MCMC for Cut Models
or
Chasing a Moving Target with MCMC

Martyn Plummer

International Agency for Research on Cancer

MCMSki

Chamonix, 6 Jan 2014

Cut models

What do we want to do?

1. Generate some random variables under one model
2. Analyze them as if they were data in another model *not necessarily coherent* with the first.

Cut models

What do we want to do?

1. Generate some random variables under one model
2. Analyze them as if they were data in another model *not necessarily coherent* with the first.

Why? (not the topic of this talk)

- ▶ Software validation (if models are coherent)
- ▶ Investigating model robustness (if they are not)
- ▶ Resolving potential conflict between data sources ("cutting feedback")
- ▶ Overcoming numerical difficulties (e.g. MCMC without cuts has poor mixing).

Cut models

What do we want to do?

1. Generate some random variables under one model
2. Analyze them as if they were data in another model *not necessarily coherent* with the first.

Why? (not the topic of this talk)

▶ Software validation (if models are coherent)
▶ Investigating model robustness (if they are not)
▶ Resolving potential conflict between data sources ("cutting feedback")
▶ Overcoming numerical difficulties (e.g. MCMC without cuts has poor mixing).

How?

▶ Multiple imputation, or
▶ Coupled Markov chains (the topic of this talk)

A simple example

Multiple imputation (MI) in JAGS

```
data {
   z ~ dnorm(0, 1)
}
model {
   z ~ dnorm(theta, 1)
   theta ~ dnorm(0, 1.0E-3)
}
```

Each chain generates a single value of z at the start of the run, which is treated as data for the rest of the run.

Advantages of MI

► Few replicates ($n \approx 20$) required to estimate mean and variance

► Just pool samples from each chain. No need for frequentist combining rules (?)

► Relatively easy to do parallel runs on a cluster (see the `dclone` package for R)

Coupled chains

Try simultaneously sampling (z, θ) at each iteration using the *naive cut algorithm*:

1. Simulate a new value of z at each iteration
2. Update θ with a standard MCMC update using the current value of z

This is a *cut* model because z is updated without reference to the current value of θ.

▶ If we were sampling from a coherent probability distribution, we would require this.

Closed-form solution

This simple example has a closed form solution.
The density of θ is given by the mixture:

$$p^*(\theta) = \int p(\theta \mid z)\phi(z)d\theta$$

Taking the limit of an improper, flat prior on θ

$$
\begin{aligned}
\theta \mid z &\sim N(z, 1) \\
z &\sim N(0, 1) \\
\theta &\sim N(0, 2)
\end{aligned}
$$

Results of naive cut algorithm by sampling method [1]
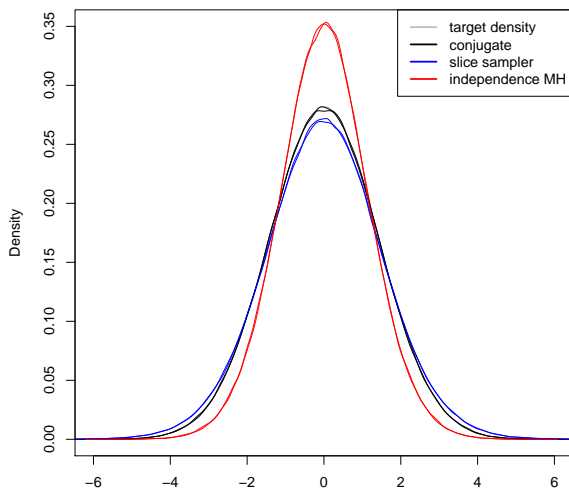
Different sampling algorithms converge to different limiting distributions
Only the conjugate sampler is correct.

[1] Using custom R code, 2 parallel chains

Cuts in measurement error models: Motivating example

There is an ecological association between HPV prevalence and cervical cancer incidence[2].
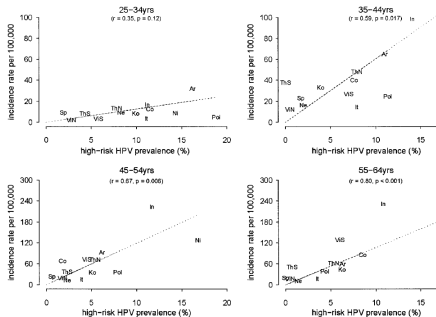
HPV is a necessary cause of cancer, but risk is modulated by other cofactors: smoking, childbirth, hormonal contraceptives, ...

[2] Maucort-Boulch *et al*, Cancer Epidemiology Biomarkers Prev, 2008

A toy measurement error model for the ecological data

We experimented with a functional measurement error model for these data, with a Poisson regression model for incidence and a binomial model for (age-specific) prevalence:

$$
\begin{aligned}
Y_i &\sim \text{Poisson}(N_i \exp(\lambda_i)) && \text{Cancer incidence data} \\
\lambda_i &= \theta_1 + \theta_2 \varphi_i && \text{Incidence rates} \\
Z_i &\sim \text{Bin}(n_i, \varphi_i) && \text{HPV prevalence data}
\end{aligned}
$$

A cut measurement error model
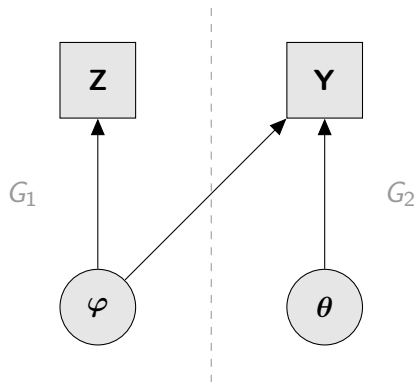
In a cut model, the graph G is divided into two sub-graphs G_1, G_2.

- Nodes in G_1 are updated ignoring nodes in G_2.
- Nodes in G_2 are updated as normal (naive cut algorithm).

- In our example, we use only prevalence survey data ($\mathbf{Z}$) to estimate HPV prevalence (φ)
- Feedback from cancer incidence rates ($\mathbf{Y}$) via the putative dose-response relationship is cut.

World Health
Organization

Definition of cut model in BUGS

OpenBUGS provides a cut function to denote cuts, and implements the naive cut algorithm when sampling cut models.

```
model {

    ## Disease model
    for (j in 1:13) {
        ncases[j] ~ dpois(mean[j])
        log(mean[j]) <- theta[1] + phi.cut[j] * theta[2] + log(Npop[j] * 1.0E-3)
    }

    theta[1] ~ dnorm(0, 1.0E-3)
    theta[2] ~ dnorm(0, 1.0E-3)

    ## Cuts
    for (j in 1:13) {
        p.cut[j] <- cut(p[j])
    }

    ## Measurement model - below the cut
    for (j in 1:13) {
        npositives[j] ~ dbin(phi[j], Nsubjects[j])
    }

    ## Exposure model - below the cut
    for (j in 1:13) {
        phi[j] ~ dunif(0, 1)
    }
}
```

World Health
Organization
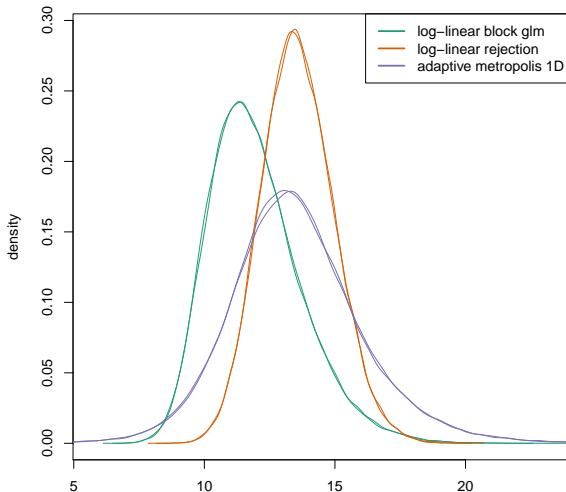
Different update methods converge to different limiting distributions.
The correct distribution could be calculated by multiple imputation, but is not shown here.

Why the naive cut algorithm does not work (1/2)

The target density of a cut model is the mixture:

$$p^*(\boldsymbol{\theta}) = \int p(\boldsymbol{\varphi} \mid \mathbf{Z}) p(\boldsymbol{\theta} \mid \boldsymbol{\varphi}, \mathbf{Y}) d\boldsymbol{\varphi}$$

This is the sampling density if we sample directly $\boldsymbol{\varphi}$ then $\boldsymbol{\theta}$ at each iteration.

- ▶ This is why the conjugate sampler worked in our simple example.

Why the naive cut algorithm does not work (2/2)

In general, MCMC methods do not sample directly from the target density but supply a reversible transition $\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t$ at iteration t. The transition is in detailed balance with the full conditional distribution:

$$p(\boldsymbol{\theta}^{t-1} \mid \mathbf{Y}, \boldsymbol{\varphi}^t) p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^t) =$$
$$p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^t) p(\boldsymbol{\theta}^t \to \boldsymbol{\theta}^{t-1} \mid \boldsymbol{\varphi}^t)$$

But for $p^*(\boldsymbol{\theta})$ to be the stationary distribution we need:

$$p(\boldsymbol{\theta}^{t-1} \mid \mathbf{Y}, \boldsymbol{\varphi}^{t-1}) p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^{t-1}, \boldsymbol{\varphi}^t) =$$
$$p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^t) p(\boldsymbol{\theta}^t \to \boldsymbol{\theta}^{t-1} \mid \boldsymbol{\varphi}^t, \boldsymbol{\varphi}^{t-1})$$

Why the naive cut algorithm does not work (2/2)

In general, MCMC methods do not sample directly from the target density but supply a reversible transition $\theta^{t-1} \to \theta^t$ at iteration t. The transition is in detailed balance with the full conditional distribution:

$$p(\theta^{t-1} \mid \mathbf{Y}, \varphi^t)p(\theta^{t-1} \to \theta^t \mid \varphi^t) =$$
$$p(\theta^t \mid \mathbf{Y}, \varphi^t)p(\theta^t \to \theta^{t-1} \mid \varphi^t)$$

But for $p^*(\boldsymbol{\theta})$ to be the stationary distribution we need:

$$p(\theta^{t-1} \mid \mathbf{Y}, \varphi^{t-1})p(\theta^{t-1} \to \theta^t \mid \varphi^{t-1}, \varphi^t) =$$
$$p(\theta^t \mid \mathbf{Y}, \varphi^t)p(\theta^t \to \theta^{t-1} \mid \varphi^t, \varphi^{t-1})$$

The balance relation uses the current and previous values of φ.

Maybe we can add a Metropolis-Hastings acceptance step, treating the move $\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t$ as a proposal to be accepted with probability $\min(1, R)$ where

$$R = \frac{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^t) p(\boldsymbol{\theta}^t \to \boldsymbol{\theta}^{t-1} \mid \boldsymbol{\varphi}^{t-1})}{p(\boldsymbol{\theta}^{t-1} \mid \mathbf{Y}, \boldsymbol{\varphi}^{t-1}) p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^t)}$$

Note that $R = 1$ in the case of direct sampling:

$$p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}) = p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi})$$

Can we modify a standard MCMC update? (2/2)

For a standard MCMC update (in detailed balance with the full conditional distribution) the acceptance ratio can be rewritten in terms of forward transitions:

$$R = \frac{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^t)}{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^{t-1})} \frac{p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^{t-1})}{p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^t)}$$

But this requires

- Explicit expressions for the transition probabilities (not available for slice sampling, Hamiltonian Monte Monte Carlo).
- Evaluation of the ratio of two normalized densities

Can we modify a standard MCMC update? (2/2)

For a standard MCMC update (in detailed balance with the full conditional distribution) the acceptance ratio can be rewritten in terms of forward transitions:

$$R = \frac{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^t)}{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^{t-1})} \frac{p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^{t-1})}{p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^t)}$$

But this requires

▶ Explicit expressions for the transition probabilities (not available for slice sampling, Hamiltonian Monte Monte Carlo).

▶ Evaluation of the ratio of two normalized densities

Can we modify a standard MCMC update? (2/2)

For a standard MCMC update (in detailed balance with the full conditional distribution) the acceptance ratio can be rewritten in terms of forward transitions:

$$R = \frac{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^t)}{p(\boldsymbol{\theta}^t \mid \mathbf{Y}, \boldsymbol{\varphi}^{t-1})} \frac{p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^{t-1})}{p(\boldsymbol{\theta}^{t-1} \to \boldsymbol{\theta}^t \mid \boldsymbol{\varphi}^t)}$$

But this requires

- Explicit expressions for the transition probabilities (not available for slice sampling, Hamiltonian Monte Monte Carlo).
- Evaluation of the ratio of two normalized densities
  - Unsuitable for most applications of MCMC where we have only unnormalized densities.

Latest attempt (not clear if this actually works)

Inspired by bridge sampling:

- Put a pseudo-prior $\pi(.)$ on φ (possibly flat)
- Run a Markov chain on (θ, φ):
  1. Standard update of θ given φ^{t-1}
  2. Attempt to update $\varphi^{t-1} \to \varphi^t$
  3. Second standard update of θ given current φ
- At each attempt to update φ, consider the move $\varphi^{t-1} \to \varphi^t$ as a Metropolis-Hastings *proposal* with acceptance ratio

$$\frac{p(\mathbf{Y} \mid \theta, \varphi^t)\pi(\varphi^t)}{p(\mathbf{Y} \mid \theta, \varphi^{t-1})\pi(\varphi^{t-1})}$$

- For each value φ^t this generates a sequence of samples $\theta^{t1}, \theta^{t2}, \ldots \theta^{tn_t}$ where n_t is the number of iterations until the next jump.

Reweighting the samples (latest attempt cont.)

- This MCMC process generates a sequence of samples with the correct conditional distribution $\boldsymbol{\theta} \mid \varphi$ but the wrong marginal distribution for φ.
- It must be reweighted:
  - Give weight $1/n_t$ to each of $\boldsymbol{\theta}^{t1} \ldots \boldsymbol{\theta}^{tn_t}$

Efficiency considerations

- Choice of pseudo-prior $\pi(\varphi)$?
- If the jump $\varphi^{t-1} \rightarrow \varphi^t$ is large, the sampler may get stuck at φ^{t-1} for a long time.
  - Tempering may be required to smoothly interpolate between the two non-overlapping distributions.
  - Likely to occur in cases of prior-data conflict.

Conclusions

- MCMC in cut models is harder than it looks.
- The naive cut algorithm (currently implemented in OpenBUGS) does not work:
  - Limiting distribution depends on transition kernel of chain.
  - Informally, MCMC updates are trying to reach equilibrium with current data values, but changes in data are forcing the chain away from equilibrium (Hence the title: chasing a moving target).
- Multiple imputation is currently the only working solution.