Perfect simulation with the Randomness Recycler for arbitrary state spaces

Mark Huber

Department of Mathematics and Institute of Statistics and Decision Sciences
Duke University

August 22, 2006

Grayscale images

Configuration: assignment of number in $[0, 1]$ to each pixel

$$\Omega = [0, 1]^V$$

Outline

1. The Model
    - Bayesian approach
    - Autonormal model
2. Perfect sampling
    - What is perfect sampling?
    - CFTP and RR
3. RR for Autonormal
    - RR for continuous state spaces
    - RR for this model
    - Analyzing the running time

Bayesian approach

Three ingredients:

1. Prior Π: probabilistic model on parameter space
2. Statistical model of data X given parameters θ
3. Bayes rule

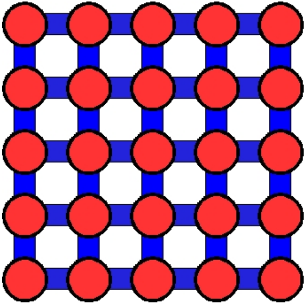
For imaging:

1. Parameters are the true image values
2. Faulty camera: Gaussian error (known variance) on each pixel

Prior provides pixel peer pressure

Idea: pixels interact with neighbors according to some graph

Prior provides pixel peer pressure



Idea: pixels interact with neighbors according to some graph

Prior + defective camera = model

Known parameters: β and σ

$$\Pi(dx) \propto \left[\prod_{\{i,j\}\in E} \exp\{-\beta(x(i)-x(j))^2\}\right] \mathbf{1}(x \in \Omega)\, dx$$

Faulty camera:

$$\mathbb{P}(X \in dx|\theta) \propto \left[\prod_{i\in V} \exp\{-(.5)\sigma^{-2}(x(i)-\theta(i))^2\}\right] \mathbf{1}(x \in \Omega)\, dx$$

Autonormal model

Besag coined term Auto-models [1], later applied to images [2]

Let d be the data configuration:

$$\pi(\mathrm{d}x) = Z^{-1}\left[\prod_{i\in V}\exp\{-(.5)\sigma^{-2}(x(i)-d(i))^2\}\right]$$
$$\left[\prod_{\{i,j\}\in E}\exp\{-\beta(x(i)-x(j))^2\}\right]$$

Node/Edge models

For a graph (V, E), let

$$\pi(\mathrm{d}x) = Z^{-1}\left[\prod_{i \in V} g_i(x(v))\right]\left[\prod_{\{i,j\} \in E} f_{\{i,j\}}(x(i), x(j))\right]$$

Problems of this form:

- Ising and Potts models
- Gas models

Not of this form:

- Random cluster model
- Spanning trees

Outline

1. The Model
   - Bayesian approach
   - Autonormal model
2. Perfect sampling
   - What is perfect sampling?
   - CFTP and RR
3. RR for Autonormal
   - RR for continuous state spaces
   - RR for this model
   - Analyzing the running time

Perfect sampling algorithms

Suppose that π is determined by a finite measure μ:

$$\pi(\mathrm{d}x) = \frac{\mu(\mathrm{d}x)}{Z}, \; Z = \int_\Omega \mu(\mathrm{d}x).$$

Definition

A *perfect sampling* algorithm generates random variates exactly from π without the need to calculate the normalizing constant Z.

CFTP and RR

Coupling from the past:

- Uses underlying Markov chain with π as stationary distribution
- Read-twice, noninterruptible, $\Theta(n \ln n)$
- Can take advantage of monotonicity

Randomness Recycler

- Uses absorbing bivariate Markov chain
- Read-once, interruptible, can be $\Theta(n)$
- Not known how to take advantage of monotonicity

Randomness Recycler is to strong stationary stopping times as
Coupling from the past is to coupling

CFTP and RR

Coupling from the past:

- Uses underlying Markov chain with π as stationary distribution
- Read-twice, noninterruptible, $\Theta(n \ln n)$
- Can take advantage of monotonicity

Randomness Recycler

- Uses absorbing bivariate Markov chain
- Read-once, interruptible, can be $\Theta(n)$
- Not known how to take advantage of monotonicity

Randomness Recycler is to strong stationary stopping times as **Coupling from the past** is to coupling

History of RR

The Randomness Recycler

- generalization of acceptance/rejection methods
- originally used to construct Strong Stationary Stopping Times
- Created for self-organizing lists [Fill,Huber]
- Applications: Ising model, Potts model, proper colorings, discrete gas models
- All discrete state spaces [3]
- Today: extension to continuous spaces

History of RR

The Randomness Recycler

- generalization of acceptance/rejection methods
- originally used to construct Strong Stationary Stopping Times
- Created for self-organizing lists [Fill,Huber]
- Applications: Ising model, Potts model, proper colorings, discrete gas models
- All discrete state spaces [3]
- Today: extension to continuous spaces

History of RR

The Randomness Recycler

- generalization of acceptance/rejection methods
- originally used to construct Strong Stationary Stopping Times
- Created for self-organizing lists [Fill,Huber]
- Applications: Ising model, Potts model, proper colorings, discrete gas models
- All discrete state spaces [3]
- Today: extension to continuous spaces

History of RR

The Randomness Recycler

- generalization of acceptance/rejection methods
- originally used to construct Strong Stationary Stopping Times
- Created for self-organizing lists [Fill,Huber]
- Applications: Ising model, Potts model, proper colorings, discrete gas models
- All discrete state spaces [3]
- Today: extension to continuous spaces

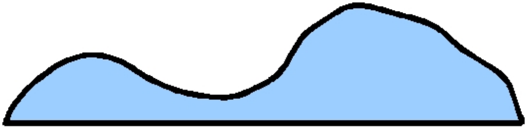
Acceptance rejection in 1 dimension

Acceptance/Rejection

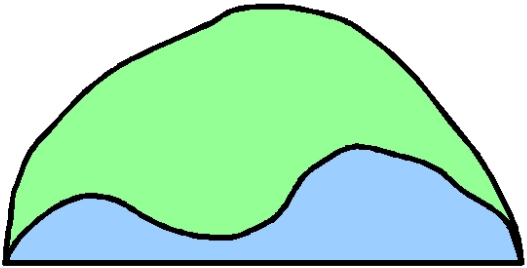
Input: $f(x) \leq m(x)$
Output: X from density proportional to $f(x)$

1) **Repeat**
2) **Draw** X from density proportional to m
3) **Draw** U uniformly from $[0, 1]$
4) **Until** $\mathbf{1}(U < f(X)/m(X))$
5) **Output** X

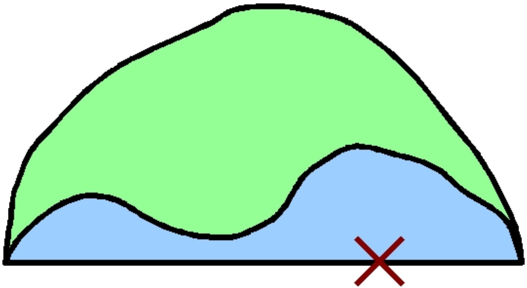
A/R in action



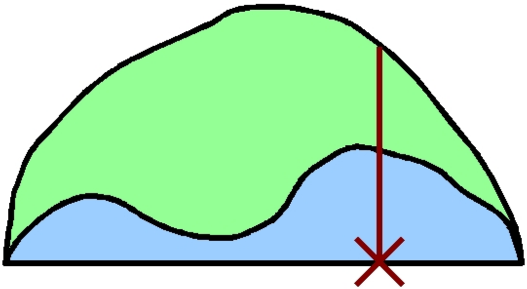
A/R in action



A/R in action



A/R in action



High dimensions: sequential acceptance/rejection

Recall our general problem:

$$\pi(\mathrm{d}x) = Z^{-1} \left[\prod_{i \in V} g_i(x(v)) \right] \left[\prod_{\{i,j\} \in E} f_{\{i,j\}}(x(i), x(j)) \right]$$

Suppose that for all $i \in V$:

$$\int_{\mathbb{R}} g_i(s)\, \mathrm{d}s < \infty$$

and for all $\{i, j\} \in E$:

$$\sup_s f_{\{i,j\}}(s) < \infty$$

Dealing with high dimensions

The idea:

1. Draw each node assignment according to g
2. Accept or reject independently at each edge using f

Dealing with high dimensions

Sequential Acceptance/Rejection

Input: n, $f_e(x)$, $g_i(x)$

1) **Repeat**
2) **Let** $flag = 1$
3) **For** each $i \in V$ do
3) **Draw** $X(i)$ from density proportional to g_i
4) **For** each $e \in E$ do
5) **Draw** U uniformly from $[0, 1]$
6) **Let** $flag \leftarrow flag \cdot \mathbf{1}(U < f_e(X(e))/sup_s f_e(s))$
7) **Until** $flag = 1$
8) **Output** X

One edge at a time

Single run through repeat loop:

Pick edge	Success
Pick edge	Success
Pick edge	Success
Pick edge	Success
Pick edge	Failure
Start Over	

The problem

- Chance of acceptance is exponential in # of dimensions
- Suppose each edge accepts with probability at least α:

$$\mathbb{P}(\text{accepting all edges}) \geq \alpha^{|E|}$$

- Need $\exp\{-\beta\} > 1 - c/n$ for linear run time

Solution is to Recycle

- Do not throw away sample after rejection
- Keep as much as possible that is still “random”
- In other words, recycle

One edge at a time...

Single run through repeat loop:

Pick edge	Success
Pick edge	Success
Pick edge	Success
Pick edge	Success
Pick edge	Failure
Recycle	
Pick edge	Success
Pick edge	Success
⋮	

Usually only small portion of sample contaminated by rejection

Outline

1. The Model
   - Bayesian approach
   - Autonormal model
2. Perfect sampling
   - What is perfect sampling?
   - CFTP and RR
3. RR for Autonormal
   - RR for continuous state spaces
   - RR for this model
   - Analyzing the running time

Elements of an RR algorithm

1. Measurable space $(\Omega, \mathcal{F})$ (the *primary state space*)
2. Measurable space $(\Omega^*, \mathcal{F}^*)$ (the *dual state space*)
3. For all $x^* \in \Omega^*$, a distribution $\Lambda(x^*, \cdot)$ on Ω
4. The target distribution π on Ω
5. A dual state x_π^* where $\Lambda(x_\pi^*, \cdot) = \pi(\cdot)$
6. An initial dual state x_0^* where $\Lambda(x_0^*, \cdot)$ is easy to simulate
7. Bivariate kernel **K** on $\Omega^* \times \Omega$ with design property

In words

RR runs a bivariate chain:

$$A_t = (\text{index for distribution of } X_t, \text{state } X_t \text{ in } \Omega)$$

always making sure that

$$\mathbb{P}(X_t \in A | \text{history of index states}) = \Lambda(\text{last index state}, A)$$

Invariant on the bivariate chain

Notation for history of process up until time t:

$$\mathcal{H}_t^* = \sigma(X_0^*, \ldots, X_t^*)$$

Desire the following invariant:

$$(\forall A \in \mathcal{F})(\mathbb{P}(X_t \in A | \mathcal{H}_{t-1}^*, X_t^* = x_t^*) = \Lambda(x_t^*, A))$$

This gives us interruptibility

$$(\forall A \in \mathcal{F})\mathbb{P}(X_T \in A | T < \infty) = \pi(A)$$

Invariant on the bivariate chain

Notation for history of process up until time t:

$$\mathcal{H}_t^* = \sigma(X_0^*, \ldots, X_t^*)$$

Desire the following invariant:

$$(\forall A \in \mathcal{F})(\mathbb{P}(X_t \in A | \mathcal{H}_{t-1}^*, X_t^* = x_t^*) = \Lambda(x_t^*, A))$$

This gives us interruptibility

$$(\forall A \in \mathcal{F})\mathbb{P}(X_T \in A | T < \infty) = \pi(A)$$

Invariant on the bivariate chain

Notation for history of process up until time t:

$$\mathcal{H}_t^* = \sigma(X_0^*, \ldots, X_t^*)$$

Desire the following invariant:

$$(\forall A \in \mathcal{F})(\mathbb{P}(X_t \in A | \mathcal{H}_{t-1}^*, X_t^* = x_t^*) = \Lambda(x_t^*, A))$$

This gives us interruptibility

$$(\forall A \in \mathcal{F})\mathbb{P}(X_T \in A | T < \infty) = \pi(A)$$

Interruptibility

Classic interruptible algorithm: acceptance/rejection

- Can abort procedure in middle
- Can start over any time without affecting output

Noninterruptible algorithm: **CFTP**

- probability of aborting early must equal 0
- otherwise introduces unknown amount of bias

How is invariant maintained?

Use the **design property**.

Notation for distribution of (X^*_{t+1}, X_{t+1}) given X^*_t and $[X_t | X^*_t] \sim \Lambda(X^*_t, \cdot)$:

$$\mathbb{P}^{\Lambda}(X^*_{t+1} \in B, X_{t+1} \in A | X^*_t = x^*) := \int_{x \in \Omega} \Lambda(x^*, \mathrm{d}x) \mathbb{P}(X^*_{t+1} \in B, X_{t+1} \in A | X^*_t = x^*, X_t = x)$$

Design property

Kernel **K** has the *design property* if for all x^* and y^* satisfying $\mathbb{P}(X^*_{t+1} \in \mathrm{d}y^* | X^*_t = x^*) > 0$ and all $A \in \mathcal{F}$:

$$\Lambda(y^*, A) = \mathbb{P}^{\Lambda}(X_{t+1} \in A | X^*_{t+1} = y^*, X^*_t = x^*)$$

Reversibility

The design property similar to reversibility for Markov chains

- Reversibility is how Gibbs and Metropolis work
- Do not need to check reversibility to use Gibbs and Metropolis
- Gibbs or Metropolis guarantee reversibility
- Similarly, there is automatic way to get design property
- Once the design property in place, generating variates easy

Generating variates with RR

The Randomness Recycler method

1) **Let** $t \leftarrow 0$, $X_0^* \leftarrow x_0^*$
2) **Choose** X_0 from distribution $\Lambda(x_0^*, \cdot)$
3) **While** $X_t^* \neq x_\pi^*$ do steps 4 and 5
3) **Choose** (X_{t+1}^*, X_{t+1}) by taking on step in bivariate chain
4) **Let** $t \leftarrow t + 1$
5) **Let** $T \leftarrow t$
6) **Output** X_T

Elements of RR for node/edge models

Begin with no edges in graph

- makes generating variate easy
- all nodes independent

Add in edges

- index needs to keep track of which edges are added

When reject recycle

- accept probability $f(x(e))/sup_s f(s)$
- reject weight $1 - f(x(e))/sup_s f(s)$
- freeze endpoints of e at their current values

Some are as before...

1. Measurable space $(\Omega, \mathcal{F})$ (the *primary state space*)

$$\Omega = [0, 1]^V, \quad \text{with Borel sets}$$

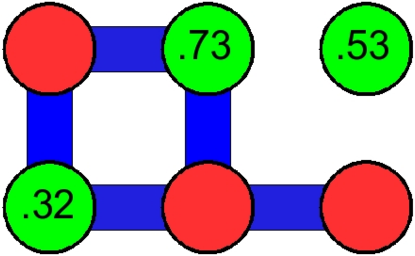
4. The target distribution π on Ω

$$\pi(\mathrm{d}x) = Z^{-1} \left[\prod_{i \in V} \exp\{-(.5)\sigma^{-2}(x(i) - d(i))^2\} \right] \left[\prod_{\{i,j\} \in E} \exp\{-\beta(x(i) - x(j))^2\} \right]$$

Dual state space

The **dual state space** keeps track of two things

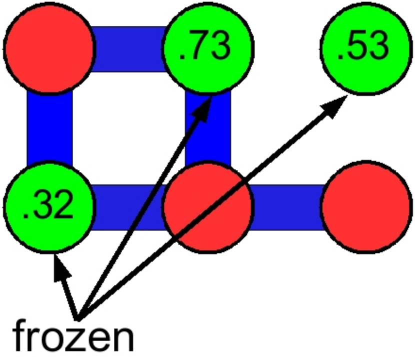
- Which edges are *enforced* in the graph
- Which nodes are *frozen* at their values



Dual state space

The **dual state space** keeps track of two things

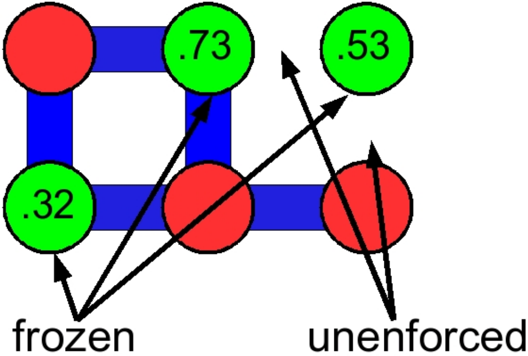
- Which edges are *enforced* in the graph
- Which nodes are *frozen* at their values



Dual state space

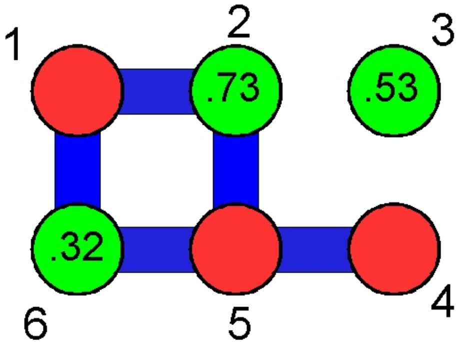
The **dual state space** keeps track of two things

- Which edges are *enforced* in the graph
- Which nodes are *frozen* at their values



Notation for index sets

2. Measurable space $(\Omega^*, \mathcal{F}^*)$ (the *dual state space*)



$$x^* = (\emptyset, .73, .53, , \emptyset, \emptyset, .32, \{\{1,2\},\{1,6\},\{2,5\},\{4,5\},\{5,6\}\})$$

How to index a distribution

3. For all $x^* \in \Omega^*$, a distribution $\Lambda(x^*, \cdot)$ on Ω

$$\begin{aligned}
\Omega(x^*) &:= \{x \in [0,1]^n : (\forall v \in V)(x^*(v) \neq \emptyset \rightarrow x(v) = x^*(v))\} \\
H(x^*, x) &:= -\frac{1}{2\sigma^2} \sum_{v \in V} (x(v) - d(v))^2 - \sum_{\{i,j\} \in x^*(n+1)} \frac{1}{2} \beta (x(i) - x(j))^2 \\
\Lambda(x^*, dx) &:= Z(x^*)^{-1} \mathbf{1}(x \in \Omega(x^*)) \exp(-H(x^*, x)),
\end{aligned}$$

Configurations in $\Omega(x^*)$ have nodes frozen at values
$H(x^*, x)$ only enforces edges in $x^*(n+1)$

Special dual states

5. A dual state x_π^* where $\Lambda(x_\pi^*, \cdot) = \pi(\cdot)$

$$x_\pi^* = (\emptyset, \ldots, \emptyset, E)$$

(x_π^* = all edges enforced, no nodes frozen)

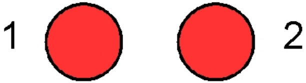
6. An initial dual state x_0^* where $\Lambda(x_0^*, \cdot)$ is easy to simulate

$$x_0^* = (\emptyset, \ldots, \emptyset, \emptyset)$$

(easy to generate when no edges!)

One step: adding an edge

Consider an edge $\{1, 2\}$ that is not enforced

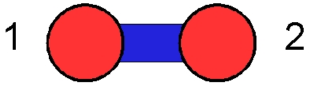


Accept the addition of the edge with probability

$$f_{\{1,2\}}(x(1), x(2))/sup_{a,b \in [0,1]} f_{\{1,2\}}(a, b)$$

One step: adding an edge

Consider an edge $\{1, 2\}$ that is not enforced



Accept the addition of the edge with probability

$$f_{\{1,2\}}(x(1), x(2))/sup_{a,b\in[0,1]} f_{\{1,2\}}(a, b)$$

Rejection

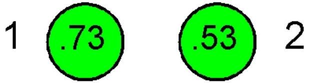
When accept, multiplies weight of configuration by:

$$f_{\{1,2\}}(x(1), x(2))$$

When reject, multiplies weight of configuration by:

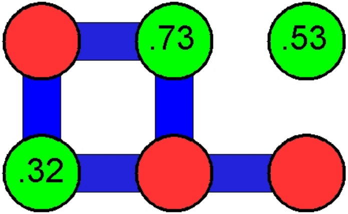
$$1 - f_{\{1,2\}}(x(1), x(2))/sup_{a,b \in [0,1]} f_{\{1,2\}}(a, b)$$

Solution: freeze endpoints of the edge



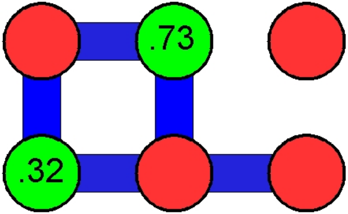
One step: Unfreezing a node

- Suppose no edges adjacent to frozen node
- Recolor node according to g_v

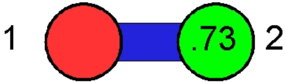


One step: Unfreezing a node

- Suppose no edges adjacent to frozen node
- Recolor node according to g_v



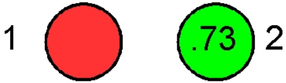
One step: removing an edge



Accept the removal of the edge with probability

$$f_{\{1,2\}}(x(1), x(2))^{-1} / sup_{a,b \in [0,1]} f_{\{1,2\}}(a, b)^{-1}$$

One step: removing an edge



Accept the removal of the edge with probability

$$f_{\{1,2\}}(x(1), x(2))^{-1} / sup_{a,b \in [0,1]} f_{\{1,2\}}(a, b)^{-1}$$

Rejection

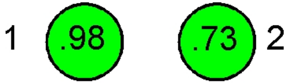
When accept, multiplies weight of configuration by:

$$f_{\{1,2\}}(x(1), x(2))^{-1}$$

When reject, multiplies weight of configuration by:

$$1 - f_{\{1,2\}}(x(1), x(2))^{-1} / sup_{a,b \in [0,1]} f_{\{1,2\}}(a, b)^{-1}$$

Solution: freeze endpoints of the edge



Algorithm overview

- If no frozen nodes exist, try to add edge
- If frozen node has adjacent edge, try to remove edge
- If frozen nodes exist with no adjacent edge, recolor nodes

How fast is algorithm?

Theorem

Let

$$\begin{aligned} \Delta &:= \text{maximum degree of the graph} \\ \tilde{p} &:= \min_{s:f(s)>0} f(s)/\max_s f(s) \\ \delta &:= -1 + (2\Delta - 1)[1 - \tilde{p}] \\ T &:= \text{number of steps taken by one run of RR.} \end{aligned}$$

If $\delta < 0$ *then*

$$\mathbb{E}[T] \leq \min\{3|E|\delta^{-1}, 3|E|^2\}.$$

Each step takes time $O(\Delta)$ *to execute.*

Proof

Consider

$\Phi(x_t^*) = \#$ of unenforced edges $+ \#$ edges next to frozen nodes in x_t^*

Can show that

$$\mathbb{E}(\Phi(X_{t+1}^*|X_t^*)) \leq X_t^* + \delta, \text{ w/ probability 1}$$

For instance, removing an edge analysis

Chance of accepting and removing an edge is at least

$$\tilde{p}$$

Chance of accepting and increasing Φ by $2\Delta - 2$ at most

$$1 - \tilde{p}$$

Hence

$$\mathbb{E}(\Phi(X_{t+1}^* | X_t^*)) \leq X_t^* - \tilde{p} + (1 - \tilde{p})(2\Delta - 2)$$

For Autonormal models

Corollary

Let

$$\begin{aligned} \Delta &:= \textit{maximum degree of the graph} \\ \delta &:= -1 + (2\Delta - 1)[1 - \exp(-\beta)] \\ T &:= \textit{number of steps taken by one run of RR.} \end{aligned}$$

If $\delta < 0$ *or equivalently:*

$$\beta \leq \ln\left(1 + \frac{1}{2\Delta - 1}\right)$$

then

$$\mathbb{E}[T] \leq \min\{3|E|\delta^{-1}, 3|E|^2\}.$$

Each step takes time $O(\Delta)$ *to execute.*

Other work

A. Gibbs [4] showed

- Gibbs sampler for Autnormal model converges in $O(n \ln n)$ time in Wasserstein metric
- Came close to similar result for perfect sampling

Method can be updated

- Gibbs + catalytic coupling + multishift coupling for uniforms gives perfect simulation with CFTP [5]
- Run time $O(n \ln n)$ (constant complex function of σ, β)

Summary

Large β (low temperature) **CFTP** wins

- Run time $\Theta(n \ln n)$
- Pick one: interruptible, read-once

Small β (high temperature) **RR** wins

- Run time $\Theta(n)$
- Interruptible, read-once

More complicated problems

- **CFTP** loses monontonic advantage, use variants
- **RR** same algorithm as presented earlier

References

J. Besag.
Spatial interaction and the statistical analysis of lattice systems (with discussion).
J. Roy. Stat. Soc. B, 36:192–236, 1974.

J. Besag.
On the statistical analysis of dirty pictures.
J. Roy. Stat. Soc. B, 48:259–302, 1986.

J. A. Fill and M. L. Huber.
The Randomness Recyler approach to perfect sampling.
In *Proc. 53rd Session of the ISI*, pages 69–72, 2000.

A. Gibbs.
Convergence in the Wasserstein metric for markov chain monte carlo algorithms with applications to image restoration.
Stochastic model, 20:473–492, 2004.

M. Huber.
How many scales does a greyscale need for rapid mixing?
preprint, 2006.