The University of Newcastle

Kerrie Mengersen

Introduction to
Bayesian Modelling - 7

Armidale 2004

MCMC Algorithms

A gourmet of samplers

- Rejection methods
- Variance reduction methods
- Adaptive rejection sampling
- Umbrella sampling
- Slice sampling

etc etc etc

EM Algorithm

the distribution of the sample $\underline{x}$ can be written as

$$\begin{aligned} f(\underline{x}|\theta) &= \int g(\underline{x},\underline{z}|\theta)\,\mathrm{d}\underline{z} \\ &= \int f(\underline{x}|\theta)\,k(\underline{z}|\underline{x},\theta)\,\mathrm{d}\underline{z} \end{aligned} \tag{1.12}$$

leading to a *complete* (unobserved) log-likelihood

$$\mathbb{L}^c(\theta|\underline{x},\underline{z}) = \mathbb{L}(\theta|\underline{x}) + \log k(\underline{z}|\underline{x},\theta)$$

where $\mathbb{L}$ is the observed log-likelihood. The EM algorithm is then based on a sequence of completions of the missing variables $\underline{z}$ based on $k(\underline{z}|\underline{x},\theta)$ and of maximisations of the expected complete log-likelihood (in θ):

EM Algorithm

0. Initialization: choose $\theta^{(0)}$,

1. Step t. For $t = 1, \ldots$

1.1 The E-step, compute

$$Q\left(\theta|\theta^{(t-1)}, \underline{x}\right) = \mathbb{E}_{\theta^{(t-1)}}\left[\log \mathbb{L}^c\left(\theta|\underline{x}, \underline{Z}\right)\right],$$

where $\underline{Z} \sim k\left(\underline{z}|\theta^{(t-1)}, \underline{x}\right)$.

1.2 The M-step, maximize $Q\left(\theta|\theta^{(t-1)}, \underline{x}\right)$ in θ and take

$$\theta^{(t)} = \arg\max_{\theta} Q\left(\theta|\theta^{(t-1)}, \underline{x}\right).$$

EM for mixtures

For an illustration in our setup, consider again the special mixture of normal distributions (1.7) where all parameters but $\underline{\theta} = (\mu_1, \mu_2)$ are known. For a simulated dataset of 500 observations and true values $p = 0.7$ and $(\mu_1, \mu_2) = (0, 2.5)$, the log-likelihood is still bimodal and running the EM algorithm on this model means, at iteration t, computing the expected allocations

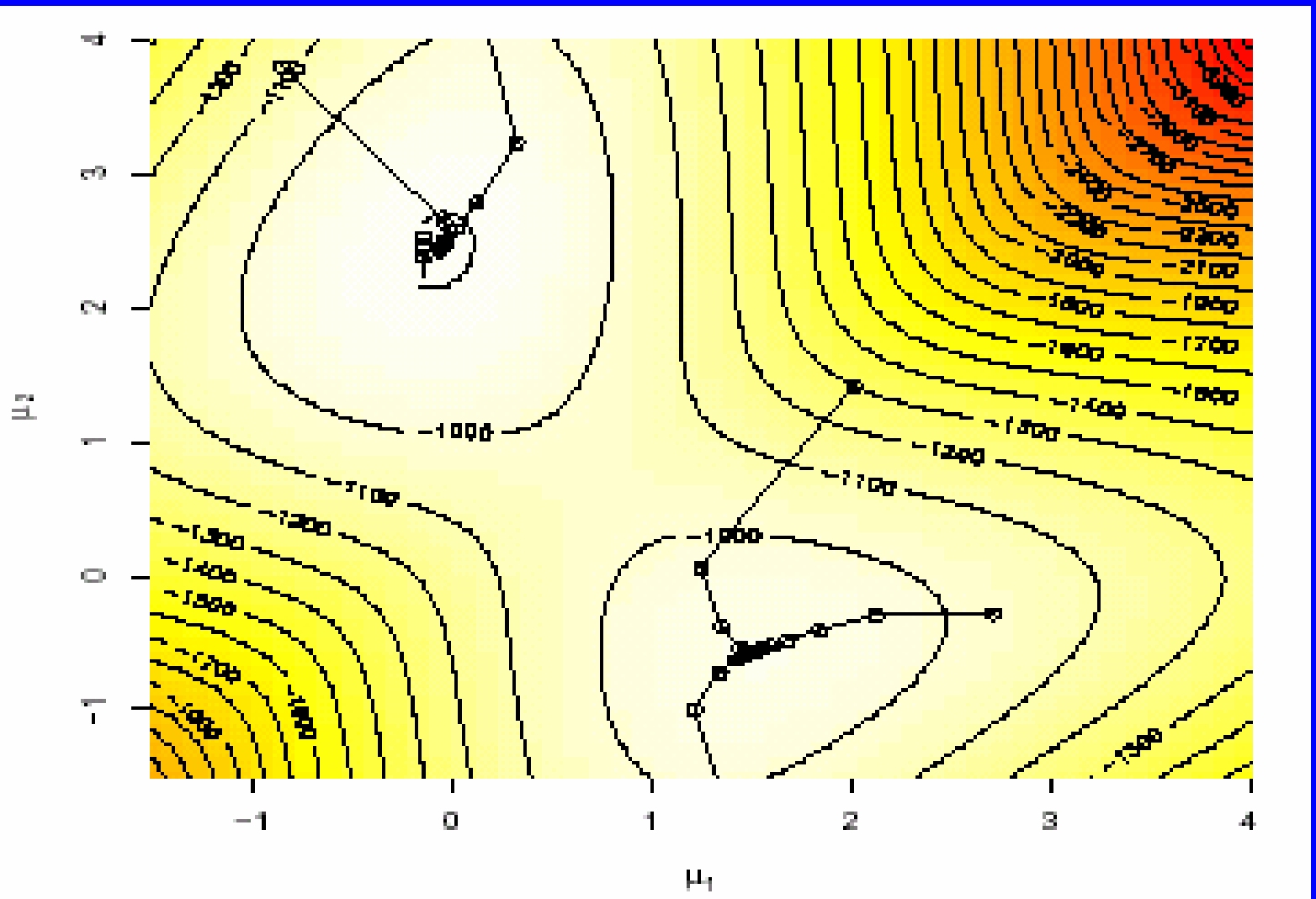
$$z_i^{(t-1)} = \mathbb{P}(Z_i = 1 | \underline{x}, \underline{\theta}^{(t-1)})$$

in the E-step and the corresponding posterior means

$$\mu_1^{(t)} = \sum_{i=1}^{n} \left(1 - z_i^{(t-1)}\right) x_i \Big/ \sum_{i=1}^{n} \left(1 - z_i^{(t-1)}\right)$$

$$\mu_2^{(t)} = \sum_{i=1}^{n} z_i^{(t-1)} x_i \Big/ \sum_{i=1}^{n} z_i^{(t-1)}$$

in the M-step. As shown on Figure 8 for five runs of EM with starting points chosen at random, the algorithm always converges to a mode of the likelihood but only two out of five sequences are attracted by the higher and more significant mode, while the other three go to the lower spurious mode (even though the likelihood is considerably smaller). This is because the starting points happened to be in the domain of attraction of the lower mode.


−1
0
1
2
3
4
μ_1
μ_2
−1000
−1100
−1200
−1300
−1400
−1500
−1600
−1700
−1800
−1900
−2000
−2100

Slice Sampling

- Markov chain sampling method that adapts to characteristics of the distribution being sampled

- Constructed using the principle that one can sample from a distribution by sampling uniformly from the region under the plot of its density function. Construct a Markov chain that converges to this uniform distribution by alternating uniform sampling in the vertical direction with uniform sampling from the horizontal `slice' defined by the current vertical position, or more generally, with some update that leaves the uniform distribution over this slice invariant.

Radford Neal (Annals of Statistics, 31, 705-767)

Slice sampling (cont)

- Introduced by Wakefield et al as a 'ratio-of-uniforms' method for generating random variables; developed by Neal as a method for 'slicing' distribution.

- If $f(\theta)$ can be written as a product $\Pi\, f_i(\theta)$, where the f_i's are positive functions (not necessarily densities), then f can be expressed as $\Pi\, I_{0<wi<fi(\theta)}$ where *I* is the indicator function.

- Thus at the *t*th iteration, simulate $\theta^{(t)}$ by generating k uniform random variables $w_1^{(t)} \sim U(0, f_1(\theta^{(t-1)}), \ldots, w_k^{(t)} \sim U(0, f_k(\theta^{(t-1)})$

- Take $\theta^{(t)} = U(A^{(t)})$, where $A(t) = \{ y: f_i(y) > w_i(t), i=1,\ldots,k \}$

- This chain converges geometrically when f is bounded and converges uniformly when k=1.

- For any MH algorithm, it is always possible to construct a better slice sampler (faster convergence)

Why slice sampling?

- Slice sampling methods are more efficient than Gibbs, easily implemented for univariate distributions, and can be used to sample from a multivariate distribution by updating each variable in turn.
- Slice sampling has the ability to adaptively choose the magnitude of changes made. It is therefore attractive for routine and automated use.
- Methods that update all variables simultaneously are also possible. These methods can adaptively choose the magnitudes of changes made to each variable, based on the local properties of the density function. More ambitiously, such methods could potentially adapt to the dependencies between variables by constructing local quadratic approximations.

Hybrid Methods

•Employ combinations of MCMC algorithms in a single analysis

- different MCMC algorithms for different parameters
- insert a MH step with larger dispersion or probability of acceptance at every nth iteration
- mode jumping proposals
- methods based on tempering
- methods based on regeneration

• Can be almost automatically constructed to ensure uniform convergence to the target distribution

Perfect sampling

Another development in MCMC that has created its own domain of research is perfect simulation, also known as exact sampling. As described in the original paper by Propp and Wilson [76] and subsequently by Kendall [47], the aim of perfect simulation is to sample directly from the stationary distribution $f(\mu)$.

Although this appears to be exactly what MCMC is aiming to avoid, there are several reasons for pursuing the idea. First, independent samples drawn directly from $f(\mu)$ may be preferable to samples obtained from MCMC algorithms, depending on the degree of dependence in the latter and the comparative computational time and complexity. Second, a single sample drawn directly from $f(\mu)$ can be used as a starting point for standard MCMC algorithms. This avoids the well-known problem of burn-in, in which the initial value of the chain may induce long-term bias.

Perfect sampling (cont)

For a finite state-space X of size k, Propp and Wilson [76] proposed an exact sampling algorithm called coupling from the past (CFTP). Here, k chains corresponding to all possible starting points in X are started at time t and run in parallel back in time, often in a coupled manner, until all the chains coalesce (take the same value) at time 0 or earlier. The realisations of the chains at time 0 then form a single $\mu^{(0)}$ from the required distribution. If the chains have not coalesced by time 0, the chains are run again from time $2t$ and this is continued until the desired result is achieved.

It can be shown that coalescence under CFTP will indeed occur in a finite number of backward iterations. In practice, however, the computation time can be unacceptably long. Alternative algorithms have been developed to improve this and other aspects of the original CFTP idea. For example, Fill [33] proposed an interruptible algorithm for perfect simulation, in which the chains can be stopped before reaching time 0 but maintain the properties of the CFTP algorithm. As a second example, if a monotonicity constraint can be constructed, so that there is stochastically a maximum state x_1 and a minimum state x_0 in X, then CFTP reduces to running only two chains from x_0 and x_1 until they coalesce at time 0, since all the intermediary paths will be between these two extreme cases.

Population Monte Carlo

As an alternative to MCMC, Cappé et al. (2003) have shown that the importance sampling technique (Robert and Casella 2004, Chapter 3) can be generalised to encompass much more adaptive and local schemes than thought previously, without relaxing its essential justification of providing a correct discrete approximation to the distribution of interest. This leads to the Population Monte Carlo (PMC) algorithm, following Iba's (2000) denomination. The essence of the PMC scheme is to learn from experience, that is, to build an importance sampling function based on the performances of earlier importance sampling proposals. By introducing a temporal dimension to the selection of the importance function, an adaptive perspective can be achieved at little cost, for a potentially large gain in efficiency. Celeux et al. (2003) have shown that the PMC scheme is a viable alternative to MCMC schemes in missing data settings, among others for the stochastic volatility model (Shephard 1996). Even with the standard choice of the full conditional distributions, this method provides an accurate representation of the distribution of interest in a few iterations. In the same way, Guillin et al. (2003) have illustrated the good properties of this scheme on a switching ARMA model (Hamilton 1988) for which the MCMC approximations are less satisfactory.

General PMC (sequential setups)

0. **Initialization.** Choose $\underline{\theta}_{(0)}^{(1)}, \ldots, \underline{\theta}_{(0)}^{(M)}$ and $\underline{p}_{(0)}^{(1)}, \ldots, \underline{p}_{(0)}^{(M)}$

1. **Step t.** For $t = 1, \ldots, T$

   1.1 For $i = 1, \ldots, M$

      1.1.1 Generate $\left(\underline{\theta}_{(t)}^{(i)}, \underline{p}_{(t)}^{(i)}\right)$ from $q_{it}(\theta, p)$,

      1.1.2 Compute

$$\rho^{(i)} = \frac{f\left(x|\underline{\theta}_{(t)}^{(i)}, \underline{p}_{(t)}^{(i)}\right) \pi\left(\underline{\theta}_{(t)}^{(i)}, \underline{p}_{(t)}^{(i)}\right)}{q_{it}\left(\underline{\theta}_{(t)}^{(i)}, \underline{p}_{(t)}^{(i)}\right)},$$

   1.2 Compute $\omega^{(i)} = \rho^{(i)} \Big/ \sum_{l=1}^{M} \rho^{(l)}$,

   1.3 Resample M values with replacement from the $\left(\underline{\theta}_{(t)}^{(i)}, \underline{p}_{(t)}^{(i)}\right)$'s using the weights $\omega^{(i)}$

PMC and normal mixtures

In the case of the normal mixture (1.7), a PMC sampler can be efficiently implemented *without* the (Gibbs) augmentation step, using normal random walk proposals based on the previous sample of (μ_1, μ_2)'s. Moreover, the difficulty inherent to random walks, namely the selection of a "proper" scale, can be bypassed by the adaptivity of the PMC algorithm. Indeed, several proposals can be associated with a range of variances v_k, $k = 1, \ldots, K$. At each step of the algorithm, new variances can be selected proportionally to the performances of the scales v_k on the previous iterations. For instance, a scale can be chosen proportionally to its *non-degeneracy rate* in the previous iteration, that is, the percentage of points generated with the scale v_k that survived after resampling. When the survival rate is null, in order to avoid the complete removal of a given scale v_k, the corresponding number r_k of proposals with that scale is set to a positive value, like 1% of the sample size.

PMC Normal mixture algorithm

0. **Initialization.** Choose $(\mu_1)_{(0)}^{(1)}, \ldots, (\mu_1)_{(0)}^{(M)}$ and $(\mu_2)_{(0)}^{(1)}, \ldots, (\mu_2)_{(0)}^{(M)}$

1. **Step t.** For $t = 1, \ldots, T$

   1.1 For $i = 1, \ldots, M$

      1.1.1 Generate k from $\mathcal{M}(1; r_1, \ldots, r_K)$,

      1.1.2 Generate $(\mu_j)_{(t)}^{(i)}$ $(j = 1, 2)$ from $\mathcal{N}\left((\mu_j)_{(t-1)}^{(i)}, v_k\right)$

      1.1.4 Compute

$$\rho^{(i)} = \frac{f\left(x|(\mu_1)_{(t)}^{(i)}, (\mu_2)_{(t)}^{(i)}\right)\pi\left((\mu_1)_{(t)}^{(i)}, (\mu_2)_{(t)}^{(i)}\right)}{\sum_{l=1}^{K}\prod_{j=1}^{2}\varphi\left((\mu_j)_{(t)}^{(i)}; (\mu_1)_{(t-1)}^{(i)}, v_l\right)},$$

   1.2 Compute $\omega^{(i)} = \rho^{(i)} \Big/ \sum_{l=1}^{M} \rho^{(l)}$,

   1.3 Resample the $(\mu_1)_{(t)}^{(i)}, (\mu_2)_{(t)}^{(i)}$'s using the weights $\omega^{(i)}$

   1.4 Update the r_l's: r_l is proportional to the number of $(\mu_1)_{(t)}^{(i)}, (\mu_2)_{(t)}^{(i)}$'s with variance v_l resampled.

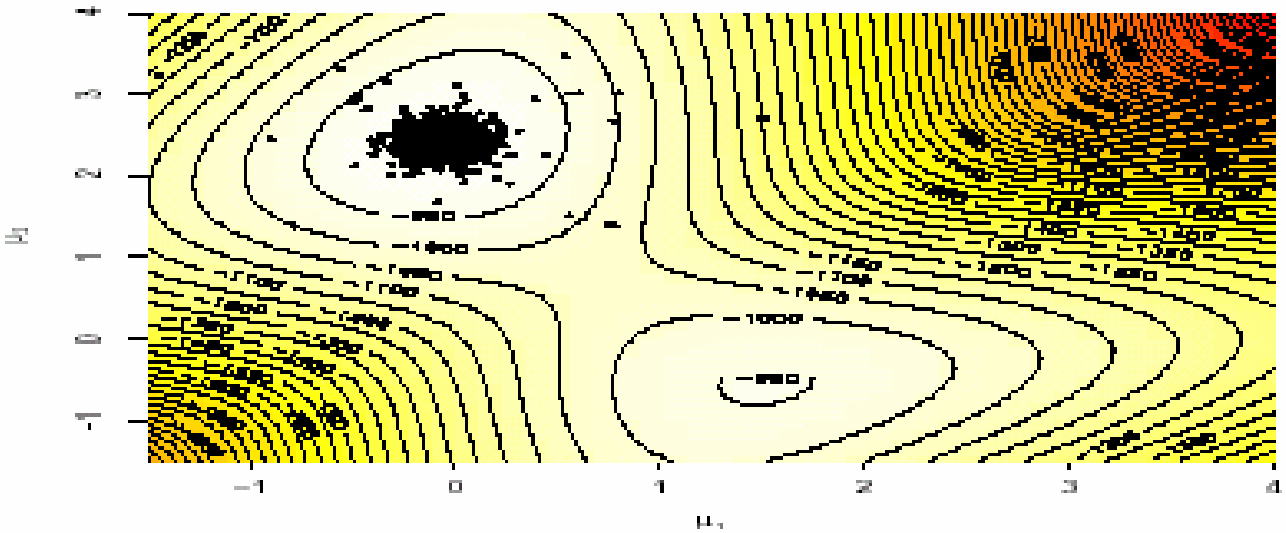


FIGURE 20. Representation of the log-posterior distribution with the PMC weighted sample after 10 iterations (the weights are proportional to the circles at each point).

Population Monte Carlo

- System of particles $(\theta_1^{(t)}, \ldots, \theta_M^{(t)})_t$ random vector evolving over time t
- Many different types, eg Particle Filter
- Vector of weights $(w_1^{(t)}, \ldots, w_M^{(t)})$
- Can approximate integrals, eg $\int h(\theta) \pi(d\theta)$ through importance sampling approximations $\Sigma_{k=1:M} w_k^{(t)} h(\theta_k^{(t)})$

Widely popular!

- **Used in engineering, computing, robotics, …**
- **Usually used in sequential settings**
  eg tracking a moving target (Doucet et al, 2001)
- **Extended to static settings with large datasets (Chopin 2000, Berzuini and Gilks 2001)**
- **Body of literature: Fearnhead, Carpenter et al, Chen, Crisan & Doucet, Godsill et al, West, Pitt & Shephard, Liu,…**

Pinball as a Particle Filter

- Use neither importance sampling schemes nor weights
- Resample whole vector at each iteration
- Use an updating system based on the standard random walk
- Avoid importance sampling justification
- Fixed number of particles (although could branch)

Pinball as adaptive MCMC

- **Simulates simultaneously a set of values in a dependent manner (Gilks & Roberts 1997)**

- **Akin to Haario & Sacksman (2001) and Andrieu &Robert (2001) but chain is homogeneous so standard ergodic theorems apply**
- **Akin to parallel MCMC, for example to assess convergence (Gelman & Rubin 1992)**

- **Akin to coupled MCMC: moves depend on the other chains**

Algorithm 1

1. Construct a grid $(\theta_1^{(0)},\ldots, \theta_M^{(0)})$ of starting values over the support of π.

2. For $t=0,..,T$
   for $k=1,..,M$ simulate
   $\theta_k^{(t+1)} \sim K_k(\theta \mid \theta_1^{(t+1)},..., \theta_{k-1}^{(t+1)}, \theta_k^{(t)}, \theta_M^{(t)})$

- $K_k(\bullet \mid \theta_1,..., \theta_M)$ are proposals with stationary distribution π that satisfy detailed balance
- If support of π is unbounded, construct grid from starting dist'n of μ or reparametrise

Theorem

By conditional balance, the stationary distribution associated with Algorithm 1 is

$(\theta_1,..,\theta_M) \sim \pi(\theta_1)\times\ldots\times\pi(\theta_M) = \pi^M(\theta_1,\ldots,\theta_M)$

So What?

- **Under standard irreducibility conditions, the Markov chain $(\theta_1,\ldots,\theta_M)$ is ergodic and positive recurrent with the correct stationary distribution.**
- **After removing influence of starting values, the particle system $(\theta_1^{(t)},\ldots,\theta_M^{(t)})$ is an iid sample from π at any given time *t*, (rather than in the long run as in regular MCMC sampling).**
- **Can then evaluate output as in regular Monte Carlo experiments, eg normal approximation confidence intervals.**

On with the Pinball

Want a proposal that speeds up mixing:

1. **'Pseudo-reference' distribution π^R that pushes particles further apart from one another**
2. **Metropolis move based on π^R**
3. **Increase efficiency of proposal with a (deterministic) delayed rejection mechanism**
4. **Use a final Metropolis move to calibrate to the true reference distribution π**

1. Repulsive Proposal

"Pseudo-reference" distribution

tempering

repulsion

$$\pi_k^R(\theta) \propto \prod_{j \neq k} exp(-\xi / \pi(\theta_j) \, ||\theta - \theta_j||^2)$$

moderator

No dependence on normalisation constant of π, can absorb it into ξ

2. Metropolis Move

Update $\theta_k^{(t)}$:

- **Propose $\theta^*_k{}^{(t)}$ using**

  $K^*_k(\theta_k \mid \theta_1^{(t+1)}, \ldots, \theta_{k-1}^{(t+1)}, \theta_k^{(t)}, \ldots, \theta_M^{(t)})$

  based on $\pi_k{}^R$

- **Accept with probability**

  $1 \wedge \pi(\theta^*_k{}^{(t)})\ \pi_k{}^R(\theta_k^{(t)}) / (\pi(\theta_k^{(t)})\ \pi_k{}^R(\theta^*_k{}^{(t)}))$

Before getting to 3… Delayed rejection

Tierney and Mira (1999); Green and Mira....

A. Propose move for θ_k.

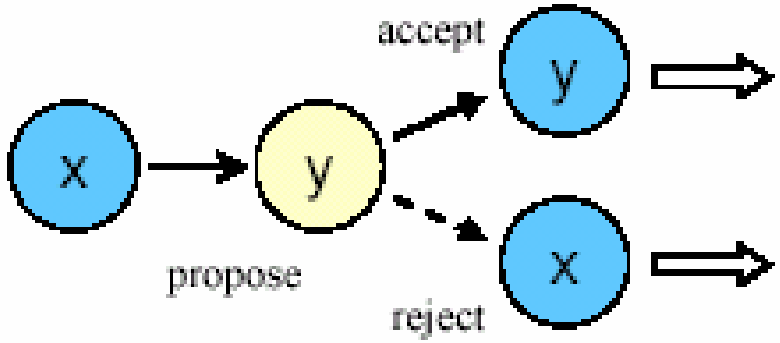
B. Accept with usual M-H probability.

C. If reject, propose new move for θ_k and accept with probability that takes into account the fact that the first move is rejected.

D. If reject, repeat C as required or until stopping rule.

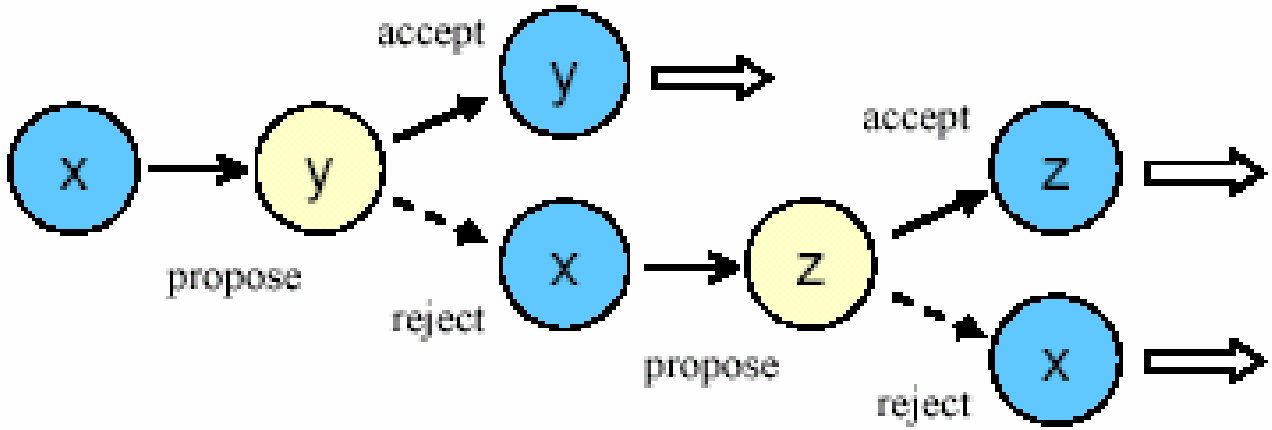
So…

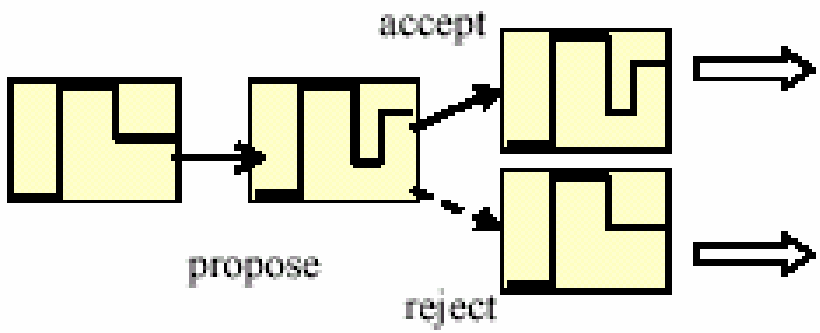
- New moves can depend on previous (rejected) proposed values.
- Moves can also depend on the other particles.
- Want detailed balance, so need reversibility, so use a random walk + iterated (deterministic) reflections if rejected.


accept
y
x
y
propose
x
reject

M-H

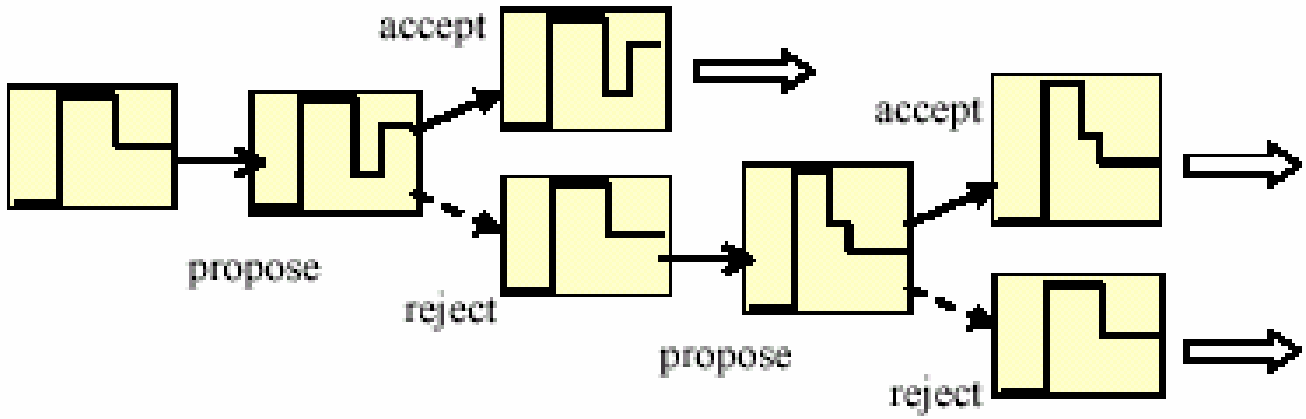
DRA


accept
y
x
y
propose
x
reject
accept
z
z
propose
x
reject


accept
propose
reject

M-H

DRA


accept
propose
reject
accept
propose
reject

Delayed rejection and the pinball

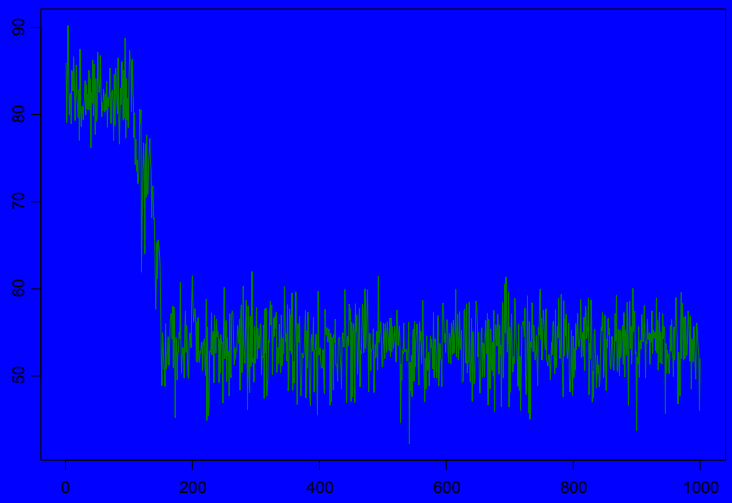
Given $(\theta_1^{(t+1)},..,\theta_k^{(t)},\ldots,\theta_M^{(t)})$:

A. Generate 1st proposal θ_{k0} from symmetric distribution based around current value $\theta_k^{(t)}$
B. Accept with M-H probability based on $\pi_k^R(\theta_k)$ *and* $\pi_k^R(\theta_{k0})$

C. If reject, 'bounce' θ_{k0} away from nearest particle θ_j to θ_{k0} through symmetry w.r.t. (θ_j, θ_{k0}) and θ_{k1}.

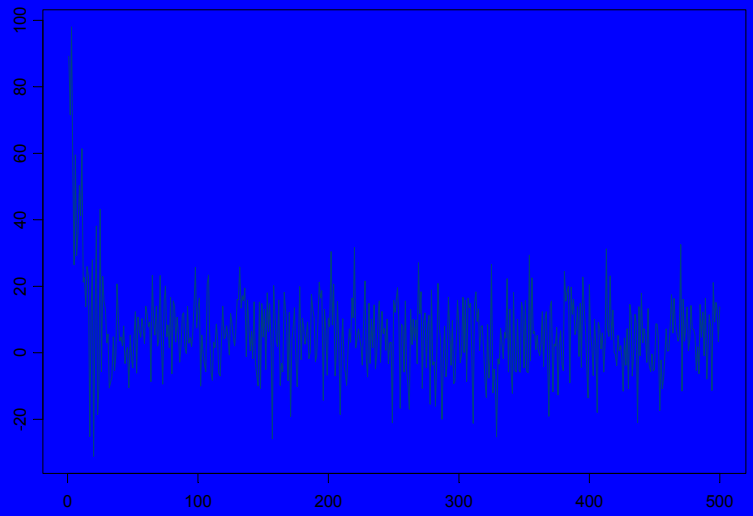
When do we stop?

- **Erase influence of starting values**
- **Check for i.i.d.**



How do we compare samplers?

- **By effective sample size (Mira, …)**
- **By estimation**



Sequential Methods

Recent interest has also focused on MCMC methods for particle ¯lters which are usually implemented in sequential settings or for processing and analysis of large datasets. A particle ¯lter describes a dynamic state-space model of a process with an underlying state of interest that evolves over time. The posterior distribution of the state is approximated by a set of weighted particles, with the weight of a particle inversely proportional to its probability mass. Numerous algorithms for updating the particles and their weights over time have been proposed. Most of these enjoy rigorous convergence properties (Crisan and Doucet, [25]) and under certain conditions can claim a Central Limit Theorem [27].

Bayesians in the Night (Strangers in the Night)

Bayesians in the night
with exchangeable glances
Assessing in the night
the prior chances
We'd be sharing risks
before the night is through.

Something in your prior
was so exciting
Something in your data
was so inviting
Something in your posterior
told me I must have you.