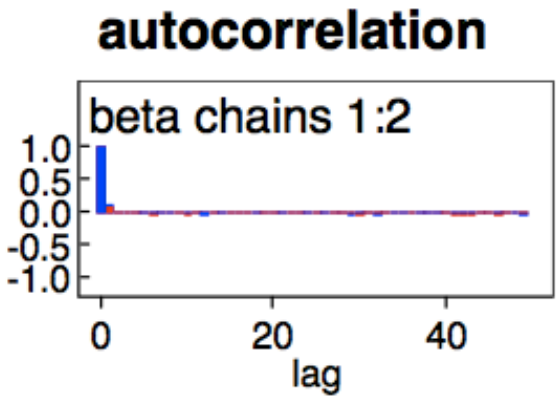
Note: $\alpha^* = \alpha + \beta\bar{x}$

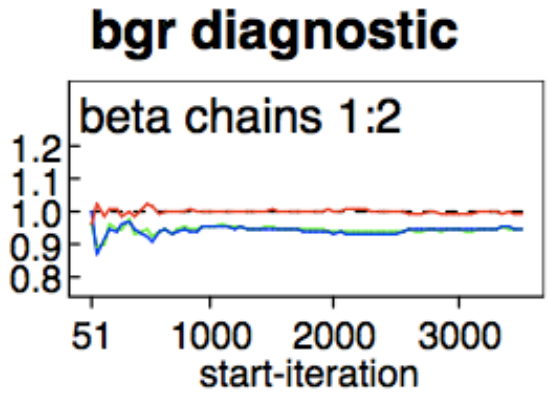




Output for ‘centred’ analysis







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

OpenBUGS steps

- Load data files
- Load multiple initial values files
- Visually inspect trace plots
- Check bgr diagnostics
- Check autocorrelation plots
- Discard burn-in samples

How many iterations after convergence?

- After convergence, further iterations are needed to obtain samples for posterior inference
- More iterations = more accurate posterior estimates
- MCMC samples are usually autocorrelated so effective sample size < actual sample size

Effective sample size and MC error

- Monte Carlo standard error (MCSE) = standard error of the mean of the posterior samples of θ as estimate of theoretical posterior expectation, $\mathbb{E}(\theta|y)$
- With independent samples, $\text{MCSE}^{ind} = s/\sqrt{N}$, where s = posterior SD of θ and N = sample size
- With autocorrelated samples, calculation of MCSE^{ac} also depends on the autocorrelation

  $\rightarrow \text{MCSE}^{ac} > \text{MCSE}^{ind}$
- An estimate of the effective sample size, N^* of an autocorrelated chain can be obtained as

$$N^* = (s/\text{MCSE}^{ac})^2$$

  - so, if $\text{MCSE}^{ac} \approx 0.05s \Rightarrow N^* \approx 1/0.05^2 = 400$
  - so, if $\text{MCSE}^{ac} \approx 0.015s \Rightarrow N^* \approx 1/0.015^2 = 4444$
  - so, if $\text{MCSE}^{ac} \approx 0.01s \Rightarrow N^* \approx 1/0.01^2 = 10000$

Deciding if your posterior sample size is large enough

- Relationship between posterior SD and MC error (previous slide) implies general rule for determining posterior sample size
  - after convergence, run MCMC simulation until the MC error $\approx$ 2 orders of magnitude smaller than the posterior SD

  $\Rightarrow$ posterior summaries will be based on effective sample size of $\approx$10,000

Output from logistic regression model with uncentred covariate

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

(MC error)/(sd) = 0.2117/3.00 = 0.07, so effective sample size $\approx 1/0.07^2 = 204$

Output from logistic regression model with centered covariate

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

(MC error)/(sd) = 0.0298/2.93 = 0.01, so effective sample size $\approx 1/0.01^2 = 10{,}000$

Key References and Further Reading

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.

Casella, G and George, EI (1992). Explaining the Gibbs sampler. *The American Statistician*, **46**, 167–174.

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

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