Introduction to MCMC for deep learning

Roadmap:

— Motivation: probabilistic modelling
— Monte Carlo, importance sampling
— Gibbs sampling, M–H
— Auxiliary variable methods

Iain Murray
School of Informatics, University of Edinburgh

“a probabilistic framework isn’t necessary,

or even always useful. . .

. . . retro-fitting our new models to some probabilistic framework has little benefit”

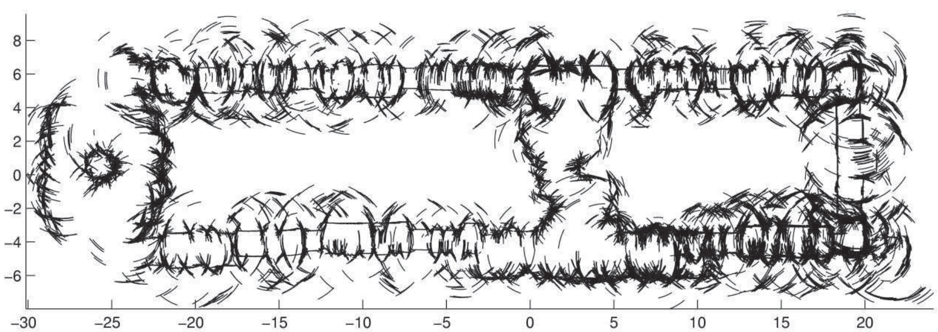
Drawing model fantasies

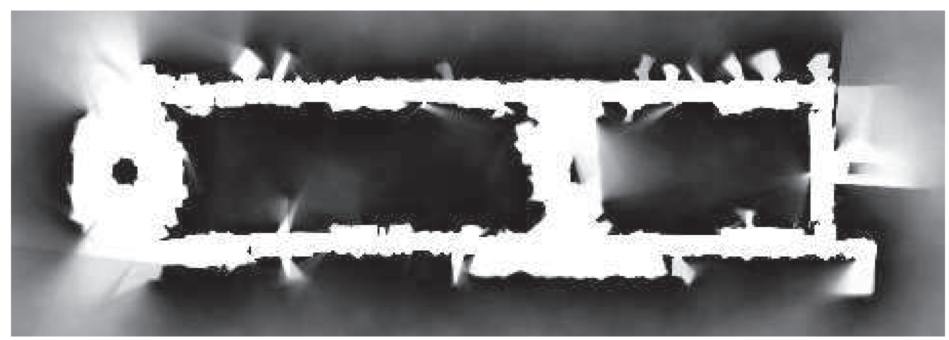
— Insight into models

— Improve learning

— Communication

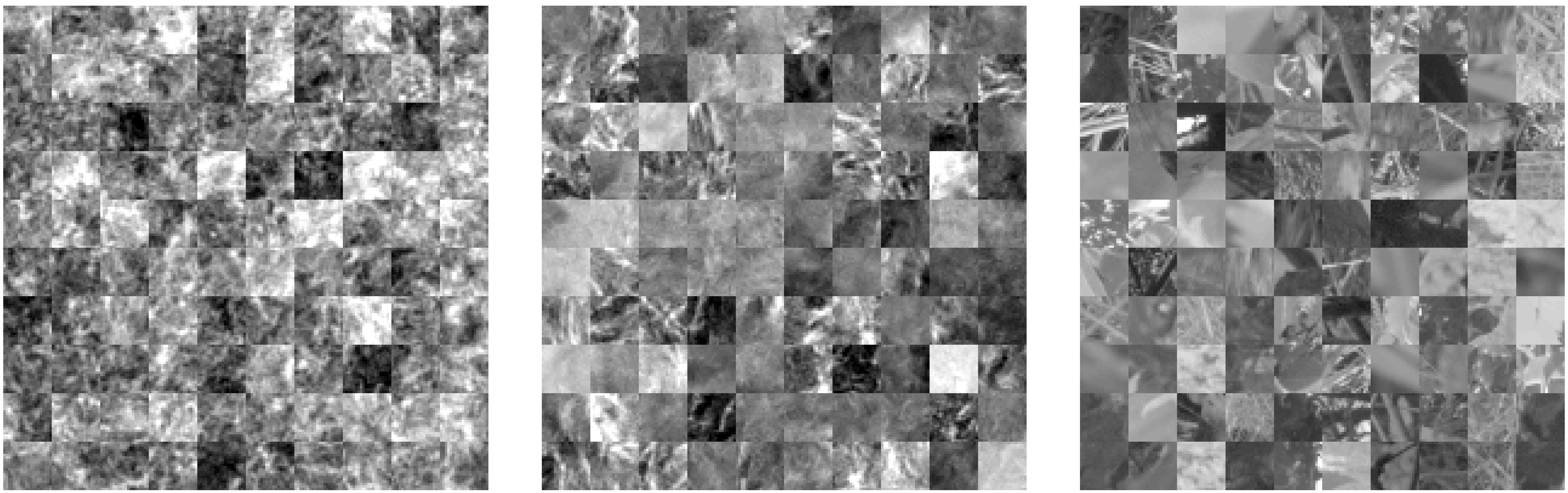
Polygonal random fields





Paskin and Thrun (2005)

Natural patch fantasies



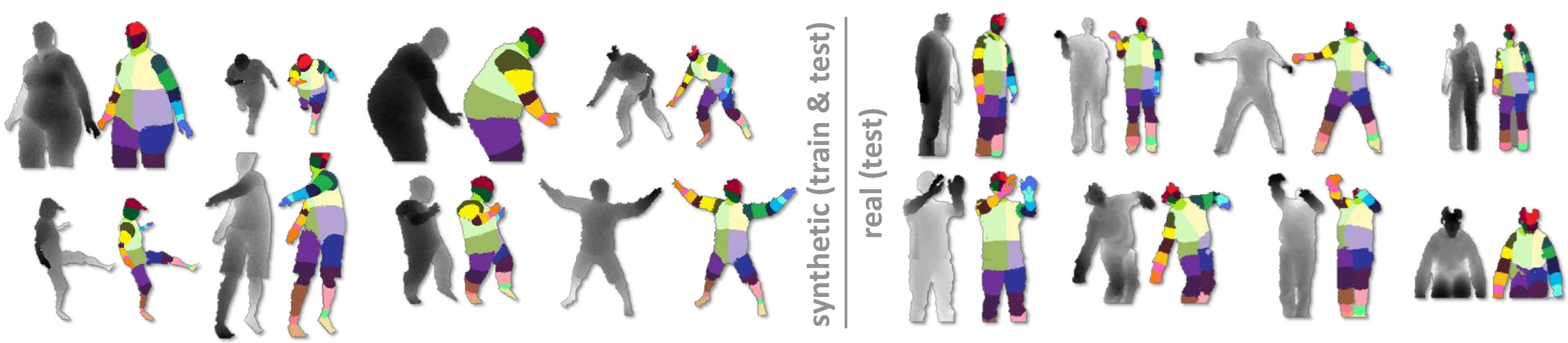
From Osindero and Hinton (2008)

Creating training data

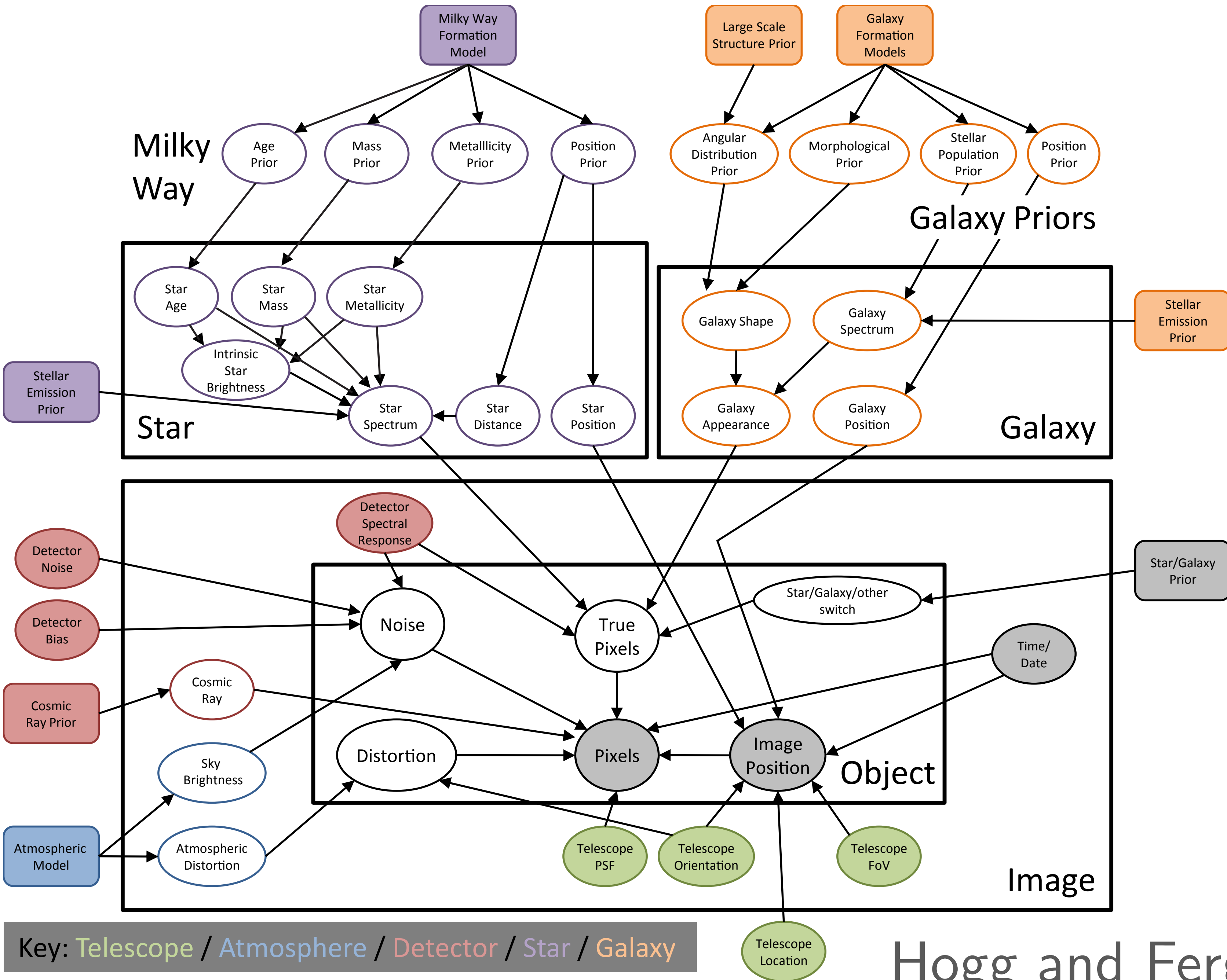
Microsoft Kinect (Shotton et al., 2011)

Shallow learning: random forest applied to fantasies

Future deep learning?



Scientific deep models



Hogg and Fergus, 2011

Roadmap

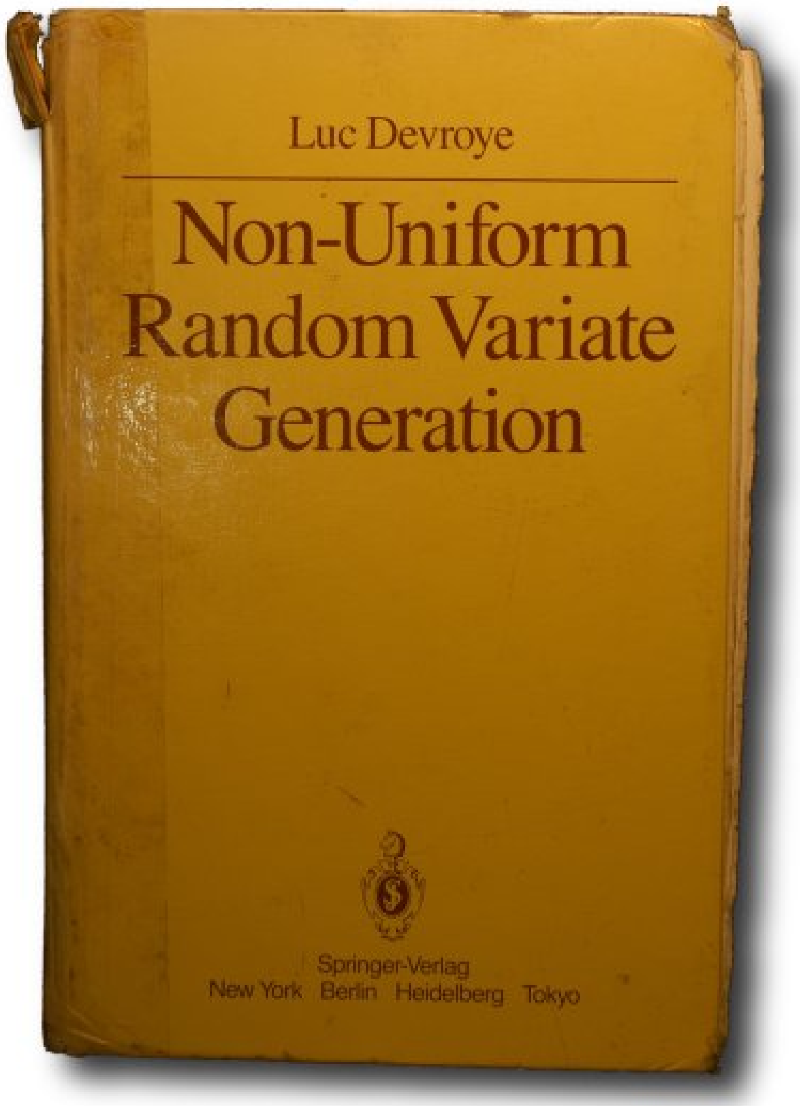
— Probabilistic models

— **Simple Monte Carlo**
Importance Sampling

— Markov chain Monte Carlo (MCMC)
Gibbs sampling, M–H

— Auxiliary variable methods
Swendsen–Wang, HMC

Sampling simple distributions



Use library routines for univariate distributions

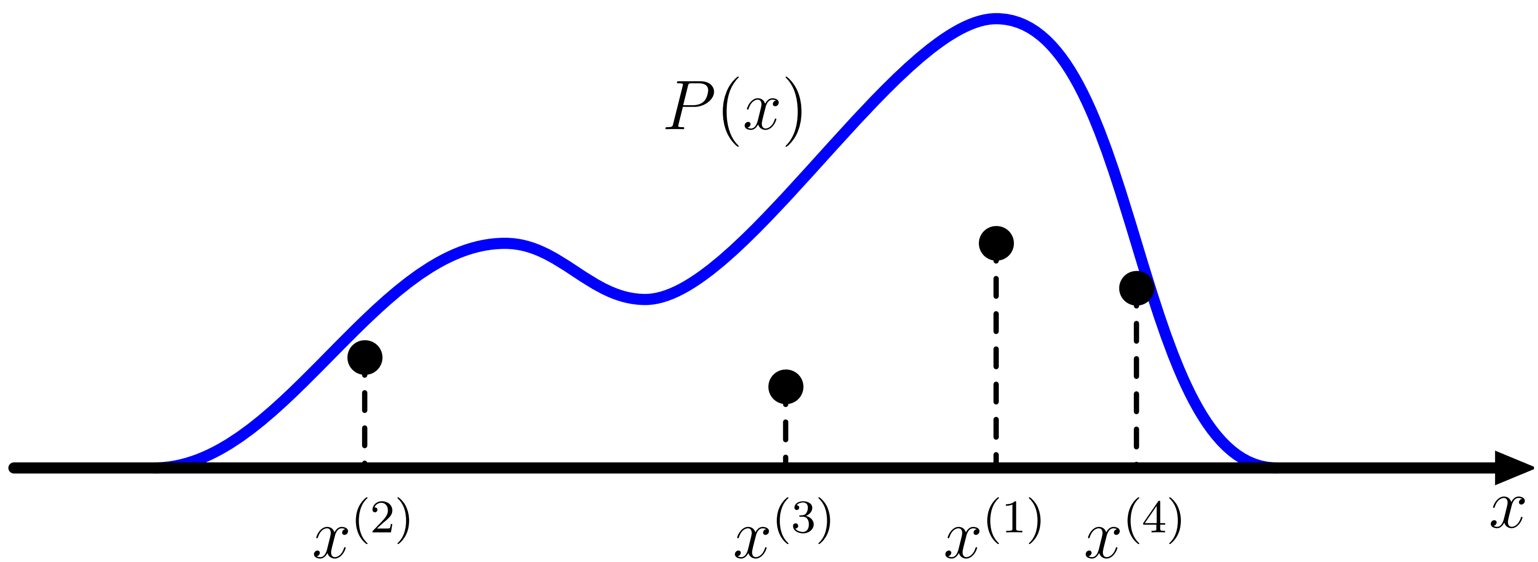
(and some other special cases)

This book (free online) explains how some of them work

`http://luc.devroye.org/rnbookindex.html`

Sampling from densities

Draw points uniformly under the curve:



Probability mass to left of point $\sim$ Uniform[0,1]

Rejection sampling

Sampling from $\pi(x)$ using tractable $q(x)$:

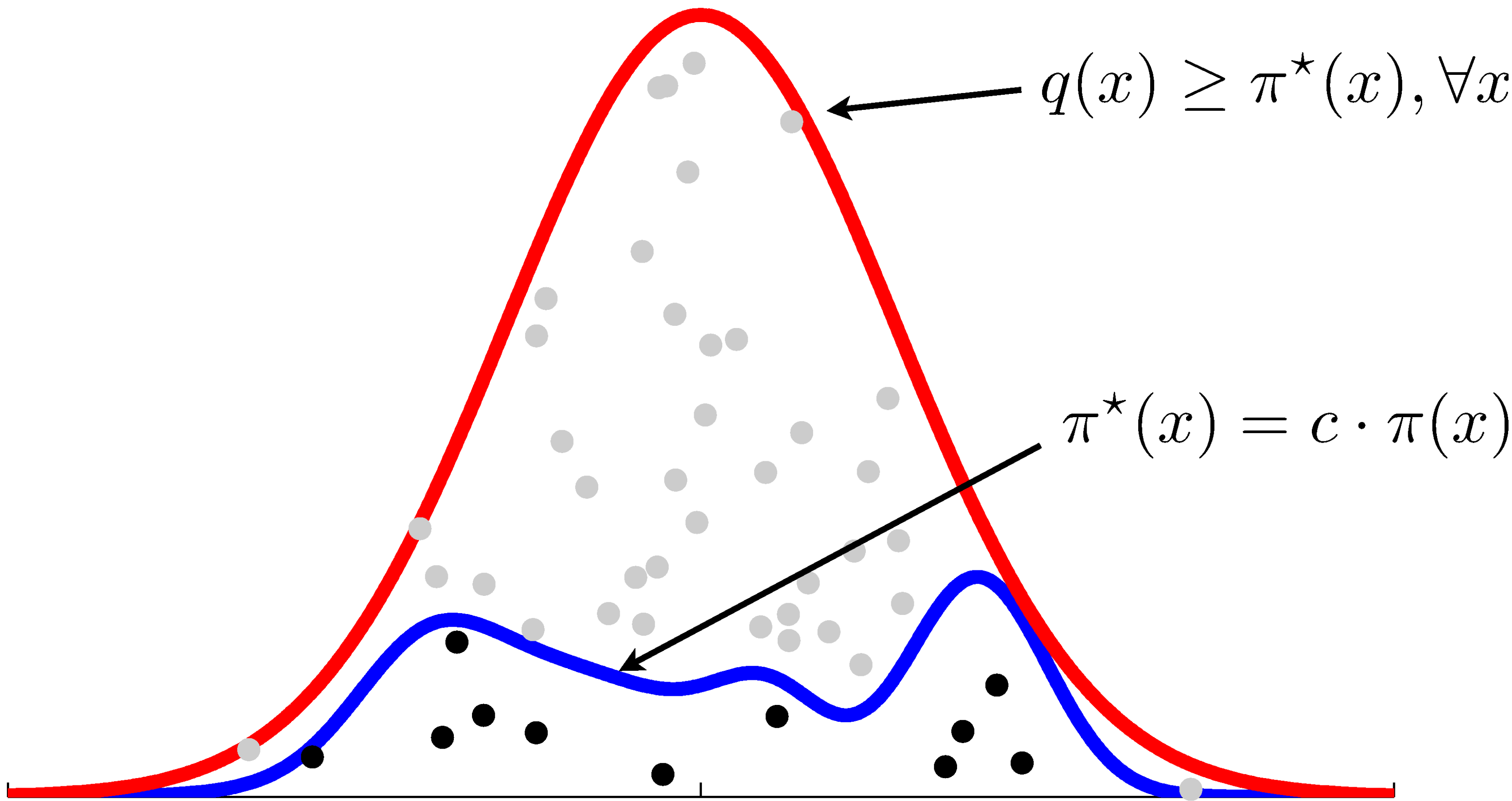
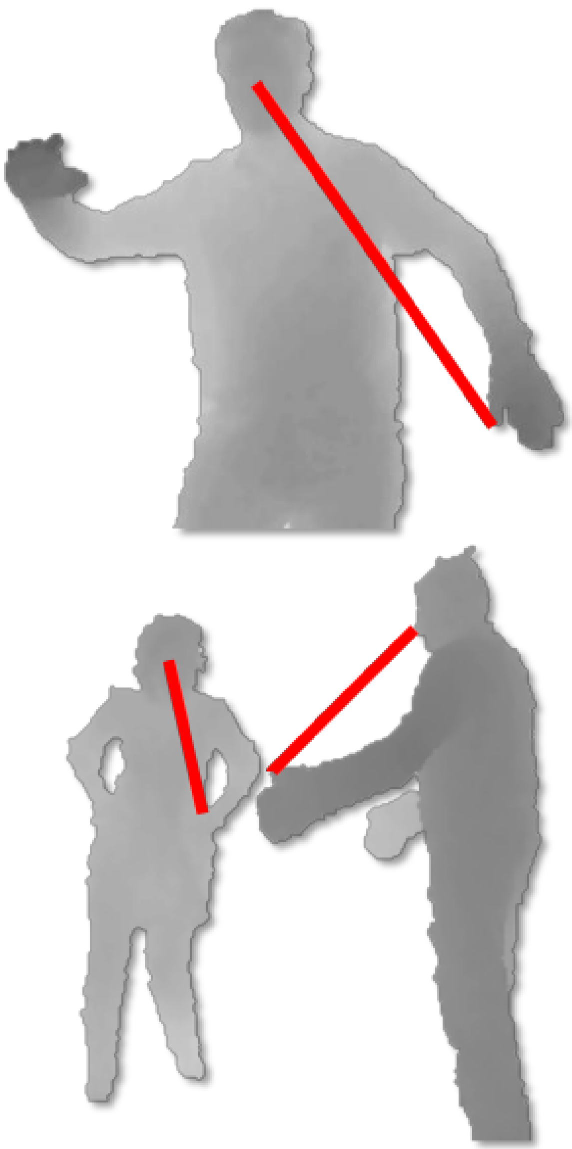


Figure credit: Ryan P. Adams

Simple Monte Carlo



$$\int f(\mathbf{x}) P(\mathbf{x})\, \mathrm{d}\mathbf{x}$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} f(\mathbf{x}^{(s)}),$$

$$\mathbf{x}^{(s)} \sim P(\mathbf{x})$$

Unbiased. Variance $\sim 1/S$

Aside: Marginalization

Function of subset, $\int f(\mathrm{x}_C) P(\mathbf{x}_C)\, \mathrm{d}\mathbf{x}_C$

Simulate all variables anyway:

$$I \approx \frac{1}{S}\sum_{s=1}^{S} f(\mathbf{x}_C^{(s)}), \qquad \mathbf{x}^{(s)} \sim P(\mathbf{x})$$

Importance sampling

Rewrite integral: expectation under simple distribution Q:

$$\int f(x)\, P(x)\, \mathrm{d}x \;=\; \int f(x)\, \frac{P(x)}{Q(x)}\, Q(x)\, \mathrm{d}x,$$

$$\approx\; \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})\, \frac{P(x^{(s)})}{Q(x^{(s)})}, \qquad x^{(s)} \sim Q(x)$$

Simple Monte Carlo applied to any integral.

Unbiased and independent of dimension?

Importance sampling (2)

Previous slide assumed we could evaluate $P(x) = P^*(x)/\mathcal{Z}_P$

$$\int f(x)\,P(x)\,\mathrm{d}x \;\approx\; \frac{\mathcal{Z}_Q}{\mathcal{Z}_P}\,\frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\,\underbrace{\frac{P^*(x^{(s)})}{Q^*(x^{(s)})}}_{w^{*(s)}}, \quad x^{(s)} \sim Q(x)$$

$$\approx\; \cancel{\frac{1}{S}}\sum_{s=1}^{S} f(x^{(s)})\,\frac{w^{*(s)}}{\cancel{\frac{1}{S}}\sum_{s'} w^{*(s')}}$$

This estimator is **consistent** but **biased**

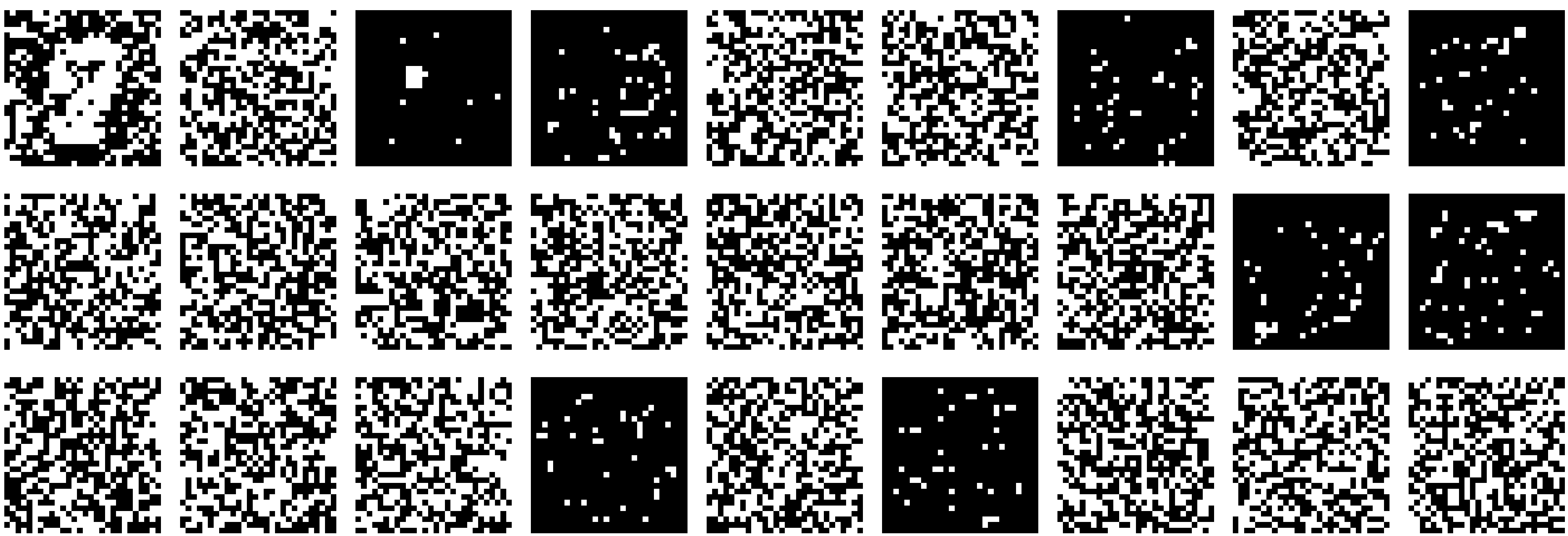
Exercise: Prove that $\mathcal{Z}_P/\mathcal{Z}_Q \approx \frac{1}{S}\sum_s w^{*(s)}$

Rejection sampling RBMs

Product of experts:

— Draw fantasy from each expert

— If they happen to be exactly the same, accept!



Application to large problems

Approximations scale badly with dimensionality

Example: $P(x) = \mathcal{N}(0, \mathbb{I}), \quad Q(x) = \mathcal{N}(0, \sigma^2\mathbb{I})$

Rejection sampling:

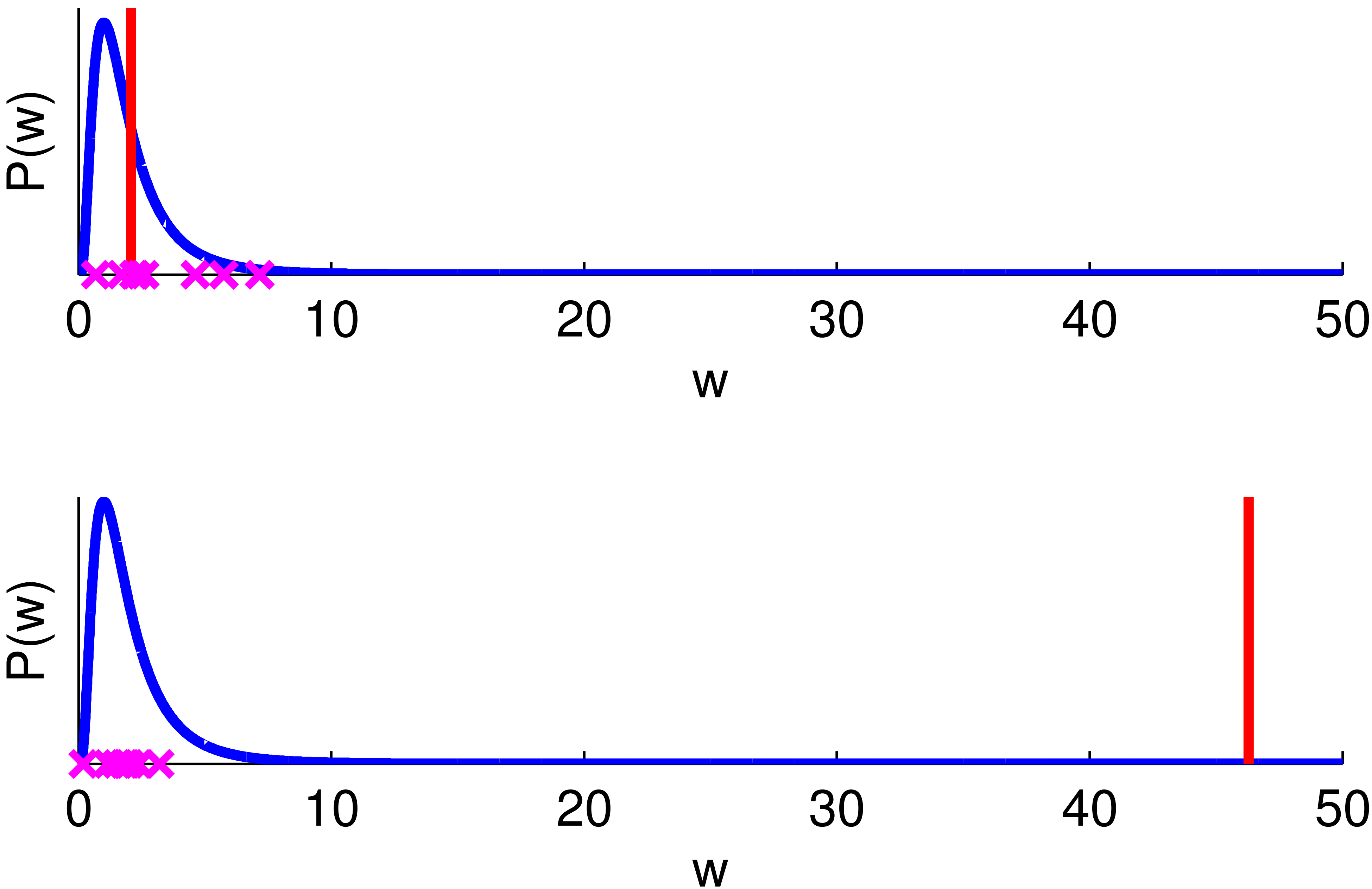
Requires $\sigma \geq 1$. Fraction of proposals accepted $= \sigma^{-D}$

Importance sampling:

$$\text{Var}[P(x)/Q(x)] = \left(\frac{\sigma^2}{2-1/\sigma^2}\right)^{D/2} - 1$$

Infinite / undefined variance if $\sigma \leq 1/\sqrt{2}$

Unbiased positive estimators



Roadmap

— Probabilistic models

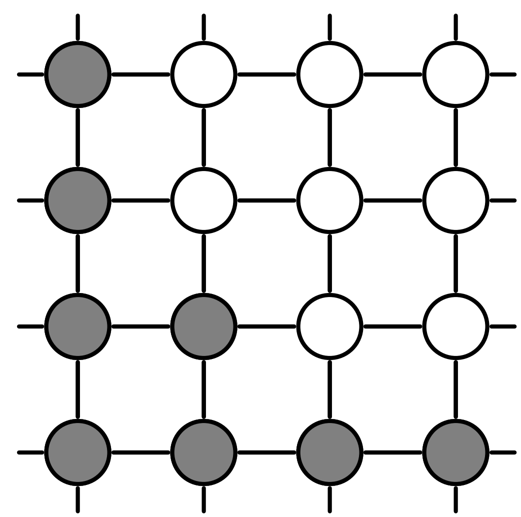
— Simple Monte Carlo
Importance Sampling

— **Markov chain Monte Carlo, MCMC**
Gibbs sampling, M–H

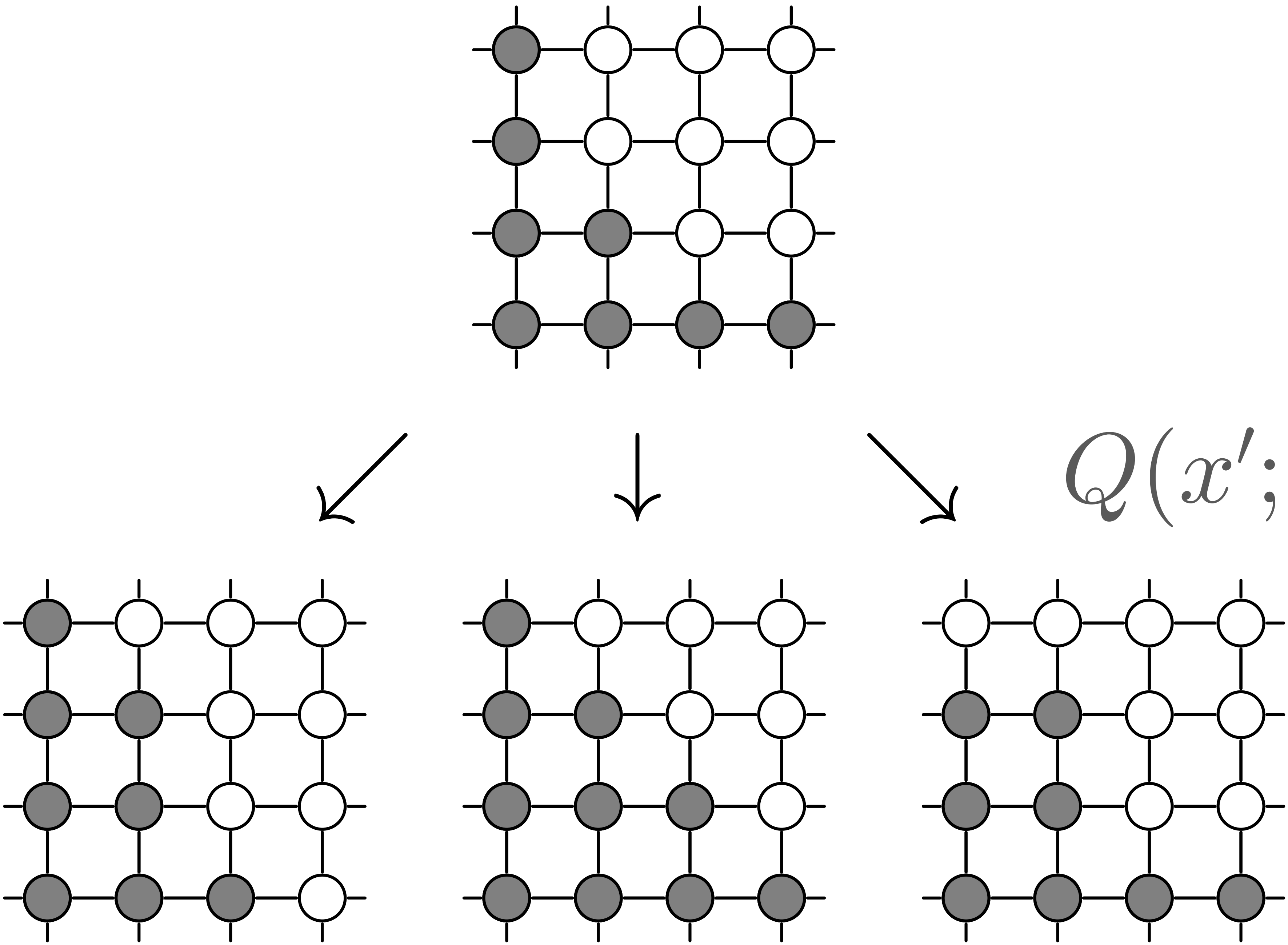
— Auxiliary variable methods
Swendsen–Wang, HMC

Target distribution

$$P(\mathbf{x}) = \frac{1}{Z} e^{-E(\mathbf{x})}$$

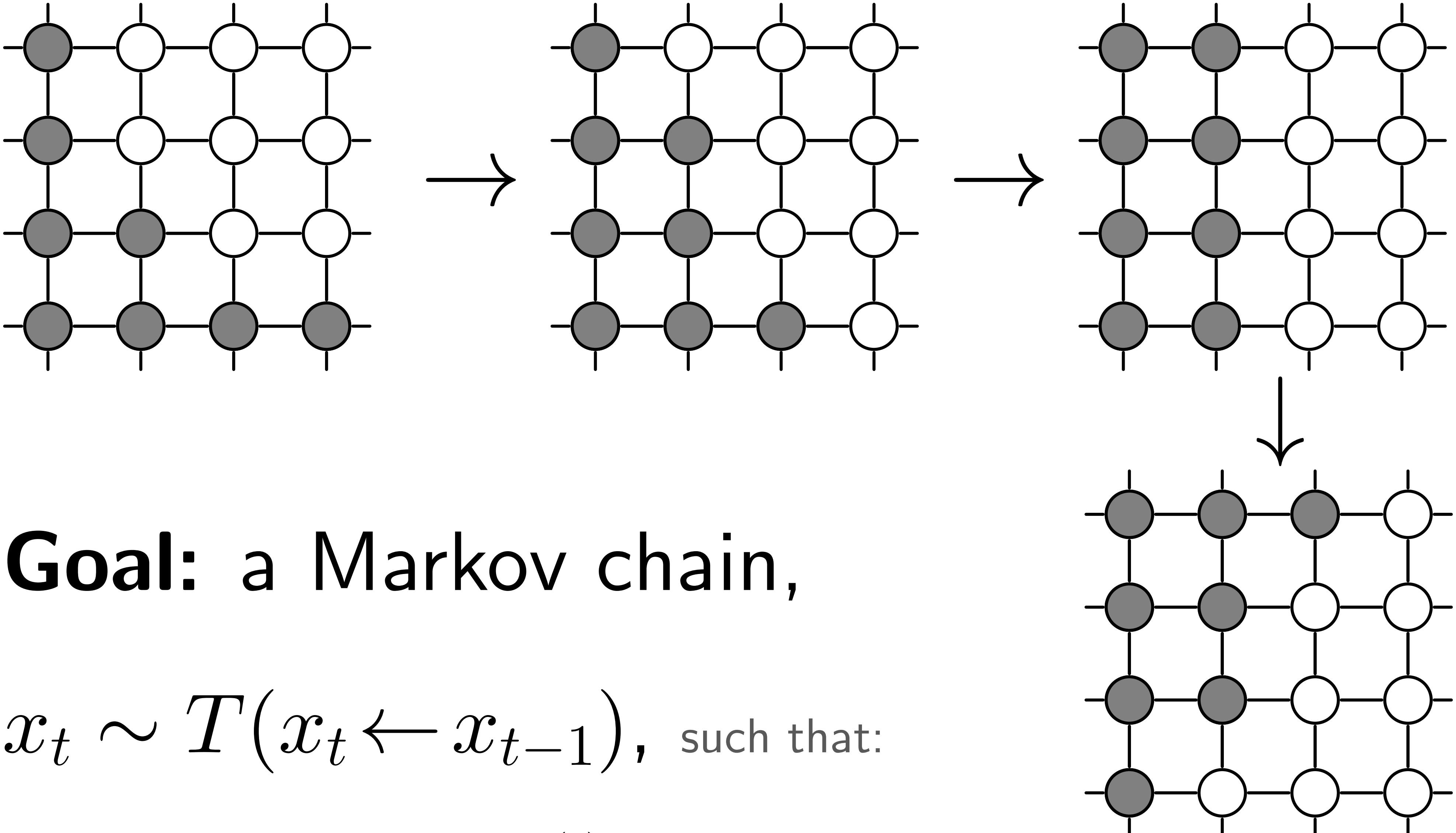
e.g. $\mathbf{x} =$ 

Local moves



$Q(x'; x)$

Markov chain exploration



Goal: a Markov chain,

$x_t \sim T(x_t \leftarrow x_{t-1})$, such that:

$P(x^{(t)}) = e^{-E(x^{(t)})}/Z$ for large t.

Invariant/stationary condition

If $x^{(t-1)}$ is a sample from P,

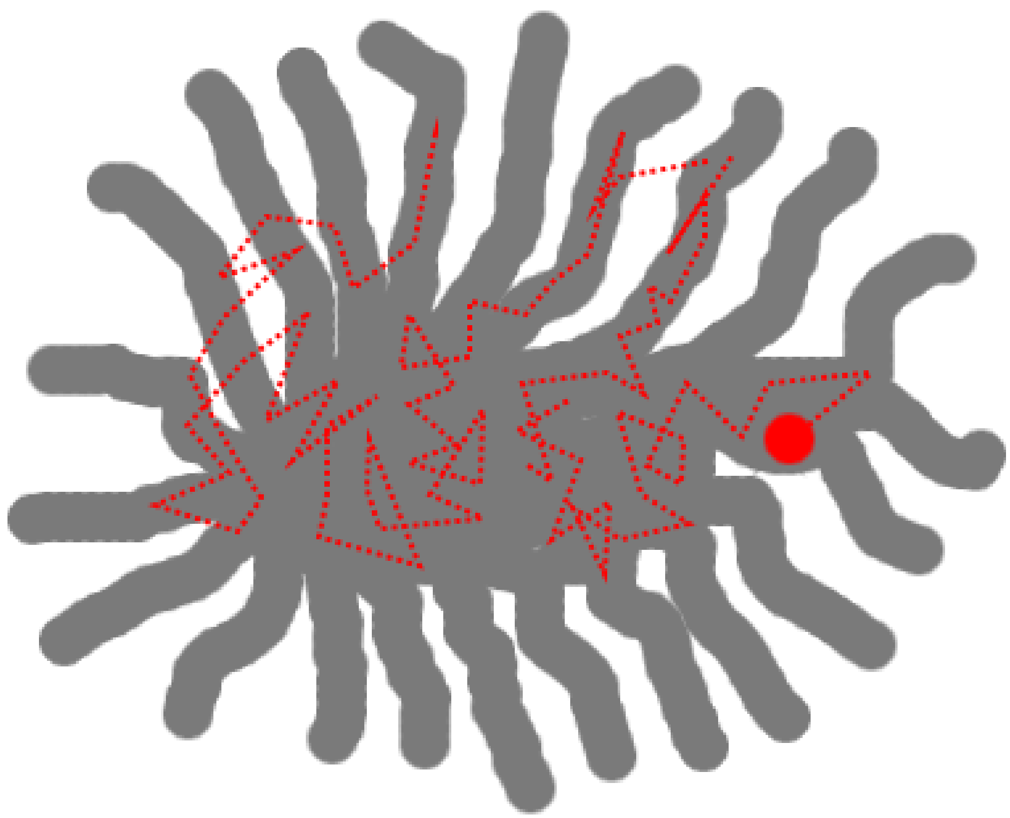
$x^{(t)}$ is also a sample from P.

$$\sum_x T(x' \leftarrow x) P(x) = P(x')$$

Ergodicity

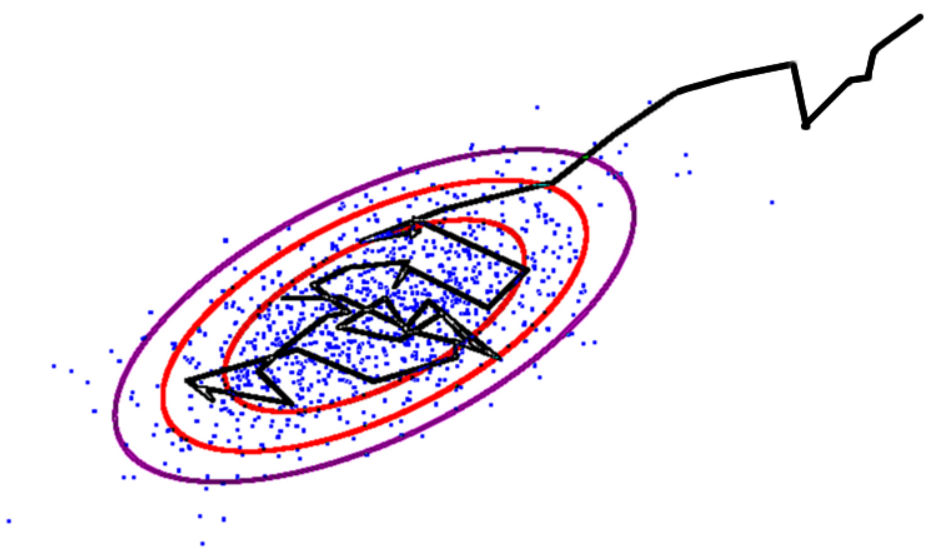
Unique invariant distribution

if 'forget' starting point, $x^{(0)}$



Quick review

MCMC: biased random walk exploring a target dist.



Markov steps,

$$x^{(s)} \sim T\big(x^{(s)} \leftarrow x^{(s-1)}\big)$$

MCMC gives approximate, correlated samples

$$\mathbb{E}_P[f] \approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})$$

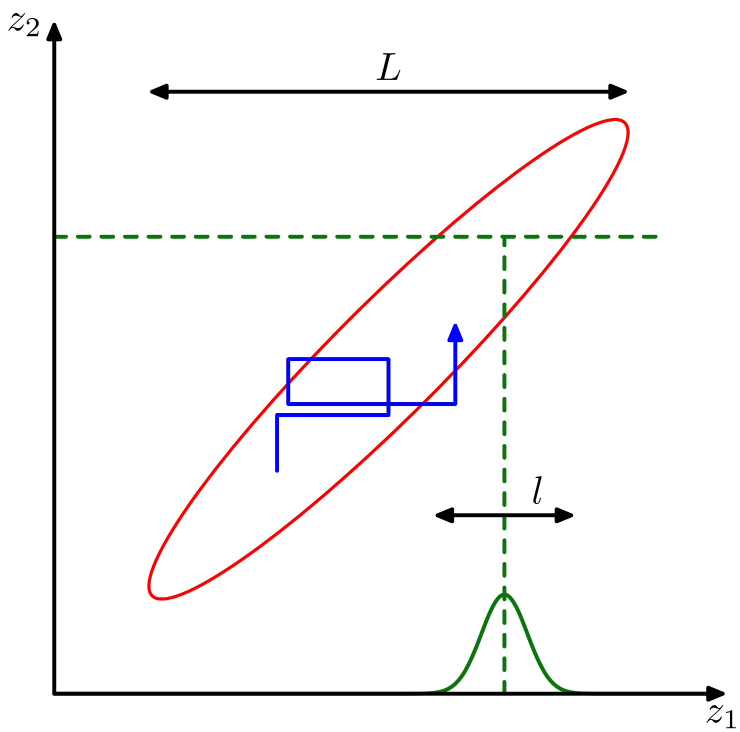
T must leave target invariant

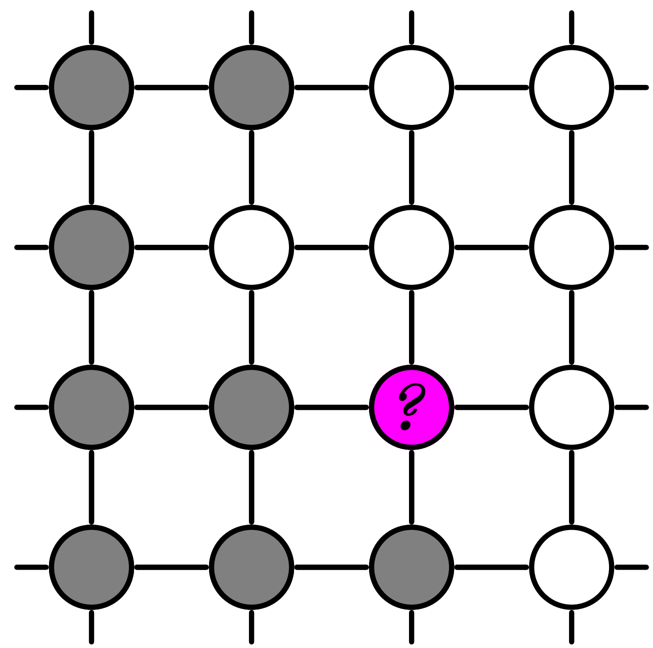
T must be able to get everywhere in K steps

Gibbs sampling

Pick variables in turn or randomly,

and resample $P(x_i|\mathbf{x}_{j\neq i})$





$$T_i(\mathbf{x}' \leftarrow \mathbf{x}) = P(x_i' \,|\, \mathbf{x}_{j\neq i})\, \delta(\mathbf{x}'_{j\neq i} - \mathbf{x}_{j\neq i})$$

Gibbs sampling correctness

$$P(\mathbf{x}) = P(x_i \,|\, \mathbf{x}_{\setminus i}) \, P(\mathbf{x}_{\setminus i})$$

Simulate by drawing $\mathbf{x}_{\setminus i}$, then $x_i \,|\, \mathbf{x}_{\setminus i}$

Draw $\mathbf{x}_{\setminus i}$: sample $\mathbf{x}$, throw initial x_i away

Reverse operators

If T leaves $P(x)$ stationary, define a *reverse operator*

$$R(x \leftarrow x') = \frac{T(x' \leftarrow x)\, P(x)}{\sum_x T(x' \leftarrow x)\, P(x)} = \frac{T(x' \leftarrow x)\, P(x)}{P(x')}.$$

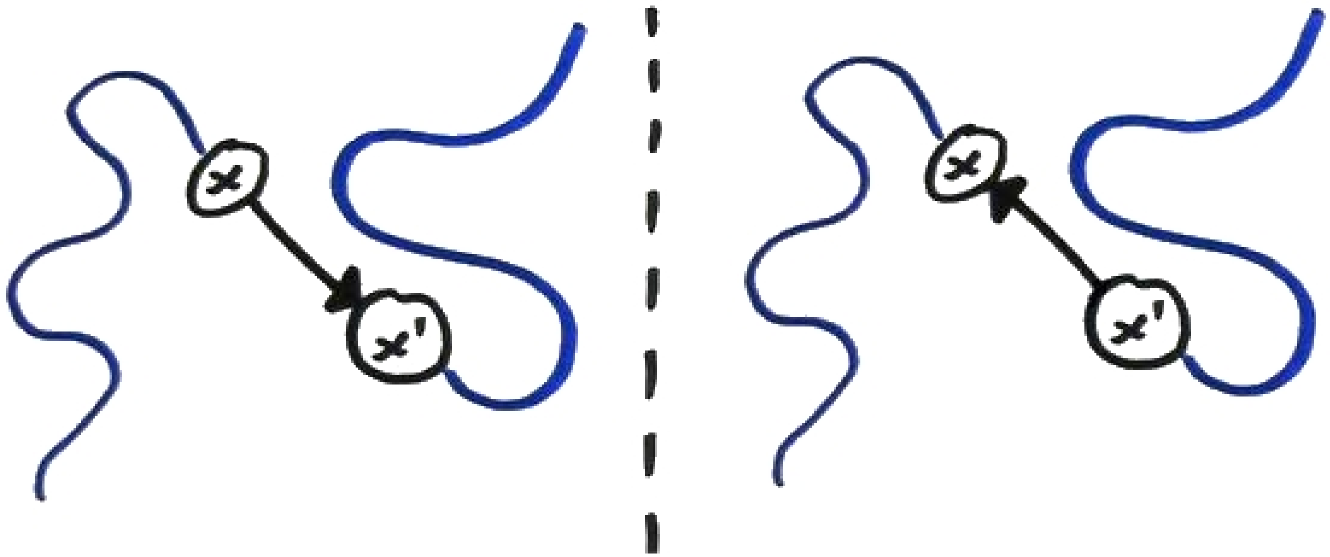
A necessary condition: there exists R such that:

$$T(x' \leftarrow x)\, P(x) \;=\; R(x \leftarrow x')\, P(x'), \qquad \forall x, x'.$$

If $R = T$, known as **detailed balance** (not necessary)

Balance condition

$$T(x' \leftarrow x)\, P(x) \;=\; R(x \leftarrow x')\, P(x')$$



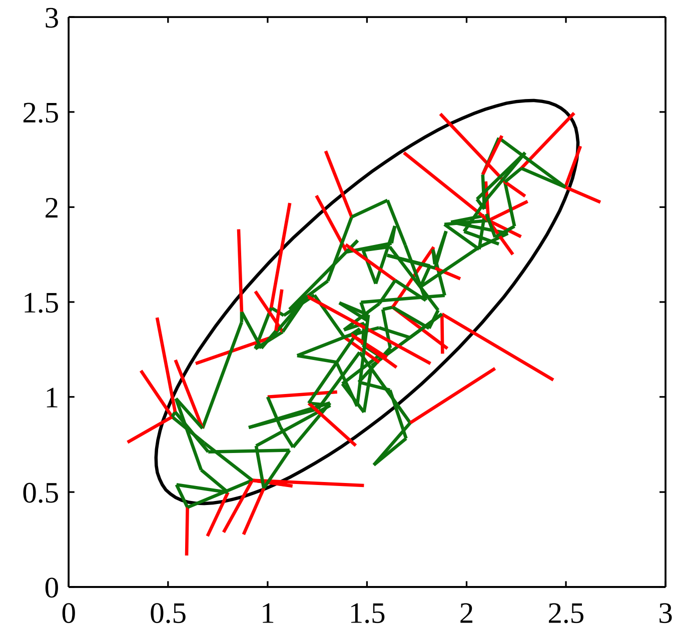
Implies that $P(x)$ is left invariant:

$$\sum_x T(x' \leftarrow x)\, P(x) \;=\; P(x') \cancelto{1}{\sum_x R(x \leftarrow x')}$$

Metropolis–Hastings

Arbitrary proposals $\sim Q$:

$$Q(x';x)\,P(x) \;\neq\; Q(x;x')\,P(x')$$



PRML, Bishop (2006)

Satisfies detailed balance by rejecting moves:

$$T(x'\!\leftarrow\!x) = \begin{cases} Q(x';x)\min\!\left(1,\ \frac{P(x')\,Q(x;x')}{P(x)\,Q(x';x)}\right) & x' \neq x \\ \dots & x' = x \end{cases}$$

Metropolis–Hastings

Transition operator

- Propose a move from the current state $Q(x'; x)$, e.g. $\mathcal{N}(x, \sigma^2)$
- Accept with probability $\min\!\left(1, \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right)$
- Otherwise next state in chain is a copy of current state

Notes

- Can use $P^* \propto P(x)$; normalizer cancels in acceptance ratio
- Satisfies detailed balance (shown below)
- Q must be chosen so chain is ergodic

$$\begin{aligned}P(x) \cdot T(x' \leftarrow x) &= P(x) \cdot Q(x'; x) \min\left(1, \frac{P(x')Q(x; x')}{P(x)Q(x'; x)}\right) = \min\Big(P(x)Q(x'; x),\ P(x')Q(x; x')\Big) \\ &= P(x') \cdot Q(x; x') \min\left(1, \frac{P(x)Q(x'; x)}{P(x')Q(x; x')}\right) = P(x') \cdot T(x \leftarrow x')\end{aligned}$$

Matlab/Octave code for demo

```
function samples = dumb_metropolis(init, log_ptilde, iters, sigma)

D = numel(init);
samples = zeros(D, iters);

state = init;
Lp_state = log_ptilde(state);
for ss = 1:iters
    % Propose
    prop = state + sigma*randn(size(state));
    Lp_prop = log_ptilde(prop);
    if log(rand) < (Lp_prop - Lp_state)
        % Accept
        state = prop;
        Lp_state = Lp_prop;
    end
    samples(:, ss) = state(:);
end
```

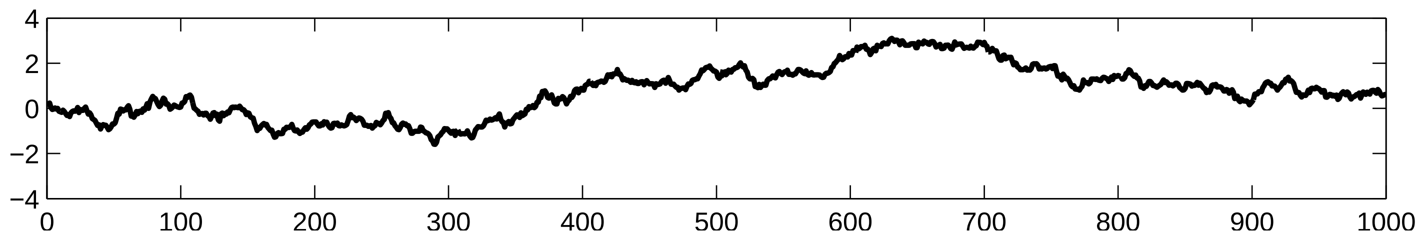
Step-size demo

Explore $\mathcal{N}(0,1)$ with different step sizes σ

```
sigma = @(s) plot(dumb_metropolis(0, @(x)-0.5*x*x, 1e3, s));
```

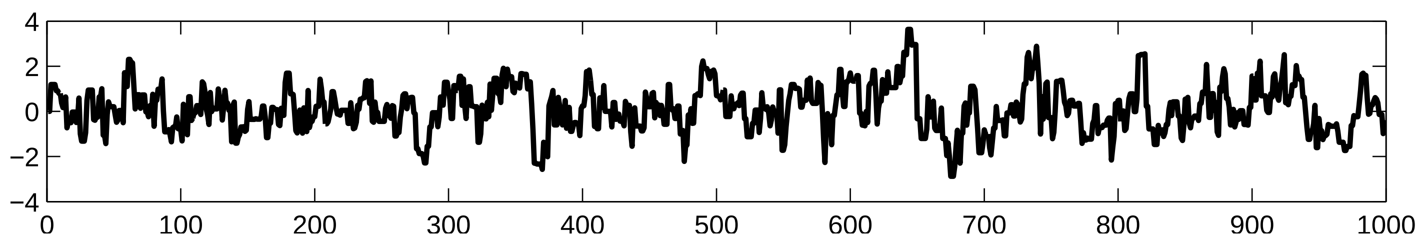
`sigma(0.1)`

99.8% accepts



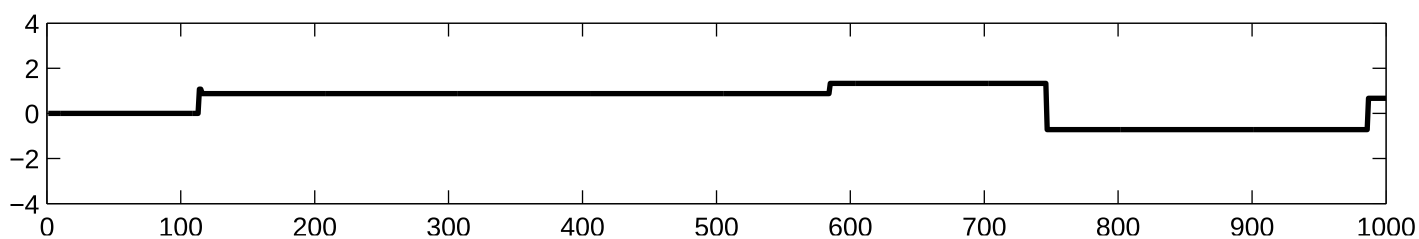
`sigma(1)`

68.4% accepts

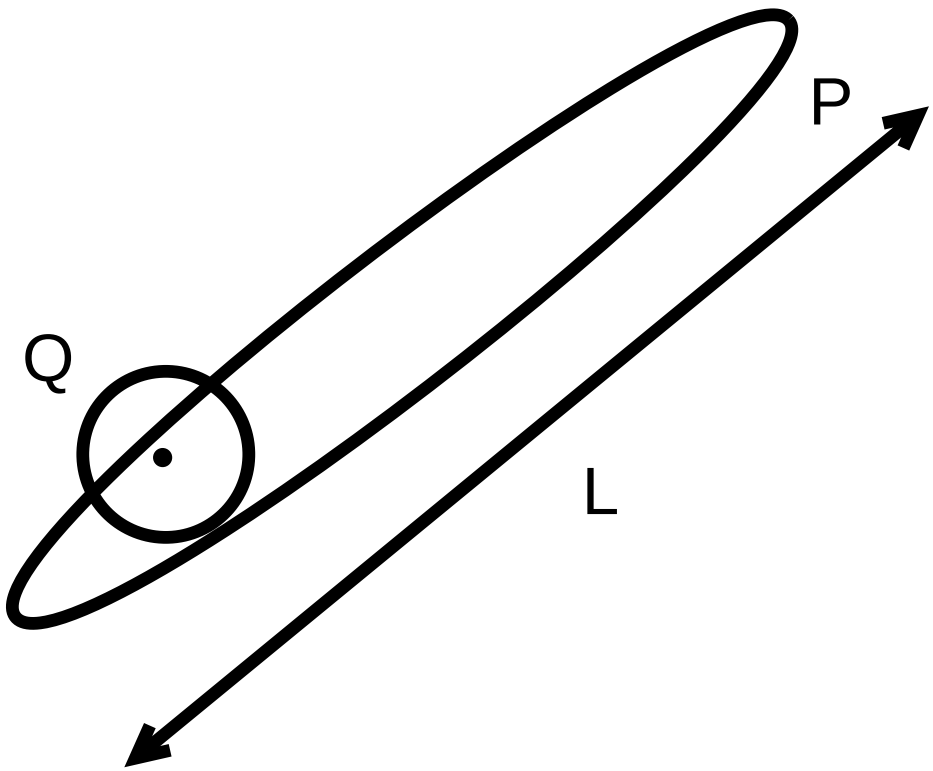


`sigma(100)`

0.5% accepts



Diffusion time



Generic proposals use
$Q(x'; x) = \mathcal{N}(x, \sigma^2)$

σ **large** $\rightarrow$ **many rejections**

σ **small** $\rightarrow$ **slow diffusion:**
$\sim (L/\sigma)^2$ iterations required

Adapted from MacKay (2003)

An MCMC strategy

Come up with good proposals $Q(x'; x)$

Combine transition operators:

$$
\begin{aligned}
x_1 &\sim T_A(\cdot \leftarrow x_0) \\
x_2 &\sim T_B(\cdot \leftarrow x_1) \\
x_3 &\sim T_C(\cdot \leftarrow x_2) \\
x_4 &\sim T_A(\cdot \leftarrow x_3) \\
x_5 &\sim T_B(\cdot \leftarrow x_4) \\
&\ldots
\end{aligned}
$$

Roadmap

- Probabilistic models
- Simple Monte Carlo
  Importance Sampling
- Markov chain Monte Carlo (MCMC)
  Gibbs sampling, M–H
- **Auxiliary variable methods**
  Swendsen–Wang, HMC

Auxiliary variables

The point of MCMC is to sum out variables, yet:

$$\int f(x)P(x)\,\mathrm{d}x = \int f(x)P(x,v)\,\mathrm{d}x\,\mathrm{d}v$$

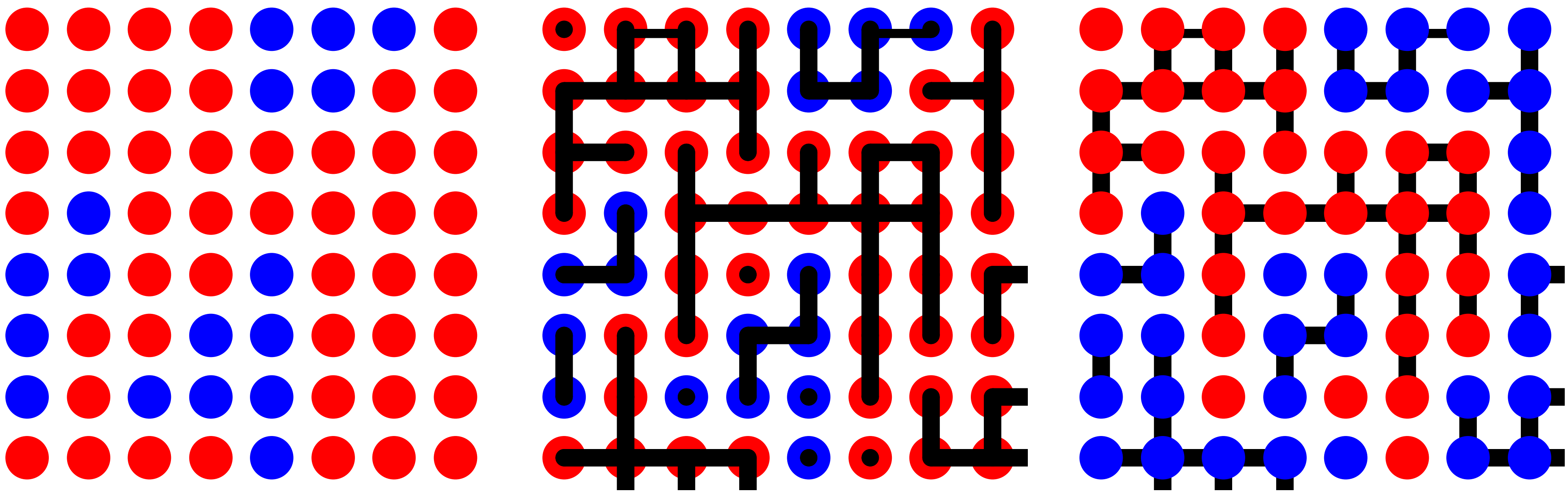
$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x,v \sim P(x,v)$$

We might want to introduce v if:

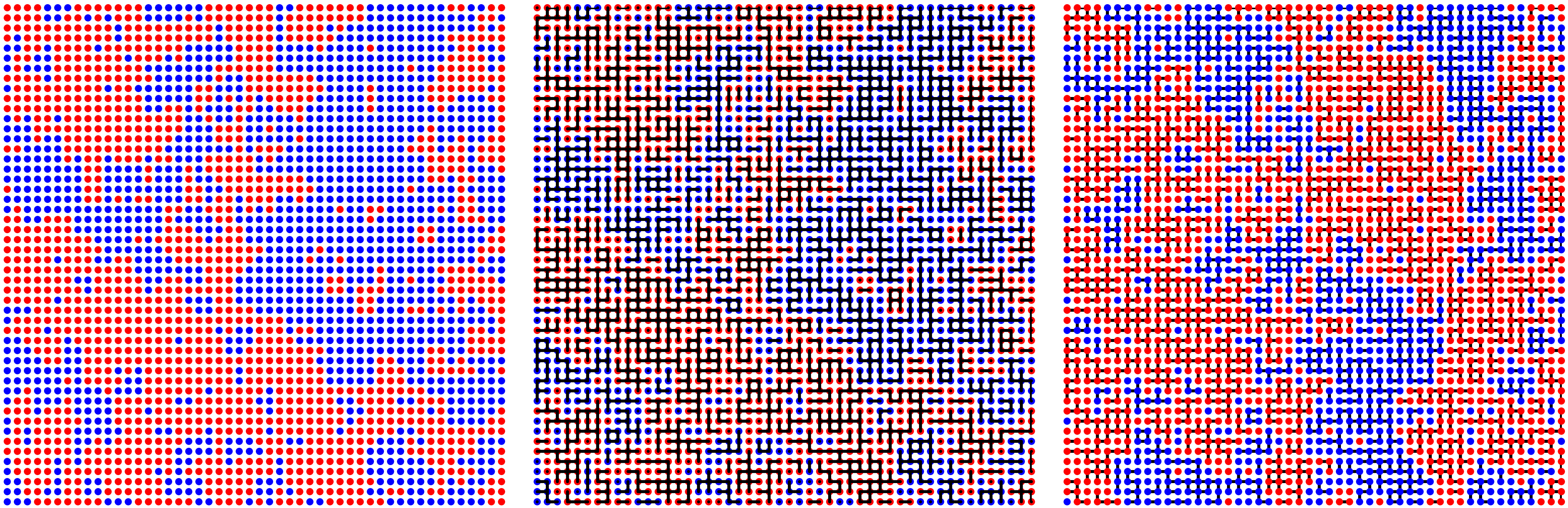
- $P(x\,|\,v)$ and $P(v\,|\,x)$ are simple (Cf RBMs, Martens and Sutskever 2010)
- $P(x,v)$ is otherwise easier to navigate

Swendsen–Wang (1987)

Seminal algorithm using auxiliary variables



Swendsen–Wang



Edwards and Sokal (1988) identified and generalized the "Fortuin-Kasteleyn-Swendsen-Wang" auxiliary variable joint distribution that underlies the algorithm.

Hamiltonian Monte Carlo (1987)

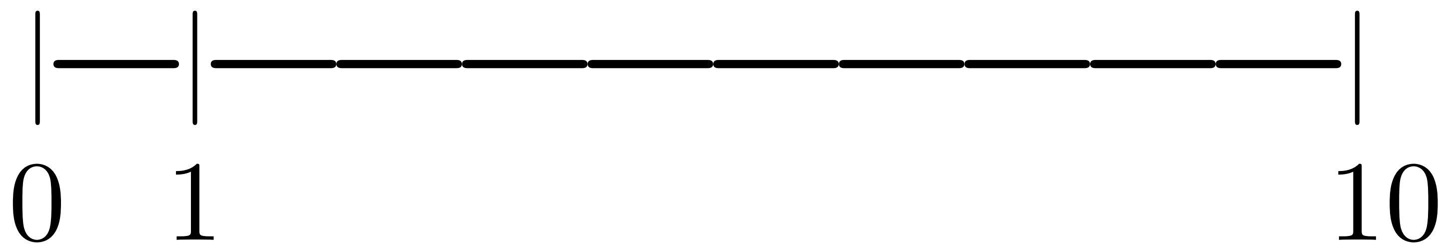
Define a joint distribution:

$$P(x, v) \propto e^{-E(x)} e^{-v^\top v/2} = e^{-H(x,v)}$$

Markov chain operators

- Gibbs sample velocity
- Simulate Hamiltonian dynamics
  - Conservation of energy means $P(x, v) = P(x', v')$
  - Metropolis acceptance probability is 1

Example / warning



Proposal: $\begin{cases} x_{t+1} = 9x_t + 1, & 0 < x_t < 1 \\ x_{t+1} = (x_t - 1)/9, & 1 < x_t < 10 \end{cases}$

Accept move with probability:

$$\min\left(1,\ \frac{P(x')\,Q(x;x')}{P(x)\,Q(x';x)}\right) = \min\left(1,\ \frac{P(x')}{P(x)}\right) \quad \text{(Wrong!)}$$

Leap-frog dynamics

a discrete approximation to Hamiltonian dynamics:

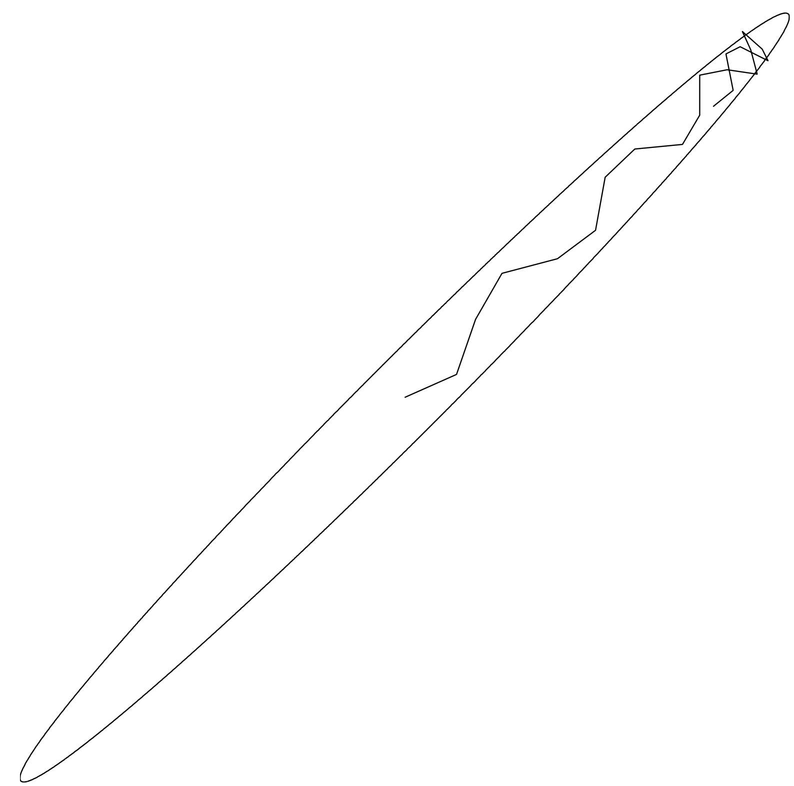
$$
\begin{aligned}
v_i(t + \tfrac{\epsilon}{2}) &= v_i(t) - \frac{\epsilon}{2}\,\frac{\partial E(x(t))}{\partial x_i} \\
x_i(t + \epsilon) &= x_i(t) + \epsilon\, v_i(t + \tfrac{\epsilon}{2}) \\
p_i(t + \epsilon) &= v_i(t + \tfrac{\epsilon}{2}) - \frac{\epsilon}{2}\,\frac{\partial E(x(t+\epsilon))}{\partial x_i}
\end{aligned}
$$

- H is not conserved
- Transformation has unit Jacobian
- Acceptance probability becomes $\min[1, \exp(H(v, x) - H(v', x'))]$

Hamiltonian Monte Carlo

The algorithm:

- Gibbs sample velocity $\sim \mathcal{N}(0, \mathbb{I})$
- Simulate L leapfrog steps
- Accept with probability $\min[1, \exp(H(v, x) - H(v', x'))]$



The original name is **Hybrid Monte Carlo**, with reference to the "hybrid" dynamical simulation method.

Hamiltonian dynamics

Recommended reading:

MCMC using Hamiltonian dynamics, Radford M. Neal, 2011.
Handbook of Markov Chain Monte Carlo
`http://www.cs.toronto.edu/~radford/ftp/ham-mcmc.pdf`

Recent developments include:

NUTS: No U-Turn Sampler
`http://arxiv.org/abs/1111.4246`

Riemann manifold Hamiltonian Monte Carlo
`http://www.dcs.gla.ac.uk/inference/rmhmc/`

Summary of auxiliary variables

— Swendsen–Wang
— Hamiltonian (Hybrid) Monte Carlo
— Slice sampling

Some of my auxiliary representation work:

Doubly-intractable distributions

Population methods for better mixing (on parallel hardware)

Being robust to bad random number generators

Slice-sampling hierarchical latent Gaussian models

Overview

— Probabilistic models

— Simple Monte Carlo

Importance Sampling

— Markov chain Monte Carlo (MCMC)

Gibbs sampling, M–H

— Auxiliary variable methods

Swendsen–Wang, HMC

Appendix slides

Finding $P(x_i\!=\!1)$

Method 1: fraction of time $x_i\!=\!1$

$$P(x_i\!=\!1) = \sum_{x_i} \mathbb{I}(x_i\!=\!1)P(x_i) \approx \frac{1}{S}\sum_{s=1}^{S} \mathbb{I}(x_i^{(s)}), \quad x_i^{(s)} \sim P(x_i)$$

Method 2: average of $P(x_i\!=\!1|\mathbf{x}_{\setminus i})$

$$\begin{aligned} P(x_i\!=\!1) &= \sum_{\mathbf{x}_{\setminus i}} P(x_i\!=\!1|\mathbf{x}_{\setminus i})P(\mathbf{x}_{\setminus i}) \\ &\approx \frac{1}{S}\sum_{s=1}^{S} P(x_i = 1|\mathbf{x}_{\setminus i}^{(s)}), \quad \mathbf{x}_{\setminus i}^{(s)} \sim P(\mathbf{x}_{\setminus i}) \end{aligned}$$

Example of "Rao-Blackwellization".

More generally

This is easy

$$I = \sum_{\mathbf{x}} f(x_i) P(\mathbf{x}) \approx \frac{1}{S} \sum_{s=1}^{S} f(x_i^{(s)}), \quad \mathbf{x}^{(s)} \sim P(\mathbf{x})$$

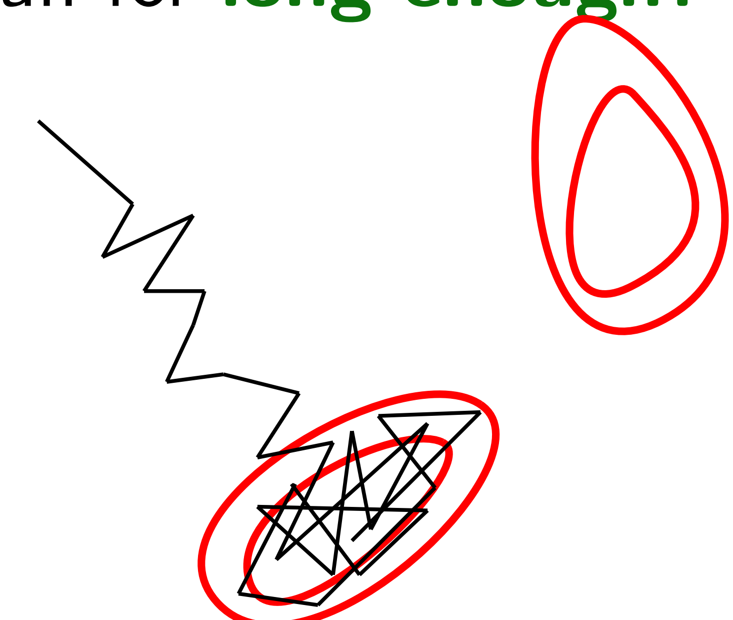
But this might be better

$$I = \sum_{\mathbf{x}} f(x_i) P(x_i | \mathbf{x}_{\setminus i}) P(\mathbf{x}_{\setminus i}) = \sum_{\mathbf{x}_{\setminus i}} \left(\sum_{x_i} f(x_i) P(x_i | \mathbf{x}_{\setminus i}) \right) P(\mathbf{x}_{\setminus i})$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} \left(\sum_{x_i} f(x_i) P(x_i | \mathbf{x}_{\setminus i}^{(s)}) \right), \quad \mathbf{x}_{\setminus i}^{(s)} \sim P(\mathbf{x}_{\setminus i})$$

How should we run MCMC?

- The samples aren't independent. Should we **thin**, only keep every Kth sample?
- Arbitrary initialization means starting iterations are bad. Should we discard a **"burn-in" period**?
- Maybe we should perform **multiple runs?**
- How do we know if we have run for **long enough?**



Forming estimates

Can *thin* samples so approximately independent.

But, **can use all samples.**

The simple Monte Carlo estimator is still:

— consistent

— unbiased if the chain has "burned in"

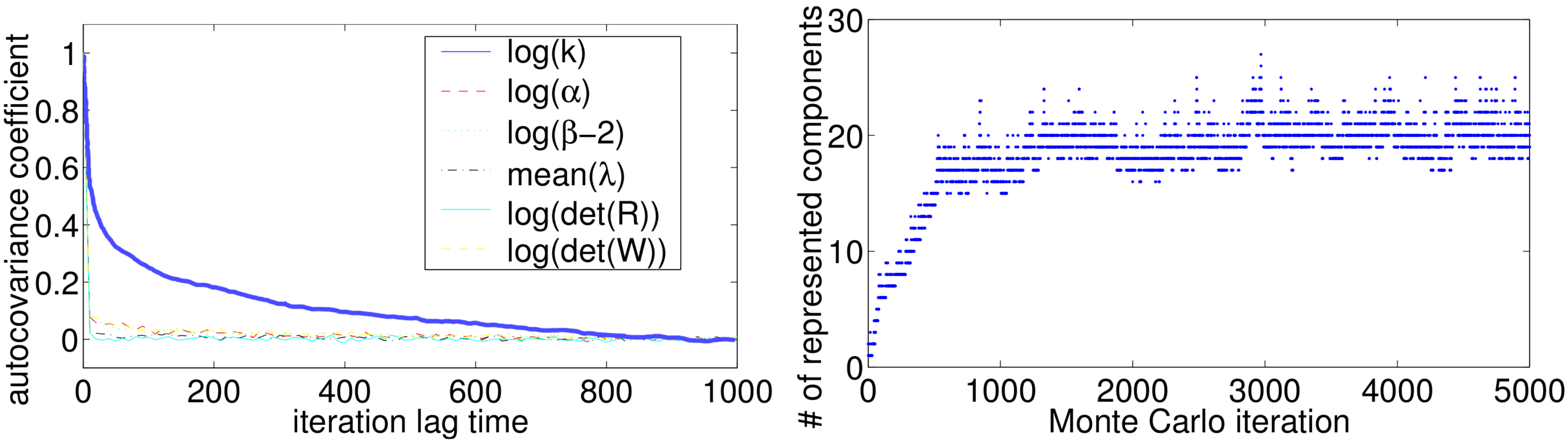
The correct motivation to thin:

if computing $f(\mathbf{x}^{(s)})$ is expensive

In some special circumstances strategic thinning can help.

Steven N. MacEachern and Mario Peruggia, Statistics & Probability Letters, 47(1):91–98, 2000.
`http://dx.doi.org/10.1016/S0167-7152(99)00142-X` — Thanks to Simon Lacoste-Julien for the reference.

Empirical diagnostics



Rasmussen (2000)

Recommendations

Diagnostic software: `R-CODA`

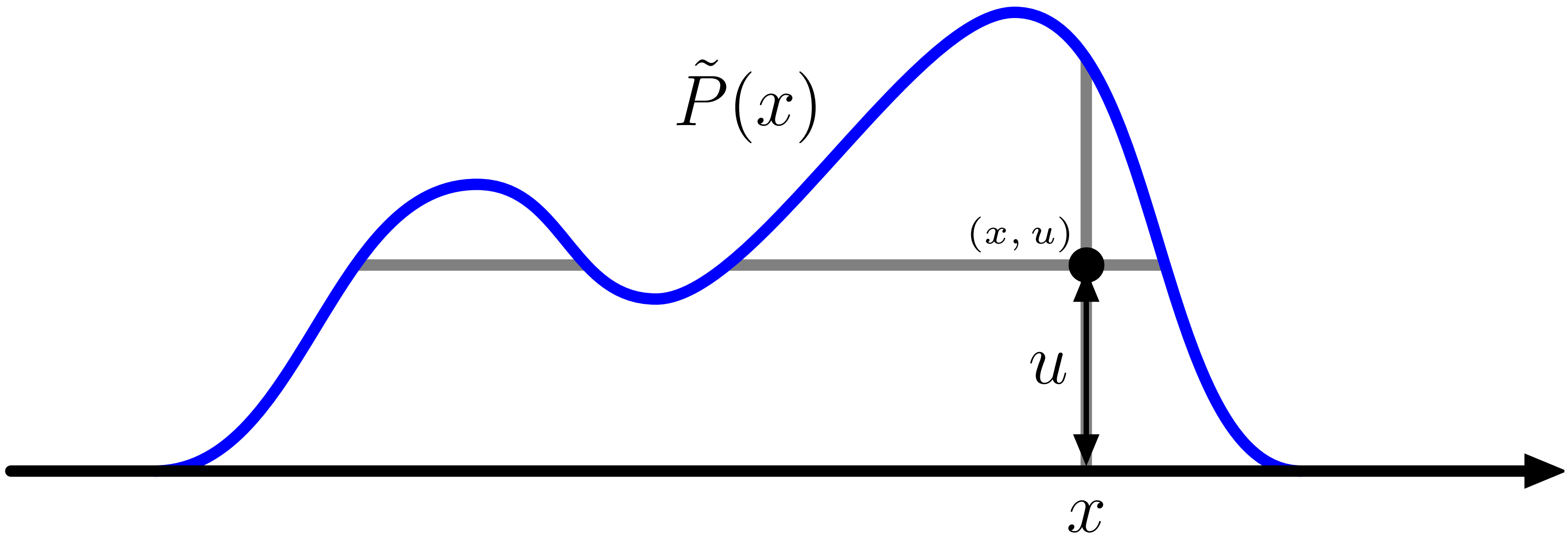
For opinion on thinning, multiple runs, burn in, etc.

Charles J. Geyer, *Statistical Science*. 7(4):473–483, 1992.

`http://www.jstor.org/stable/2246094`

Slice sampling idea

Sample point uniformly under curve $\tilde{P}(x) \propto P(x)$

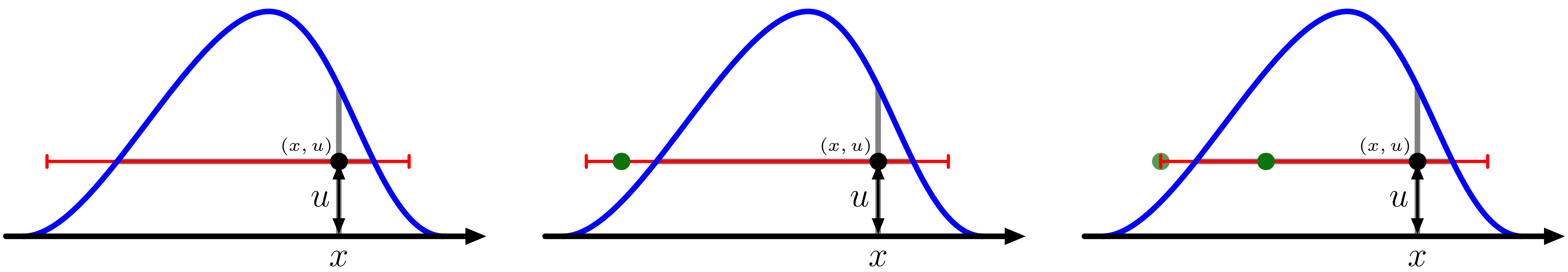


$$p(u|x) = \text{Uniform}[0, \tilde{P}(x)]$$

$$p(x|u) \propto \begin{cases} 1 & \tilde{P}(x) \geq u \\ 0 & \text{otherwise} \end{cases} = \text{“Uniform on the slice”}$$

Slice sampling

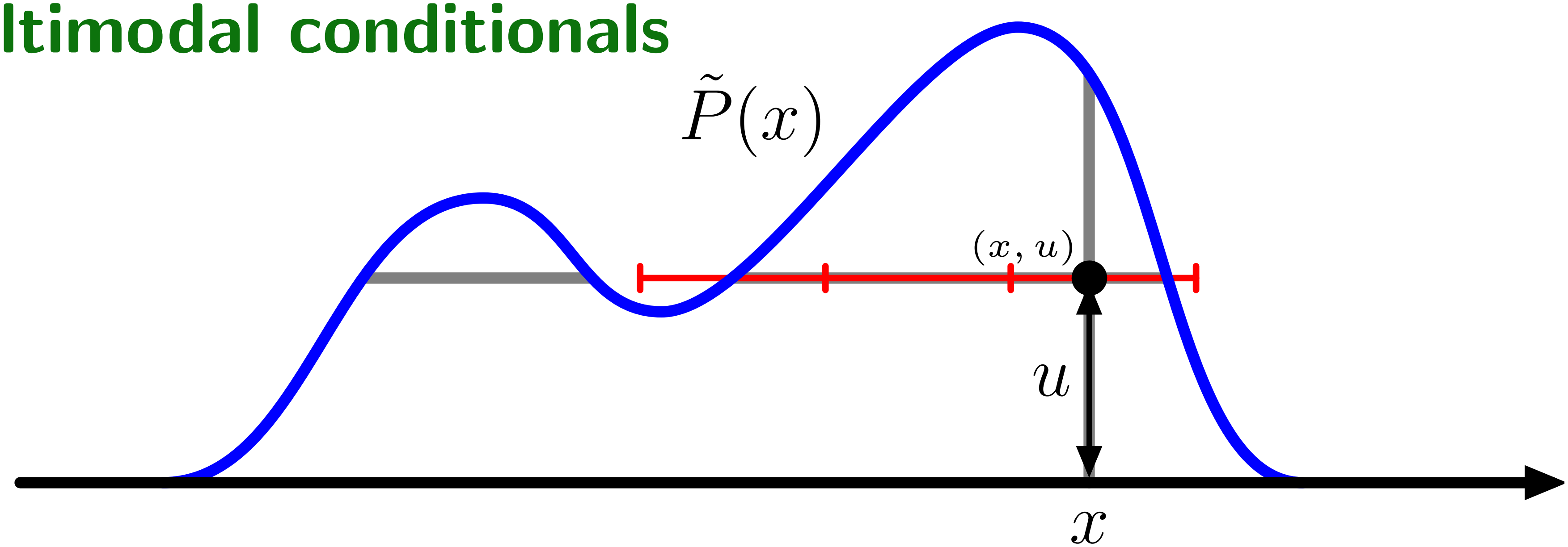
Unimodal conditionals



- bracket slice
- sample uniformly within bracket
- shrink bracket if $\tilde{P}(x) < u$ (off slice)
- accept first point on the slice

Slice sampling

Multimodal conditionals



- place bracket randomly around point
- linearly step out until bracket ends are off slice
- sample on bracket, shrinking as before

Satisfies detailed balance, leaves $p(x|u)$ invariant

Slice sampling

Advantages of slice-sampling:

- Easy — only require $\tilde{P}(x) \propto P(x)$
- No rejections
- Tweak params not too important

There are more advanced versions.
Neal (2003) contains *many* ideas.

References

Further reading (1/2)

General references:

Probabilistic inference using Markov chain Monte Carlo methods, Radford M. Neal, Technical report: CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993. `http://www.cs.toronto.edu/~radford/review.abstract.html`

Various figures and more came from (see also references therein):

Advances in Markov chain Monte Carlo methods. Iain Murray. 2007. `http://www.cs.toronto.edu/~murray/pub/07thesis/`

Information theory, inference, and learning algorithms. David MacKay, 2003. `http://www.inference.phy.cam.ac.uk/mackay/itila/`

Pattern recognition and machine learning. Christopher M. Bishop. 2006. `http://research.microsoft.com/~cmbishop/PRML/`

Specific points:

If you do Gibbs sampling with continuous distributions this method, which I omitted for material-overload reasons, may help:

Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation, Radford M. Neal, *Learning in graphical models*, M. I. Jordan (editor), 205–228, Kluwer Academic Publishers, 1998. `http://www.cs.toronto.edu/~radford/overk.abstract.html`

An example of picking estimators carefully:

Speed-up of Monte Carlo simulations by sampling of rejected states, Frenkel, D, *Proceedings of the National Academy of Sciences*, 101(51):17571–17575, The National Academy of Sciences, 2004. `http://www.pnas.org/cgi/content/abstract/101/51/17571`

A key reference for auxiliary variable methods is:

Generalizations of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm, Robert G. Edwards and A. D. Sokal, *Physical Review*, 38:2009–2012, 1988.

Slice sampling, Radford M. Neal, *Annals of Statistics*, 31(3):705–767, 2003. `http://www.cs.toronto.edu/~radford/slice-aos.abstract.html`

Bayesian training of backpropagation networks by the hybrid Monte Carlo method, Radford M. Neal, Technical report: CRG-TR-92-1, Connectionist Research Group, University of Toronto, 1992. `http://www.cs.toronto.edu/~radford/bbp.abstract.html`

An early reference for parallel tempering:

Markov chain Monte Carlo maximum likelihood, Geyer, C. J, *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, 156–163, 1991.

Sampling from multimodal distributions using tempered transitions, Radford M. Neal, *Statistics and Computing*, 6(4):353–366, 1996.

Further reading (2/2)

Software:

Gibbs sampling for graphical models: `http://mathstat.helsinki.fi/openbugs/` `http://www-ice.iarc.fr/~martyn/software/jags/`
Neural networks and other flexible models: `http://www.cs.utoronto.ca/~radford/fbm.software.html`

CODA: http://www-fis.iarc.fr/coda/

Other Monte Carlo methods:

Nested sampling is a new Monte Carlo method with some interesting properties:
Nested sampling for general Bayesian computation, John Skilling, *Bayesian Analysis*, 2006.
(to appear, posted online June 5). `http://ba.stat.cmu.edu/journal/forthcoming/skilling.pdf`

Approaches based on the "multi-canonicle ensemble" also solve some of the problems with traditional tempterature-based methods:
Multicanonical ensemble: a new approach to simulate first-order phase transitions, Bernd A. Berg and Thomas Neuhaus, *Phys. Rev. Lett*, 68(1):9–12, 1992. `http://prola.aps.org/abstract/PRL/v68/i1/p9_1`

A good review paper:
Extended Ensemble Monte Carlo. Y Iba. Int J Mod Phys C [Computational Physics and Physical Computation] 12(5):623-656. 2001.

Particle filters / Sequential Monte Carlo are famously successful in time series modeling, but are more generally applicable.
This may be a good place to start: `http://www.cs.ubc.ca/~arnaud/journals.html`

Exact or perfect sampling uses Markov chain simulation but suffers no initialization bias. An amazing feat when it can be performed:
Annotated bibliography of perfectly random sampling with Markov chains, David B. Wilson
`http://dbwilson.com/exact/`

MCMC does not apply to *doubly-intractable* distributions. For what that even means and possible solutions see:
An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants, J. Møller, A. N. Pettitt, R. Reeves and K. K. Berthelsen, *Biometrika*, 93(2):451–458, 2006.
MCMC for doubly-intractable distributions, Iain Murray, Zoubin Ghahramani and David J. C. MacKay, *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Rina Dechter and Thomas S. Richardson (editors), 359–366, AUAI Press, 2006.
`http://www.gatsby.ucl.ac.uk/~iam23/pub/06doubly_intractable/doubly_intractable.pdf`