Nearly Perfect Sampling

Duncan Murdoch

Department of Statistical and Actuarial Sciences
University of Western Ontario

March 22, 2012

Outline

1. Perfect Sampling
    - Background
    - Practical issues
    - Fill's Rejection Sampler
2. Nearly Perfect Sampling
    - Motivation
    - Rat Data Example
    - Seed Data Example
3. Conclusions

MCMC

- We want to study the distribution of $X \sim \pi(\cdot)$, $X \in \mathcal{S}$.
- If we could simulate i.i.d. values $X_i \sim \pi(\cdot)$, $i = 1, \ldots, n$, then we could approximate quantities like $E(f(X))$ by $(1/n) \sum_{i=1}^{n} f(X_i)$.
- We don't know how to simulate from $\pi(\cdot)$ directly, but we do know how to sample from a Markov chain X_t, whose steady-state distribution is $\pi(\cdot)$.

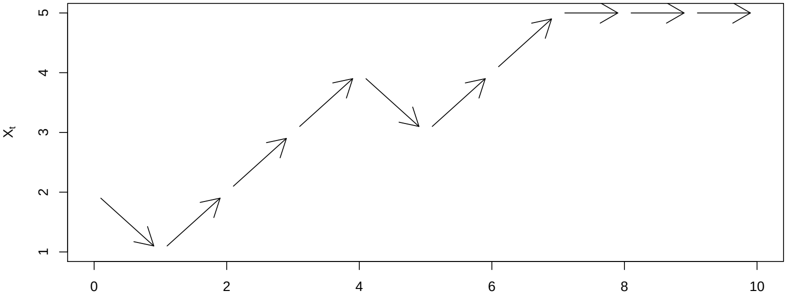
My interest is in $\mathcal{S} = R^n$ for n of reasonable size, with $\pi(\cdot)$ being a posterior. The examples have $n = 65$ and 26.

Coupling from the past (CFTP)

- Problem: The distribution of the X_t values converges to $\pi(\cdot)$, and averages converge to expectation w.r.t. $\pi(\cdot)$, but how quickly?
- Idea (Propp and Wilson, 1996): Compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. If they all give the same answer, it must be in steady-state!
- Write our Markov chain as $X_{t+1} = \phi(X_t, U_{t+1})$, where U_t is an i.i.d. sequence from some distribution, and $\phi(\cdot, \cdot)$ is a fixed function.
- Think of $\phi(\cdot, \cdot)$ as the computer program used to write a simulation of the Markov chain. U_t is the output of the computer's pseudo-random number generator used to update the state.

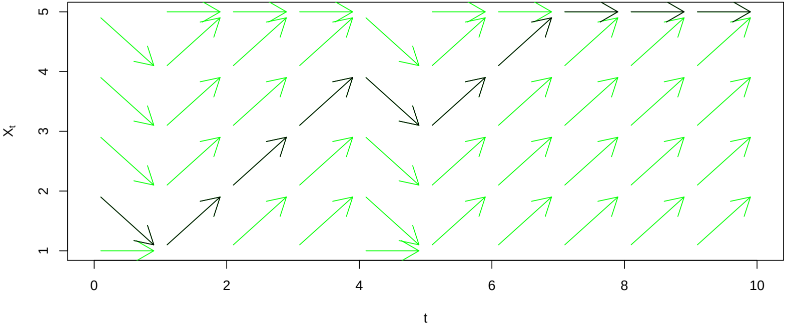
Example: Random walk on $1, \ldots, 5$

$$
\begin{aligned}
X_{t+1} &= \phi(X_t, U_{t+1}) \\
\phi(x, u) &= \min[\max(x+u, 1), 5] \\
U_t &= \pm 1 \text{ (with equal probability)}
\end{aligned}
$$



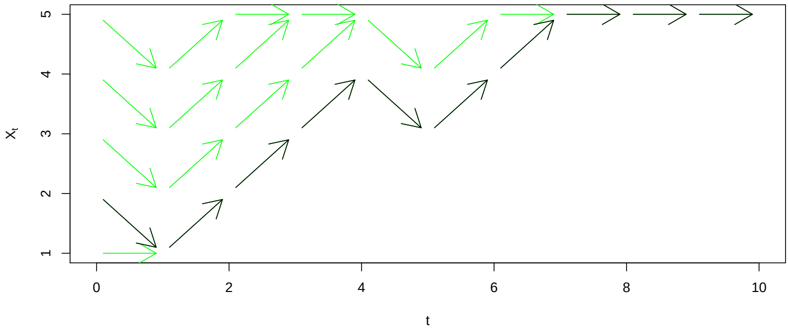
Coupling

Using $\phi(\cdot,\cdot)$ lets us imagine paths that were not sampled. Fix U_t and apply $\phi(x, U_{t+1})$ to all of $\mathcal{S}$.



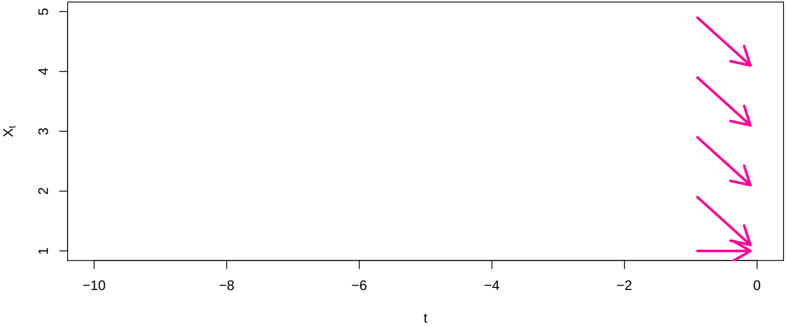
Coupling continued. . .

Paths may *coalesce*: regardless of the initial state, the value of X_t is the same for large enough t. The past is forgotten; no initialization bias remains.



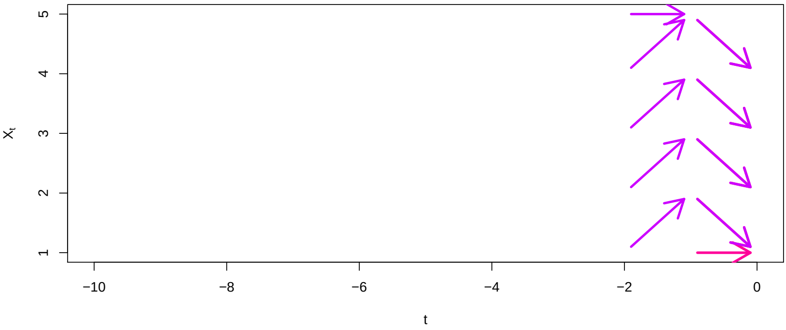
Coupling from the past (CFTP)

To avoid a coalescence time bias, fix the observation time *before* testing for coalescence: compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. WLOG, observe at time $t = 0$.



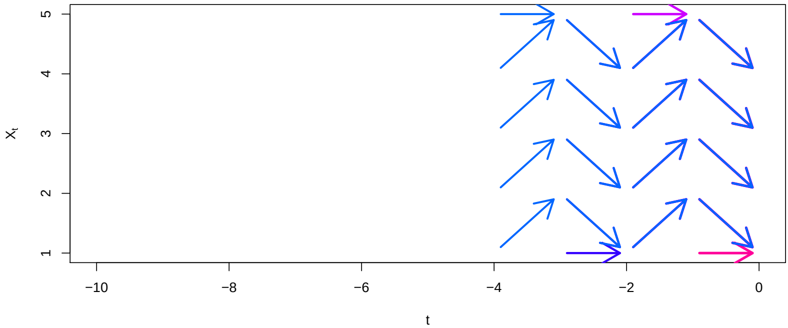
Coupling from the past (CFTP)

To avoid a coalescence time bias, fix the observation time *before* testing for coalescence: compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. WLOG, observe at time $t = 0$.



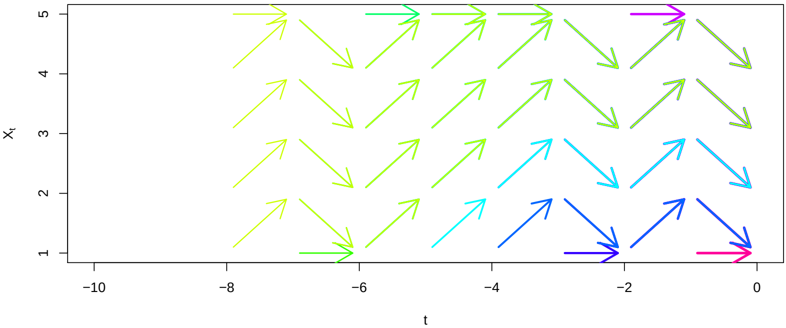
Coupling from the past (CFTP)

To avoid a coalescence time bias, fix the observation time *before* testing for coalescence: compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. WLOG, observe at time $t = 0$.



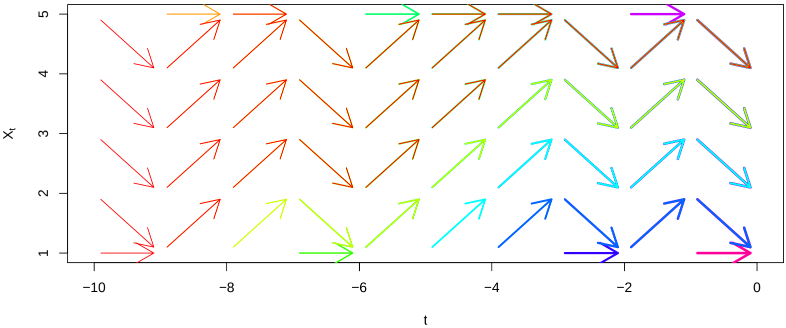
Coupling from the past (CFTP)

To avoid a coalescence time bias, fix the observation time *before* testing for coalescence: compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. WLOG, observe at time $t = 0$.



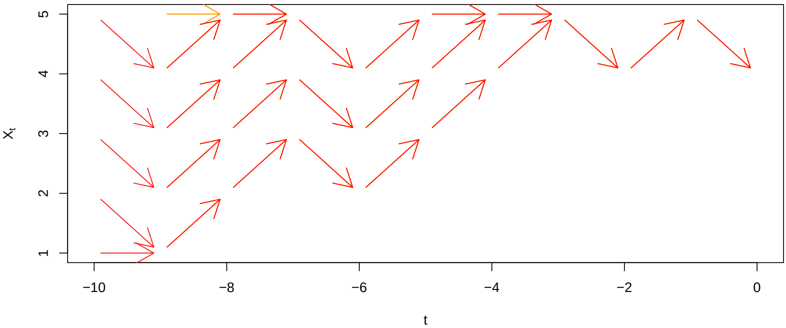
Coupling from the past (CFTP)

To avoid a coalescence time bias, fix the observation time *before* testing for coalescence: compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. WLOG, observe at time $t = 0$.



Coupling from the past (CFTP)

To avoid a coalescence time bias, fix the observation time *before* testing for coalescence: compute the result of an infinitely long run from the past by coupling all possible tails of shorter runs. WLOG, observe at time $t = 0$.



What is involved in doing CFTP?

- Either $\pi(\cdot)$ or the Markov chain is given.
- Whether or not the Markov chain is given, we have some flexibility in its specification.
- The coupling is up to us.
- Detecting coalescence is up to us.

Choosing a coupling

- Need to write $X_{t+1} = \phi(X_t, U_{t+1})$, with U_t i.i.d.
- Want coalescence. This can be tricky when the state space $\mathcal{S}$ is large (e.g. R^n).
- Want easy coalescence detection.

Random walk Metropolis

- Random walk Metropolis is a very simple MCMC sampler; it should be easy to find a coupling.
- A naive choice is to use common random inputs: Given $X_t = x$ we calculate a proposal $Y = x + Z$, where Z is a draw from a symmetric distribution (e.g. $N(0,1)$). We also draw $U \sim \text{Unif}(0,1)$. Then

$$X_{t+1} = \begin{cases} Y & \text{if } U < \pi(Y)/\pi(x) \\ x & \text{otherwise} \end{cases}$$

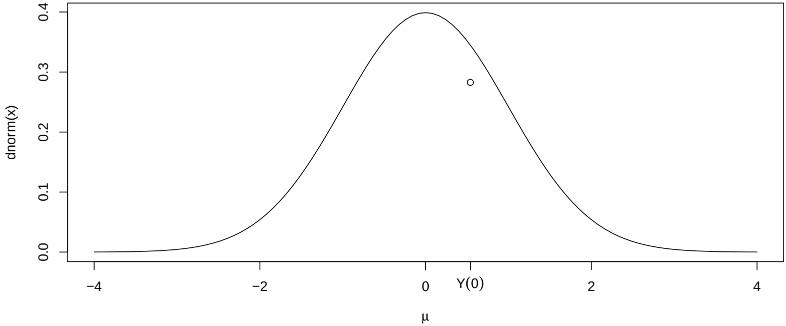
The naive coupling fails!

If $\mathcal{S}$ contains an interval, then this sampler won't coalesce, and CFTP will fail every time:

- If states x_1 and $x_2 \neq x_1$ both accept the proposal, their difference remains.
- If both reject, their difference remains.
- The only ways to coalesce are for $Z = x_2 - x_1$ and state x_1 accepts while x_2 rejects: probability zero events when $\mathcal{S}$ is continuous.

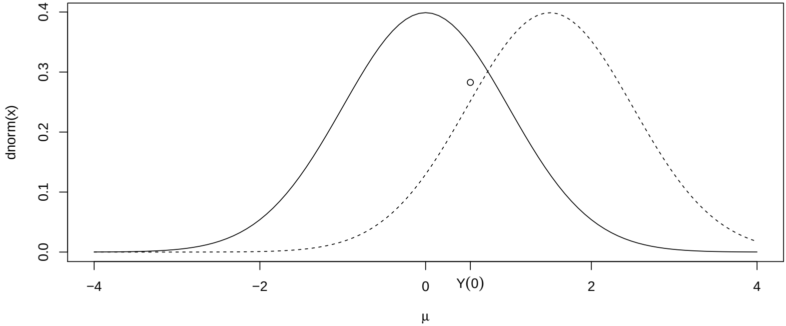
Wilson's Multishift Coupler

Wilson (2000a) invented a very clever coupler for location families which we can use for the proposals in Metropolis. For example, with $X_t = x$ we want $Y(x) \sim N(x, 1)$, and we want coalescence for different values of x.



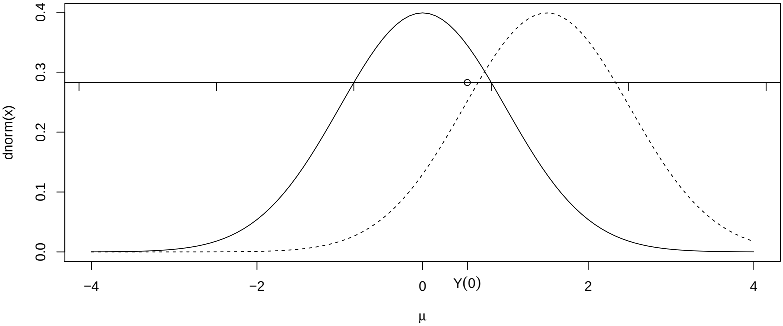
Wilson's Multishift Coupler

Wilson (2000a) invented a very clever coupler for location families which we can use for the proposals in Metropolis. For example, with $X_t = x$ we want $Y(x) \sim N(x, 1)$, and we want coalescence for different values of x.



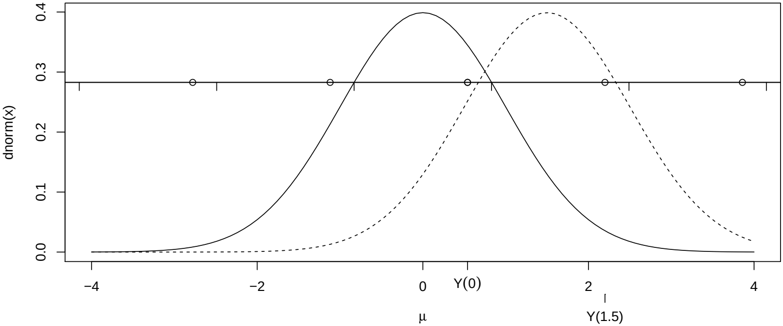
Wilson's Multishift Coupler

Wilson (2000a) invented a very clever coupler for location families which we can use for the proposals in Metropolis. For example, with $X_t = x$ we want $Y(x) \sim N(x, 1)$, and we want coalescence for different values of x.



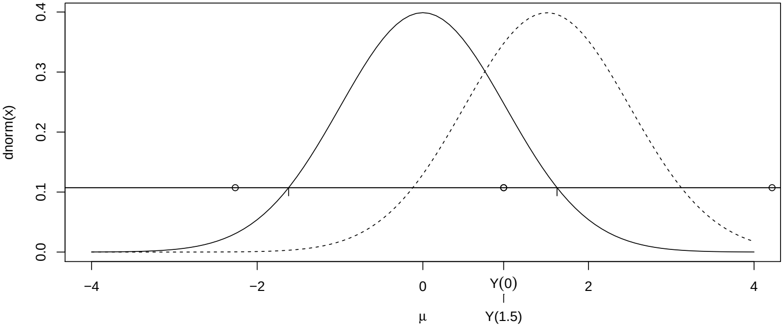
Wilson's Multishift Coupler

Wilson (2000a) invented a very clever coupler for location families which we can use for the proposals in Metropolis. For example, with $X_t = x$ we want $Y(x) \sim N(x, 1)$, and we want coalescence for different values of x.

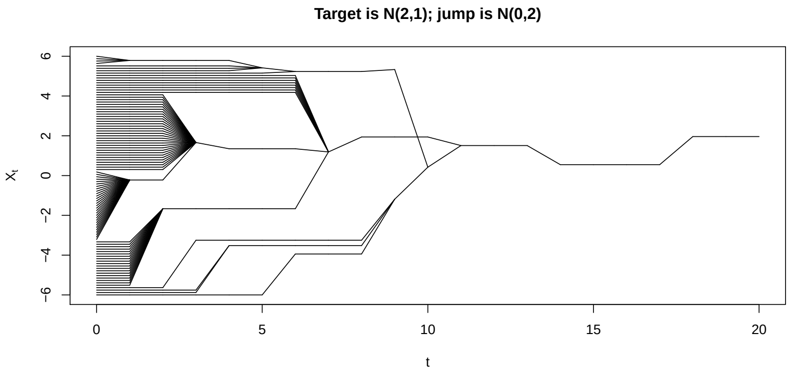


Wilson's Multishift Coupler

Wilson (2000a) invented a very clever coupler for location families which we can use for the proposals in Metropolis. For example, with $X_t = x$ we want $Y(x) \sim N(x, 1)$, and we want coalescence for different values of x.



Coupled Metropolis Sampler



- Not quite monotone in x.
- Works for bounded sets, but not unbounded ones.

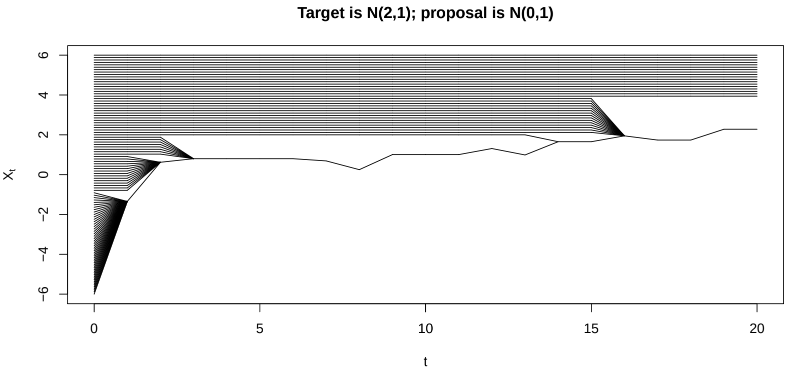
Mixing with an "independence sampler"

In an independence sampler Y is drawn from a fixed distribution $p(\cdot)$, independent of X_t.
We accept Y when

$$U < \frac{\pi(Y)/p(Y)}{\pi(X_t)/p(X_t)}$$

If we choose $p(\cdot)$ with heavier tails than $\pi(\cdot)$, the ratio is large when X_t is sufficiently far out in the tails: the set of possible X_t values is reduced to a compact set.

Coupled Independence Sampler



Independence samplers are usually not very good for MCMC, but can be used in combination with a better sampler.

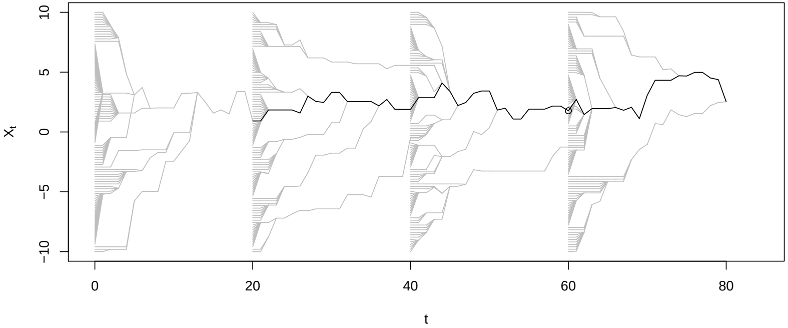
Why doesn't everybody use CFTP?

Unless the problem has a very nice structure, proving that all paths coalesce is hard.

- If updates maintain ordering of points, we can track just minimal and maximal points—but this is uncommon.
- Doing the bookkeeping to track coalescence is hard without that.
- In high dimensions, things are worse: tracking is very difficult, and coalescence is very slow.

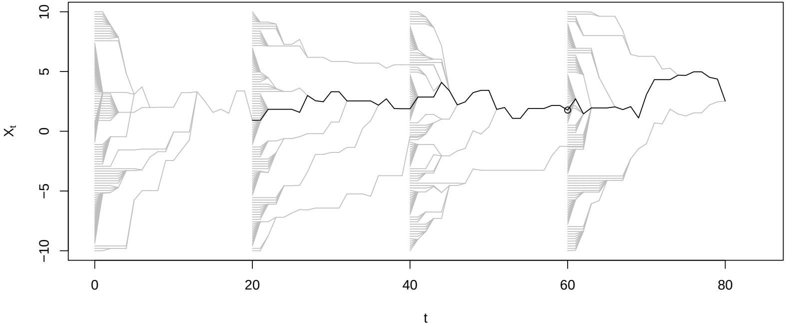
Implementation

- CFTP is tricky because of the backwards search.
- Wilson (2000b) described “read-once CFTP”.



Implementation

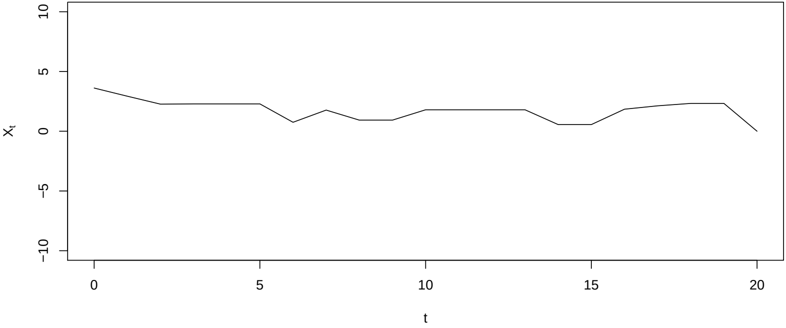
- CFTP is tricky because of the backwards search.
- Wilson (2000b) described “read-once CFTP”.



- This is related to Fill's rejection sampler (Fill et al., 2000).

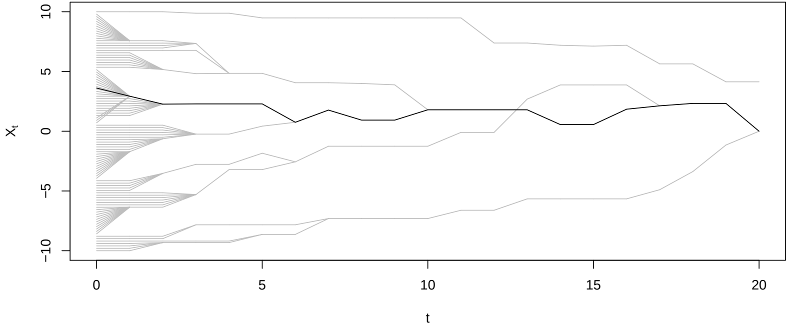
Fill's Rejection Sampler

Run the π-reversal of the chain backwards from time T to 0,



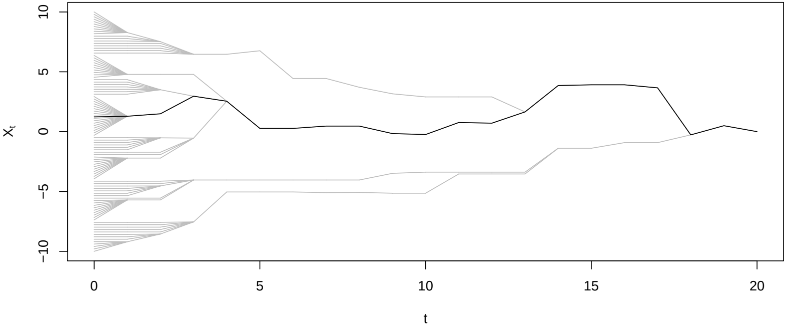
Fill's Rejection Sampler

Run the π-reversal of the chain backwards from time T to 0, then run coupled chains forward. If they coalesce, output the time 0 value of the first chain.



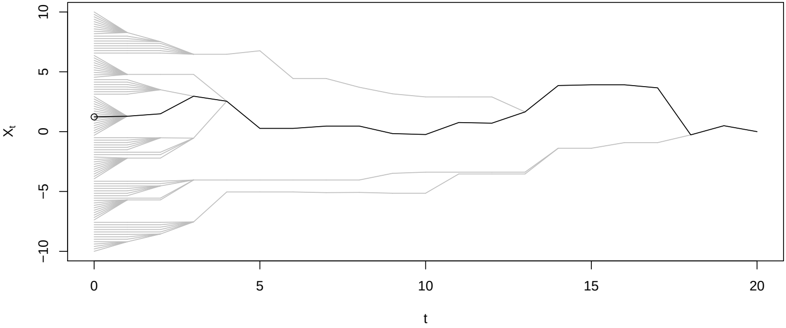
Fill's Rejection Sampler

Run the π-reversal of the chain backwards from time T to 0, then run coupled chains forward. If they coalesce, output the time 0 value of the first chain.



Fill's Rejection Sampler

Run the π-reversal of the chain backwards from time T to 0, then run coupled chains forward. If they coalesce, output the time 0 value of the first chain.



Fill is difficult to implement

- In most situations, it is difficult to couple the forward chain to the reversed path.
- With both Metropolis and independence samplers these are easy to do.
- Coalescence detection is still hard...

Implementing Fill for the independence sampler

1. Set X_T to an arbitrary value with $\pi(X_T) > 0$.
2. The sampler is reversible, so use it to generate $X_{T-1}, \ldots, X_0$.
3. To couple the forward paths: if $X_{t+1} = X_t$, generate Y and U values until we get a rejection. If $X_{t+1} \neq X_t$, set $Y = X_{t+1}$, generate U on the range that indicates acceptance. Use the same (Y, U) for all other states.

Metropolis is very similar even if we are using Wilson's multishift coupler for the proposals.

Motivation for Nearly Perfect Sampling

Perfect sampling is too hard to use in practice, but maybe we don't need to be perfect.

- Is it good enough to choose a finite set of starting points, and check for coalescence of those?
- Johnson (1996) suggested this as a convergence diagnostic, but he was working before CFTP.
- What couplers can we use for this?

My First Guesses

- It is easy to couple Metropolis and independence samplers, and they are all we need.
- We'll want to put them in a particular order: independence first to force bounded support, then Metropolis to get coalescence in the centre.
- Fill's sampler may be usable: both Metropolis and independence are reversible, and it is easy to find the compatible forward coupler.

A Real Example

- WinBUGS includes an example of a normal growth curve model with 65 parameters: separate intercept and slope for each of 30 rats, plus 5 common parameters: mean and variance of the intercepts and slopes, variance of the observations.
- It is not hard to write R code to evaluate the log posterior (leaving out the normalizing constant) for the full model; that's all we need for Metropolis and Independence sampling.

Rats: a normal hierarchical model

This example is taken from section 6 of Gelfand *et al* (1990), and concerns 30 young rats whose weights were measured weekly for five weeks. Part of the data is shown below, where Y_{ij} is the weight of the ith rat measured at age x_j.

Weights Y_{ij} of rat i on day x_j

	$x_j = 8$	15	22	29	36
Rat 1	151	199	246	283	320
Rat 2	145	199	249	293	354
....					
Rat 30	153	200	244	286	324

A plot of the 30 growth curves suggests some evidence of downward curvature.

The model is essentially a random effects linear growth curve

$$Y_{ij} \sim \text{Normal}(\alpha_i + \beta_i(x_j - x_{bar}), \tau_c)$$

$$\alpha_i \sim \text{Normal}(\alpha_c, \tau_\alpha)$$

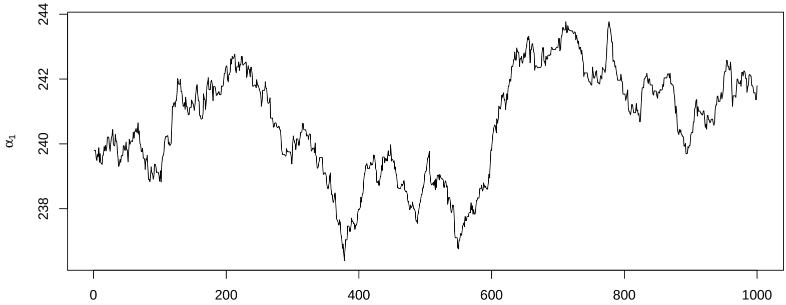
$$\beta_i \sim \text{Normal}(\beta_c, \tau_\beta)$$

where $x_{bar} = 22$.

Preliminaries

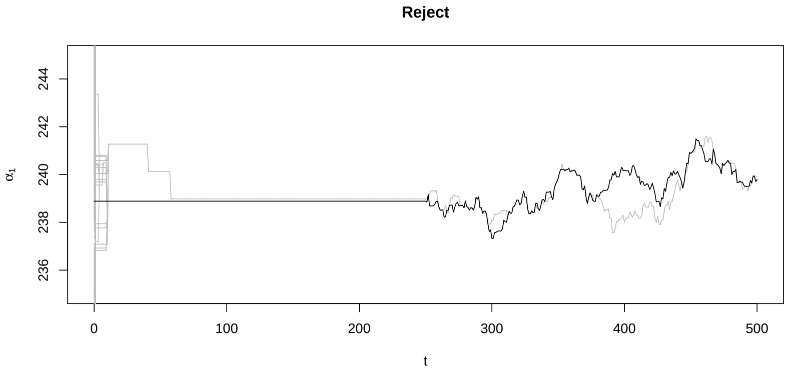
First, try a single path.

- Choose X_0 by separate linear regressions.
- Choose the scale for the jumps of Metropolis by a simple Metropolis run.
- Also choose the proposal for the independence sampler.



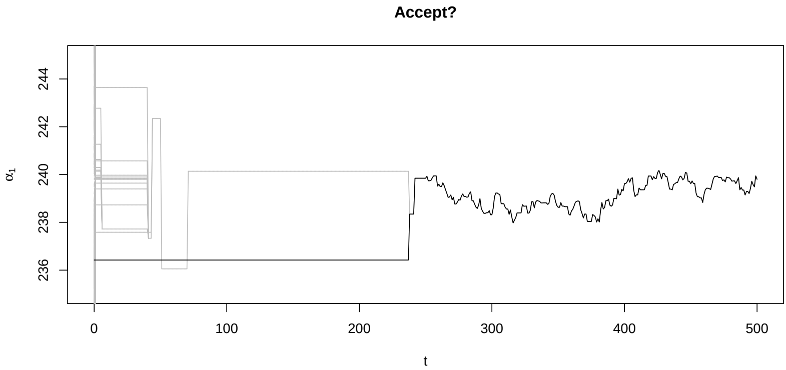
Run coupled paths

- Choose multiple starting values around X_0.
- Run Fill's algorithm with independence and Metropolis samplers.



Run coupled paths

- Choose multiple starting values around X_0.
- Run Fill's algorithm with independence and Metropolis samplers.



Did it work?

Did it work?

There are reasons to doubt that it actually worked.

- There are often (as in the first simulation) points which reject *all* independence proposals, because it is very hard to get close enough to the target density.
- The Metropolis updates rarely coalesce in high dimensions. To get coalescence of all components of two paths, we need 65 independent events to occur: this is rare unless the probabilities are very high.

Did it work?

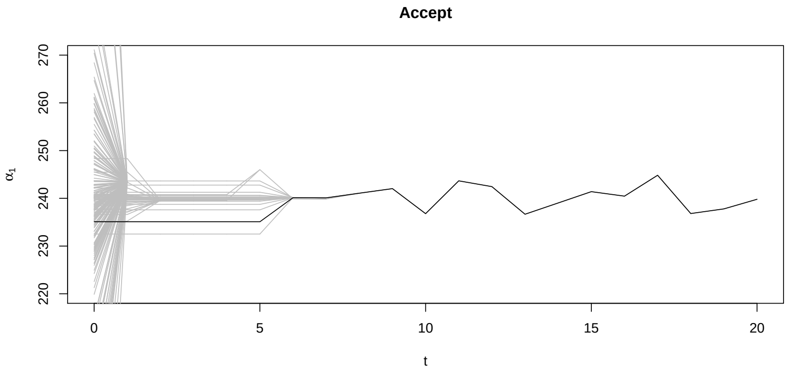
There are reasons to doubt that it actually worked.

- There are often (as in the first simulation) points which reject *all* independence proposals, because it is very hard to get close enough to the target density.
- The Metropolis updates rarely coalesce in high dimensions. To get coalescence of all components of two paths, we need 65 independent events to occur: this is rare unless the probabilities are very high.
- Gibbs sampler updates converge better than either independence or Metropolis updates, but those are harder to do automatically (though BUGS does).

Gibbs updates

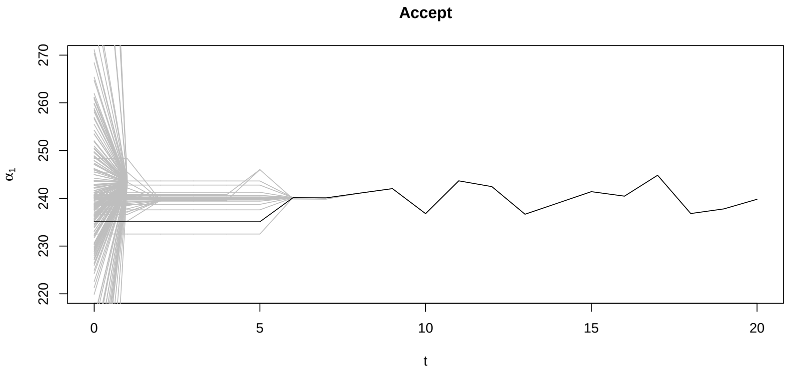
- Two component Gibbs: in the posterior, regression parameters (α_i, β_i) are independent bivariate normal given the others. Couple updates in the naive way.
- In a two component Gibbs sampler, only one component needs to coalesce to make the whole chain coalesce.
- Use Metropolis for the hyperparameters, but start with some independence samples to handle the tails.

It works!



- Using the GIbbs sampler effectively reduces the dimension from 65 to 5.

It works!



- Using the GIbbs sampler effectively reduces the dimension from 65 to 5.
- This problem is too easy...

Seeds: Random effect logistic regression

This example is taken from Table 3 of Crowder (1978), and concerns the proportion of seeds that germinated on each of 21 plates arranged according to a 2 by 2 factorial layout by seed and type of root extract. The data are shown below, where r_i and n_i are the number of germinated and the total number of seeds on the *i* th plate, *i* =1,...,N. These data are also analysed by, for example, Breslow: and Clayton (1993).

seed O. aegyptiaco 75						*seed O. aegyptiaco 73*					
Bean			Cucumber			Bean			Cucumber		
r	n	r/n	r	n	r/n	r	n	r/n	r	n	r/n
10	39	0.26	5	6	0.83	8	16	0.50	3	12	0.25
23	62	0.37	53	74	0.72	10	30	0.33	22	41	0.54
23	81	0.28	55	72	0.76	8	28	0.29	15	30	0.50
26	51	0.51	32	51	0.63	23	45	0.51	32	51	0.63
17	39	0.44	46	79	0.58	0	4	0.00	3	7	0.43
			10	13	0.77						

The model is essentially a random effects logistic, allowing for over-dispersion. If p_i is the probability of germination on the *i* th plate, we assume

$$r_i \sim \text{Binomial}(p_i, n_i)$$

$$\text{logit}(p_i) = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_{12} x_{1i} x_{2i} + b_i$$

$$b_i \sim \text{Normal}(0, \tau)$$

Random Effects Logistic Regression

- A hierarchical model with 26 parameters
- No conjugate prior structure, so Gibbs sampling is harder.
- Using Metropolis within Gibbs works, but fails to coalesce.

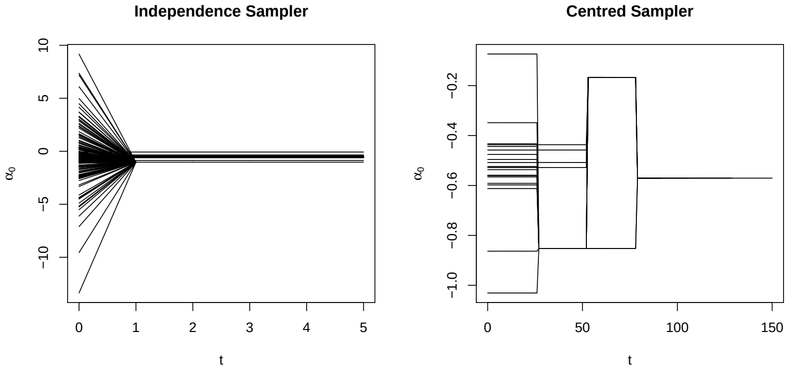
Random Effects Logistic Regression

- A hierarchical model with 26 parameters
- No conjugate prior structure, so Gibbs sampling is harder.
- Using Metropolis within Gibbs works, but fails to coalesce.
- A Metropolis-Hastings sampler that uses

$$Y_{t+1}|X_t \sim N(\rho X_t + (1-\rho)\widehat{X}, \sigma^2)$$

  works well, i.e. a Normal proposal centred on the line between X_t and the posterior mode.
- Doing updates one component at a time seems to be fastest.

Seed example continued...



Questions

- How far is finite coalescence from complete coalescence? Can we tell this in practice?

Questions

- How far is finite coalescence from complete coalescence? Can we tell this in practice?
- Are there better couplers for Metropolis-Hastings, or for other Markov chains?

Questions

- How far is finite coalescence from complete coalescence? Can we tell this in practice?
- Are there better couplers for Metropolis-Hastings, or for other Markov chains?
- Can we learn something about tuning single path MCMC by tuning couplers?

The `coupling` package

- I wrote a package in R to implement these algorithms, to explore the "nearly perfect" idea.
- The package is not ready for release yet, but it includes:

  Sampling algorithms CFTP, ROCFTP and Fill's algorithms, using finite sets of starting points.

  Distributions Various proposal distributions for independence and Metropolis samplers.

  Utilities Utility functions for plotting, detecting coalescence, etc.

  Examples Full code for worked examples: Rat example, Seed example.
- This talk was written using code from the `coupling` package.

References

Fill, J. A., Machida, M., Murdoch, D. J., and Rosenthal, J. S. (2000). Extension of Fill's perfect rejection sampling algorithm to general chains. *Random Structures and Algorithms*, 17:290–316.

Johnson, V. E. (1996). Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *Journal of the Americal Statistical Association*, 91:154–166.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252.

Wilson, D. (2000a). Layered multishift coupling for use in perfect sampling algorithms (with a primer on CFTP). In Madras, N., editor, *Monte Carlo Methods—Fields Institute Communications Vol. 26*. AMS.

Wilson, D. B. (2000b). How to couple from the past using a read-once source of randomness. *Random Structures and Algorithms*, 16:85–113.