Perfect simulation for image analysis

Mark Huber
Fletcher Jones Foundation Associate Professor of Mathematics and Statistics and George R. Roberts Fellow
Mathematical Sciences
Claremont McKenna College

A prior for pictures

An image is a (rectangular) grid of pixels

- Neighboring pixels often alike
- A simple approach is the Ising model

In Ising model each pixel is either 0 (black) or 1 (white).

$$p(x) \propto \prod_{\text{edges } \{i,j\}} \exp(-\beta(x(i) - x(j))^2)$$

Ising model

Parameter β called inverse temperature

- When $\beta = 0$, pixels uniform and independent
- When $\beta = \infty$, all pixels same color

$\beta = 0$

Better pictures use grayscale

Instead of $\{0, 1\}$, use $[0, 1]$ to allow gray pixels

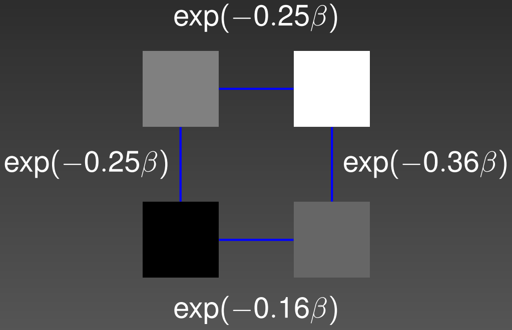
Can generate much more detail

Autonormal model

J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society Series B*, Vol. 48, No. 3, pp. 259–302, 1986.

Besag introduced the autonormal model

$$p(x) \propto \prod_{\text{edges } \{i,j\}} \exp(-\beta(x(i) - x(j))^2)$$



Markov chain Monte Carlo easy for this model

Suppose look at value of state at a single node

- Conditioned on neighboring pixel values...
- ...distribution of pixel value is a truncated normal

Add a density for observed image

y = observed image, x = actual image

$$\ell(y|x) \propto \prod_i \exp(-(y(i) - x(i))^2/[2\kappa])$$

Let $N_A(\mu, \sigma)$ be normal dist. conditioned to lie in A. Then:

$$[y(i)|x(\text{neighbors of } i)] \sim \mathrm{N}_{[0,1]}(\mu_{\text{neighbors}}, \kappa)$$

Autonormal Posterior

$$\pi(y) \propto \left[\prod_{\text{edges } \{i,j\}} \exp(-\beta(x(i) - x(j))^2) \right] \cdot \left[\prod_i \exp(-(y(i) - x(i))^2/[2\kappa]) \right]$$

The big question

How do we generate samples from π?

Two approaches

Markov chain theory

- Create a recurrent Harris chain
- Then limiting distribution equals stationary distribution

Perfect simulation

- Draws exactly from desired distribution
- No need to know the mixing time of the chain
- Often employs recursion

Markov chain theory

Pros

- Easy to build Harris chain where lim dist = stat dist

Cons

- Difficult to show that limit reached quickly
- Without that, not an algorithm
- At best, get approximate samples from π

Perfect simulation

Pros

- Get exact draws from π
- True algorithms

Cons

- Can be difficult to build

Perfect simulation ideas

All of these can be applied to this problem

- Acceptance Rejection [von Neumann 1951]
- Coupling from the Past [Propp Wilson 1996]
- Bounding chains [H. 1999]
- FMMR [Fill et al. 2000]
- Randomness Recycler [Fill H. 1999]
- Partially Recursive Acceptance Rejection [H. 2011]

Only one provably fast for this problem

- Monotonic Coupling from the Past

Standard Harris chain rapidly mixing

A. Gibbs. Convergence in the Wasserstein metric for Markov chain Monte Carlo algorithms with applications to image restoration. *Stochastic Models*, Vol. 20, No. 4, pp. 473–492, 2004

Showed that the standard heat bath Markov chain distribution converged quickly ($O(n \ln n)$) using the Wasserstein metric.

Wasserstein (earth mover's metric)

$$W_p(X_t, \pi)^p = \inf(\mathbb{E}[d(X_t, Y)^p]) \quad (Y \sim \pi)$$

Perfect simulation

M. Huber. Perfect simulation for image restoration. *Stochastic Models*, Vol. 23, No. 3, pp. 475–487, 2007

Utilized monotonic Coupling From the Past protocol

- Generates perfect samples in $O(n \ln n)$ time
- Tricky part is dealing with continuous state space

Coupling from the past (CFTP)

Start with an unknown draw from π

?

Coupling from the past (CFTP)

Start with an unknown draw from π

? $\longrightarrow$?

function that preserves π
(like steps in a Markov chain)

Coupling from the past (CFTP)

Start with an unknown draw from π

$$? \xrightarrow[\text{(like steps in a Markov chain)}]{\text{function that preserves } \pi} ?$$

If function output independent of input,
resulting output is from π

Coupling from the past (CFTP)

Start with an unknown draw from π

? ⟶ ?

function that preserves π
(like steps in a Markov chain)

If function output independent of input,
resulting output is from π

Otherwise use CFTP to draw first stat state
Use same function to get final stat state

CFTP with more notation

Suppose $\phi : \Omega \times [0, 1] \to \Omega$ satisfies

$$\text{if } X \sim \pi \text{ and } U \sim \text{Unif}([0, 1]) \text{ then } \phi(X, U) \sim \pi,$$

Then CFTP runs as follows:

CFTP	*Output:* Y
1)	Draw $U \leftarrow \text{Unif}([0, 1])$
2)	If $\#(\phi(\Omega, U)) = 1$
3)	$Y \leftarrow$ the sole element of $\Phi(\Omega, U)$
4)	Else
5)	$Y \leftarrow \phi(\text{CFTP}, U)$

Actually running CFTP

Key question when implementing CFTP:

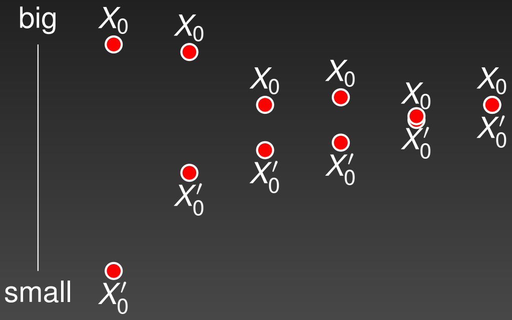
$$\text{How do we know when } \#(\phi(\Omega, U)) = 1?$$

One answer, use monotonic coupling from the past

Monotonic coupling from the past (MCFTP)

Suppose that chain is *monotonic*

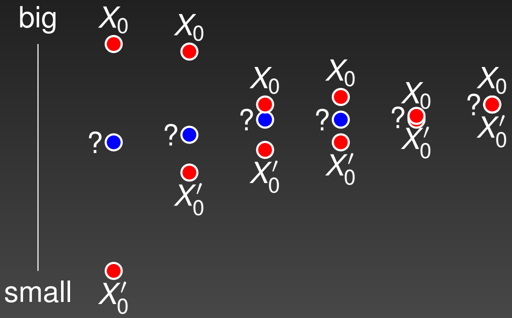
Means if $X_t \preceq X'_t$ then $X_{t+1} \preceq X'_{t+1}$



Monotonic coupling from the past (MCFTP)

Start chains at biggest and smallest state

Unknown stationary state gets squished between them



The other part of CFTP

What if the big chain and small chain don't come together?

The big idea:

- Try running big and small chain together
- If they end up in the same state (*couple*) that is draw
- Otherwise, use recursion:
  - Use CFTP to generate first unknown stationary state
- Run known stat state forward using same random choices
- Result is our draw

Two general frameworks for creating Markov chains

Gibbs sampling [heat bath]

- (Random scan) choose node uniform at random
- Draw new value for node from π conditioned on rest of state

Metropolis

- (Random scan) choose node uniform at random
- Propose new value for that node
- Accept new value for node with easily calculated probability

Forcing continuous chains to couple

Gibbs: Often monotonic, but does not couple

Metropolis: Often not monotonic, but does couple

Solution:

- Run several steps of heat bath to get upper and lower process close
- Take a Metropolis step to get actual coupling

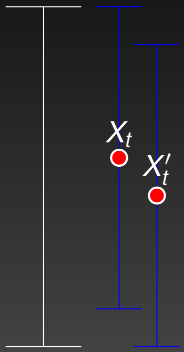
Specifics

In Metropolis: propose value $X_t(i)$ moves to

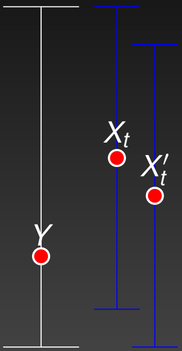
$$Y \sim \text{Unif}([X_t(i) - \epsilon, X_t(i) + \epsilon]).$$

- Want ϵ small so chance of accepting close to 1
- Want ϵ large so coupling easier

How to couple uniforms

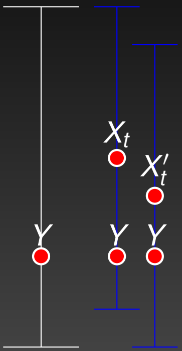


How to couple uniforms



Draw uniformly from interval containing all intervals

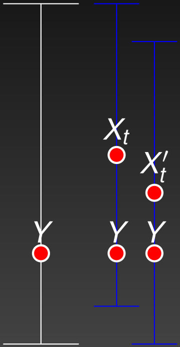
How to couple uniforms



Draw uniformly from interval containing all intervals

If Y works, use as draw from interval

How to couple uniforms



Draw uniformly from interval containing all intervals

If Y works, use as draw from interval

(Otherwise draw independently of Y

Difficulty when near 0 or 1

Modify move somewhat:

$$Y \sim \text{Unif}([\max\{0, X_t(i) - \epsilon\}, \min\{1, X_t(i) + \epsilon]\}).$$

Same technique applies

Other approaches

MCFTP not the only game in town!

- Other pefect simulation protocols often easier to apply to continuous state spaces
- Randomness Recycler [H. & Fill 2001]
- Partially Recursive Acceptance Rejection [H. 2009]
- MCFTP still only one that provably runs in polynomial time

Summary

Perfect simulation for continous state space

- MCFTP still useful
- Gibbs brings states close together...
- ...Metropolis finishes the job
- Other protocols are out there if needed