Bayesian Inference MCMC

BMTRY 763

Modern Posterior inference

- Unlike the usual ML estimates of risk, a Bayesian model is described by a distribution and so a range of values of risk will arise (some more likely than others)
- Posterior distributions are sampled to give a range of these values (*posterior sample*)
- This contains a large amount of information about the parameter of interest

A Bayesian Model

- A Bayesian model consists of a likelihood and prior distributions
- The product of the likelihood and the prior distributions gives the most important distribution: ***the posterior distribution***
- In Bayesian modeling all the inference about parameters is made from the posterior distribution.

Posterior Sampling

- There are two basic methods used for this:
  - Gibbs Sampling
  - Metropolis-Hastings sampling
- These are examples of Markov chain Monte Carlo (McMC) methods

Gibbs Sampling

- This requires knowledge of the conditional distributions of the parameters given the other parameters
- This is a fast algorithm as it always yields a new sample value at each iteration
- WinBUGS was developed for this method (**B**ayesian **I**nference **U**sing **G**ibbs **S**ampling)

Gibbs Sampling

- Example: set of parameters:

$$\{\alpha, \beta, \gamma\}$$

conditional distributions

$$\alpha^{new} \leftarrow [\alpha \mid \beta, \gamma, \mathrm{Y}]$$

$$\beta^{new} \leftarrow [\beta \mid \alpha^{new}, \gamma, \mathrm{Y}]$$

$$\gamma^{new} \leftarrow [\gamma \mid \alpha^{new}, \beta^{new}, \mathrm{Y}]$$

Y is the data

Metropolis-Hastings (MH)

- This is a simple algorithm for updating parameters and sampling posterior distributions.
- It does not require knowledge of the conditional distributions BUT does **not** guarantee a useful new sample value at each iteration
- Simple to implement
- WinBUGS now includes MH updating

MH sampling

$L(Y \mid \alpha, \beta, \gamma)$ likelihood

$Pr(\alpha, \beta, \gamma)$ prior distribution

$P = Pr(\alpha, \beta, \gamma \mid Y)$ posterior distribution

$P \approx L(Y \mid \alpha, \beta, \gamma).Pr(\alpha, \beta, \gamma)$

Proposal distribution:

step1) $\alpha^{new} \leftarrow q(\alpha^{new}, \alpha)$

step2) $R = \dfrac{P(\alpha^{new}, \beta, \gamma).q(\alpha^{new}, \alpha)}{P(\alpha, \beta, \gamma).q(\alpha, \alpha^{new})}$

step 3) *if* $(R > U)$ accept α^{new}

else keep α

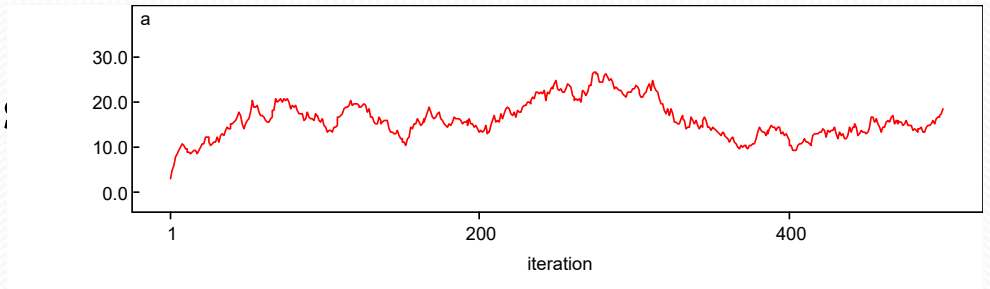
where $U \sim U(0,1)$

How Posterior Sampling Works

- In general a posterior distribution could be so complex that we must use simulation to obtain samples.
- Both Gibbs sampling and MH use simulation to generate sample values over large numbers of iterations. These methods are **iterative** and they must converge to a stable state (the Posterior distribution)
- **Convergence** must be checked

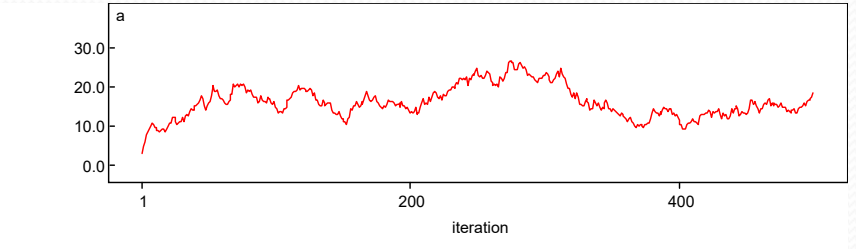
Checking Convergence

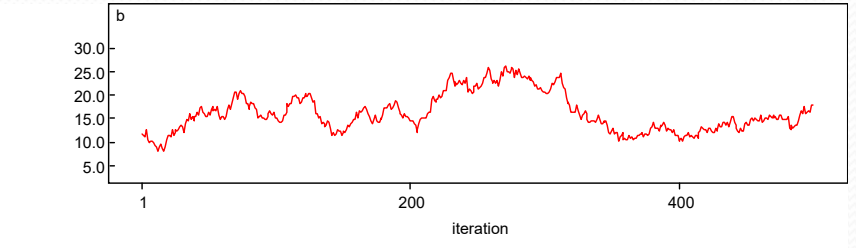
- A time series of parameters can be monitored
- Multiple parameter series can be checked
- Overall model fit measures (Deviance) can be monitored
- The final state of the sampler should be independent of the initial state



Convergence Measures

- Single Chain: Q-Q plots and cusums of parameters or overall measures
- Multi-chain: BGR statistic (used in WinBUGS)





Convergence

McMc methods require the use of diagnostics to assess whether the iterative simulations have reached the equilibrium distribution of the markov chain. Sampled chains require to be run for an intial burn-in period until they can be assumed to provide approximately correct samples from the posterior distribution of interest. This burn -in period can vary considerably between different problems. In addition, it is important to ensure that the chain manages to explore the parameter space properly so that the sampler does not ‘stick’ in local maxima of the surface of the distribution. Hence, it is crucial to ensure that a burn-in period is adequate for the problem considered. Judging convergence has been the subject of much debate and can still be regarded as art rather than science: a qualitative judgement has to be made at some stage as to whether the burn-in period is long enough..

There are a wide variety of methods now

available to assess convergence of chains within McMC. cite: robcasella and cite: chenib provide recent reviews. The available methods are largely based on checking the distributional properties of samples from the chains.

Single chain methods

First, global methods for assessing convergence have been proposed which involve monitoring functions of the posterior probability.at each iteration. These methods look or stabilisiation of the probability value. This value forms a time series, and special cusum methods have been proposed (cite: yumyk).

Second, graphical methods have been proposed which allow the comparison of the whole distribution of successive samples. Quantile-quantile plots of successive lengths of single variable output from the sampler can be used for this purpose.

Multi-chain methods

Single chain methods can, of course, be applied to each of a multiple of chains. In addition, there are methods that can only be used for multiple chains. The Gelman-Rubin statistic was proposed as a method for assessing the convergence of multiple chains via the comparison of summary measures across chains (cite: gelrubin, cite: brookgel, cite: robcasella, Ch. 8). There is some debate about whether it is useful to run one long chain as opposed to multiple chains with different start points. The advantage of multiple chains is that they provide evidence for the robustness of convergence across different subspaces. However, as long as a single chain samples the parameter space adequately, then these have benefits. The reader is referred to cite: robcasella, chapter 8 for a thorough discussion of diagnostics and their use.

M-H versus Gibbs Algorithms

- There are advantages and disadvantages to M-H and Gibbs methods. The Gibbs Sampler provides a *single* new value for each θ at each iteration, but requires the evaluation of a conditional distribution.
- On the other hand the M-H step does not require evaluation of a conditional distribution but does not guarantee the acceptance of a new value.
- In addition, block updates of parameters are available in M-H, but not usually in Gibbs steps (unless joint conditional distributions are available).
- If conditional distributions are difficult to obtain or computationally expensive, then M-H can be used and is usually available.

- In summary, the Gibbs Sampler may provide faster convergence of the chain if the computation of the conditional distributions at each iteration are not time consuming.
- The M-H step will usually be faster at each iteration, but will not necessarily guarantee exploration.
- In straightforward hierarchical models where conditional distributions are easily obtained and simulated from, then the Gibbs Sampler is likely to be favoured.
- In more complex problems, such as many arising in spatial statistics, resort may be required to the M-H algorithm.

A simple M-H example Assume that for m regions, the count n_i $i = 1, \ldots, m$ is observed. In addition, the expected count in the i th region, e_i is also observed. Assume also that the counts are independently distributed and have a Poisson distribution with $E(n_i) = \theta . e_i$, where θ is a constant parameter describing the relative risk over the whole study window. The likelihood in this case, bar a constant, is given by

$$L(\theta) = \exp(-\theta \sum_{i=1}^{m} e_i) . \prod_{i=1}^{m} (\theta e_i)^{n_i} .$$

Assuming a flat prior distribution for θ, then the M-H sampler for this problem reduces to a stochastic exploration of the likeihood surface. Hence the following sampler criterion is found for the θ parameter in this case:

$$\frac{L(\theta')}{L(\theta)} = \exp\{s_e(\theta - \theta')\} . \left(\frac{\theta'}{\theta} \right)^{s_n}$$

$$where\ s_e = \sum_{i=1}^{m} e_i \text{ and } s_n = \sum_{i=1}^{m} n_i .$$

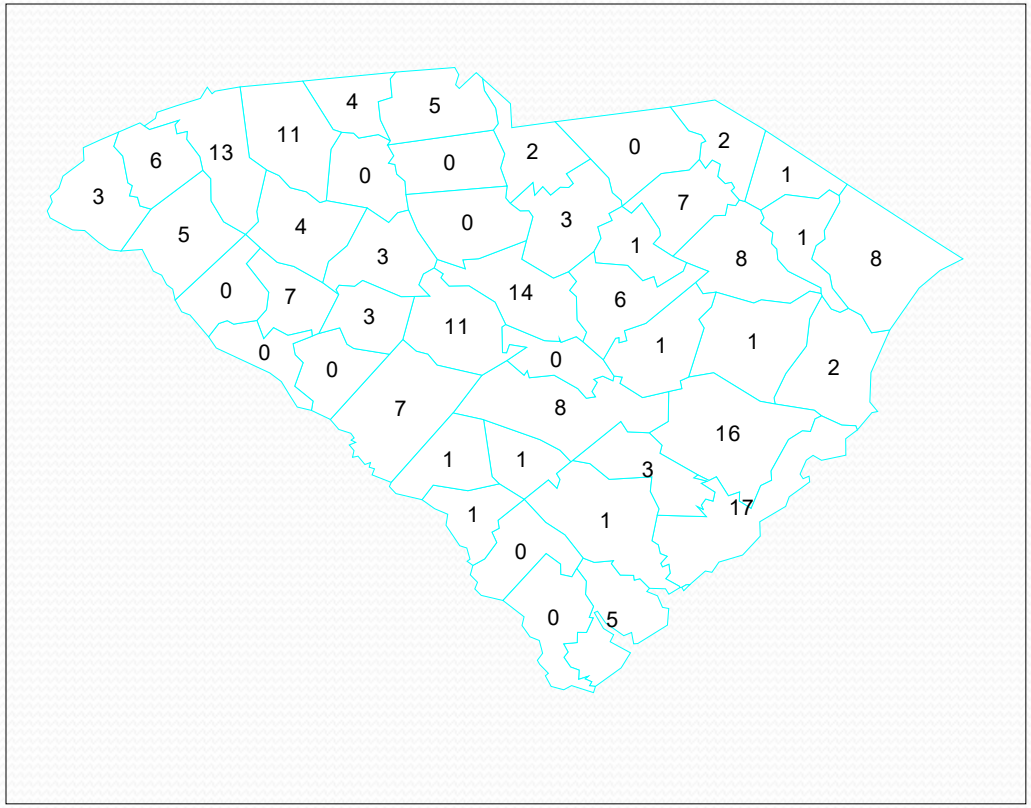
Special Methods

- Alternative methods exist for poseterior sampling when the basic Gibbs or M-H updates are not feasible or appropriate.
- For example, if the range of the parameters are restricted then *slice sampling* can be used (Robert and Casella (1999), Ch. 7; see also Radford Neal's web site: http://www.cs.toronto.edu/~radford/).
- When exact conditional ditributions are not available but the posterior is log-concave then adaptive rejection sampling algoirthms can be used.
- The most general of these algorithms (adaptive rejection sampling (ARS) algorithm; Robert and Casella (1999) p.57-59) has wide applicability for continuous distributions, although may not be efficient for specific cases.
- Block updating can also be used to effect in some situations.

- When generalised linear model components are included then block updating of the covariate parameters can be effected via multuivariate updating.

A WinBUGS example

- South Carolina congenital deaths 1990



WinBUGS Code for Poisson-Gamma Model

```
model
{
for (i in 1:m)
{
    # Poisson likelihood for observed counts
    y[i]~dpois(mu[i])
    mu[i]<-e[i]*theta[i]
    # Relative Risk
    theta[i]~dgamma(a,b)
}

# Prior distributions for "population" parameters
a~dexp(0.1)
b~dexp(0.1)

# Population mean and population variance
mean<-a/b
var<-a/pow(b,2)
}
```

- Poisson likelihood
- Gamma (a,b) prior
- Exponential hyper-priors for a and b

Data Entry

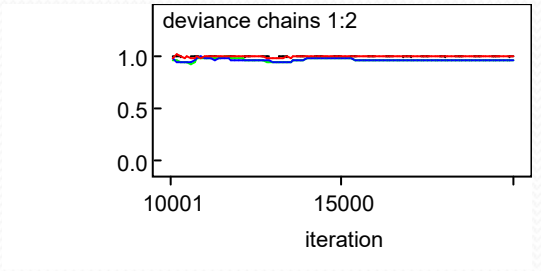
```
list(
m=46,
y=c(0,7,1,5,1,1,5,16,0,17,4,0,0,1,1,7,1,3,0,0,8,2,13,7,0,8,0,3,2,4,1,11,0,1,2,3,3,8,6,14,3,11,
    6,0,1,5),
e=c(1.1297788827,6.667008775,0.650279674,6.988864371,0.95571406,1.123210345,5
    .908349156,8.539026017,0.601016062,18.92051111,2.272694617,1.73736337,2.019
    808077,1.688099759,1.747216093,3.221840201,1.835890594,5.221942834,0.9787
    03751,1.254579976,6.407553754,2.676656232,16.57884744,3.077333607,1.08708
    3697,7.606301637,1.018114641,2.15774619,2.844152512,2.955816698,0.985272233
    ,9.22871658,0.38097193,1.855596038,1.579719813,1.579719813,2.647098065,4.79
    1707292,4.144711859,15.70852363,0.765228101,11.32077795,6.256478678,1.50089
    8035,2.085492893, 7.297583004))
```

Initial Values

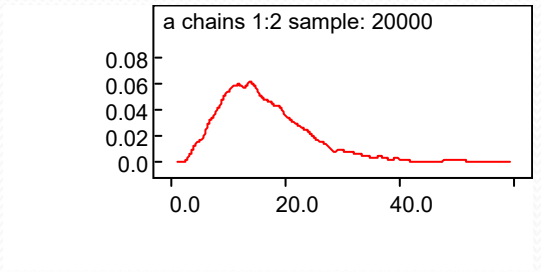
- list(a=0.1,b=0.1,theta=c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1))
- list(a=0.2,b=0.2,theta=c(1,1,1,1.3,1,1,1,1.5,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0.1,1,1,1,1,1,1,1))

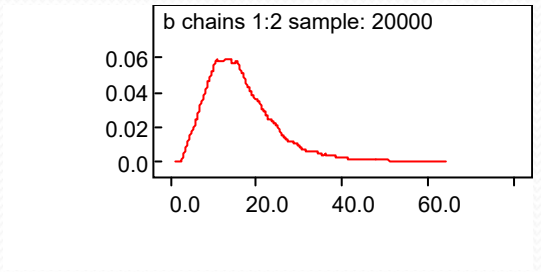
Run: 10000 burn-in

- BGR statistic

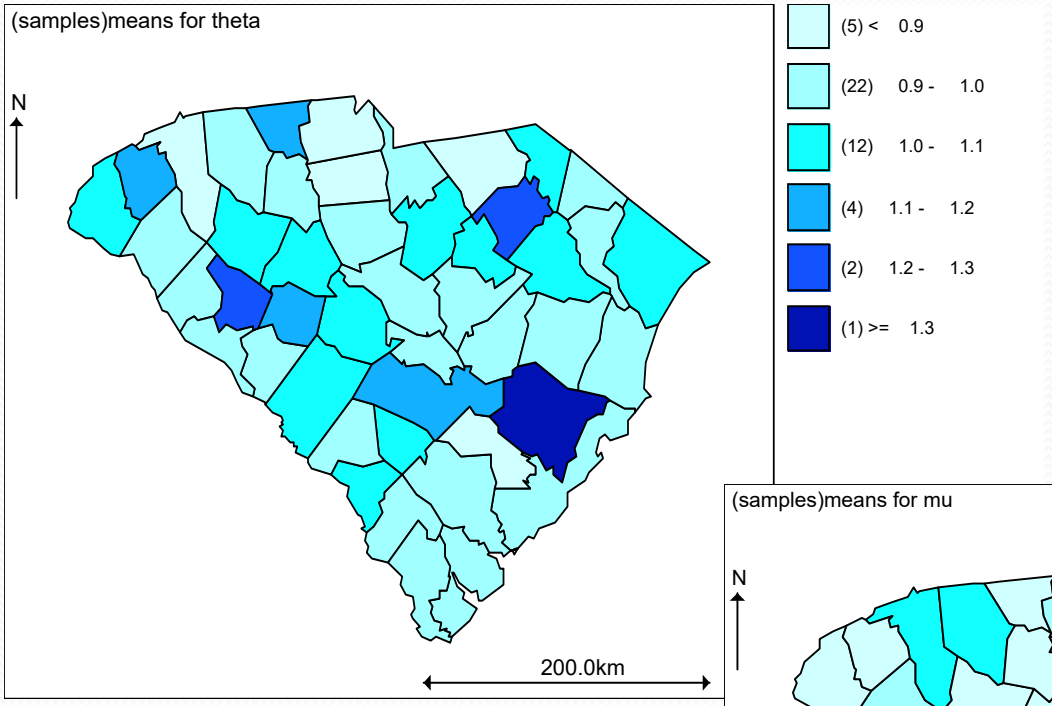


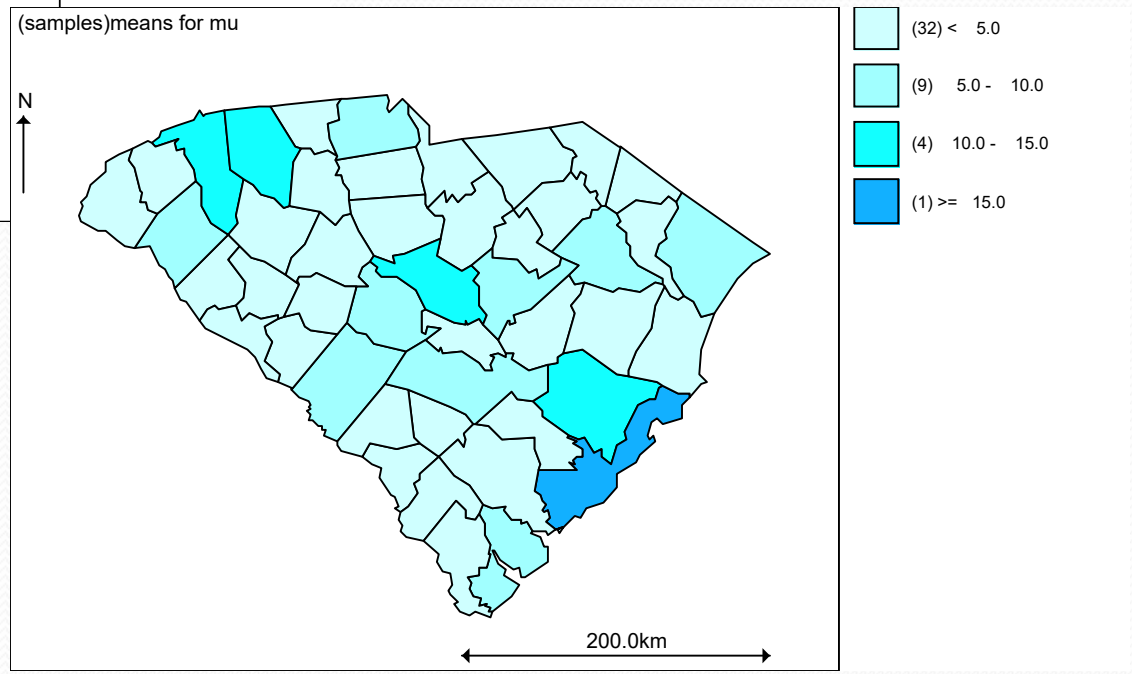
- Posterior Marginal density estimates





Relative risk and mean maps





Posterior Summary Statistics

- a small sample of summary statistics available (a, b, and first 10 regions for theta: mean, sd, MC error 2.5%,median, 95% CI)

Nodestatistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
a	16.13	6.184	0.5473	6.429	15.71	29.17	10001	10000
b	16.01	6.195	0.5483	6.386	15.59	29.62	10001	10000
theta[1]	0.9392	0.2624	0.004627	0.4738	0.9261	1.511	10001	10000
theta[2]	1.024	0.2313	0.002664	0.6236	1.006	1.529	10001	10000
theta[3]	1.034	0.2839	0.003567	0.5578	1.008	1.674	10001	10000
theta[4]	0.9111	0.2119	0.003457	0.5321	0.8986	1.364	10001	10000
theta[5]	1.02	0.2802	0.003969	0.5459	0.9956	1.638	10001	10000
theta[6]	1.002	0.2706	0.003371	0.5391	0.9766	1.606	10001	10000
theta[7]	0.9624	0.2277	0.002804	0.5637	0.9443	1.459	10001	10000
theta[8]	1.324	0.2608	0.00755	0.8846	1.301	1.913	10001	10000
theta[9]	0.9715	0.276	0.003904	0.4931	0.9477	1.582	10001	10000
theta[10]	0.9474	0.1705	0.002318	0.6418	0.9369	1.315	10001	10000