Markov chain Monte Carlo

Machine Learning Summer School 2009

http://mlg.eng.cam.ac.uk/mlss09/

Iain Murray

http://www.cs.toronto.edu/~murray/

A statistical problem

What is the average height of the MLSS lecturers?
Method: measure their heights, add them up and divide by $N = 20$.

What is the average height f of people p in Cambridge $\mathcal{C}$?

$$E_{p \in \mathcal{C}}[f(p)] \equiv \frac{1}{|\mathcal{C}|} \sum_{p \in \mathcal{C}} f(p), \quad \text{“intractable”?}$$

$$\approx \frac{1}{S} \sum_{s=1}^{S} f\left(p^{(s)}\right), \quad \text{for random survey of } S \text{ people } \{p^{(s)}\} \in \mathcal{C}$$

Surveying works for large and notionally infinite populations.

Simple Monte Carlo

Statistical sampling can be applied to any expectation:

In general:

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

Example: making predictions

$$p(x|\mathcal{D}) = \int P(x|\theta,\mathcal{D})P(\theta|\mathcal{D})\,\mathrm{d}\theta$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} P(x|\theta^{(s)},\mathcal{D}), \quad \theta^{(s)} \sim P(\theta|\mathcal{D})$$

More examples: E-step statistics in EM, Boltzmann machine learning

Properties of Monte Carlo

Estimator: $\int f(x)P(x)\,\mathrm{d}x \approx \hat{f} \equiv \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim P(x)$

Estimator is unbiased:

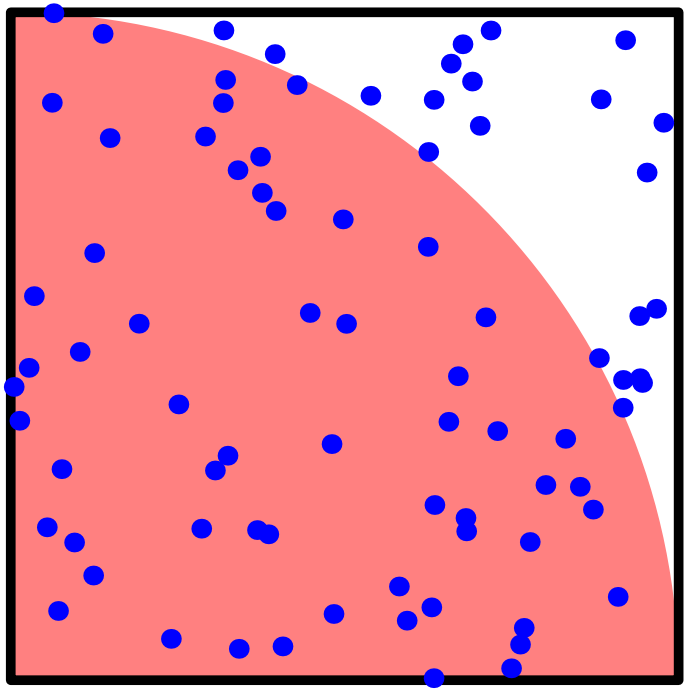
$$\mathbb{E}_{P(\{x^{(s)}\})}\left[\hat{f}\right] = \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}_{P(x)}[f(x)] = \mathbb{E}_{P(x)}[f(x)]$$

Variance shrinks $\propto 1/S$:

$$\mathrm{var}_{P(\{x^{(s)}\})}\left[\hat{f}\right] = \frac{1}{S^2}\sum_{s=1}^{S}\mathrm{var}_{P(x)}[f(x)] = \mathrm{var}_{P(x)}[f(x)]\,/S$$

"Error bars" shrink like $\sqrt{S}$

A dumb approximation of π



$$P(x,y) = \begin{cases} 1 & 0<x<1 \text{ and } 0<y<1 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = 4 \iint \mathbb{I}\left((x^2+y^2)<1\right) P(x,y)\, \mathrm{d}x\, \mathrm{d}y$$

```
octave:1> S=12; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.3333
octave:2> S=1e7; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.1418
```

Aside: don't always sample!

> *"Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse."*
>
> — Alan Sokal, 1996

Example: numerical solutions to (nice) 1D integrals are fast

```
octave:1> 4 * quadl(@(x) sqrt(1-x.^2), 0, 1, tolerance)
```

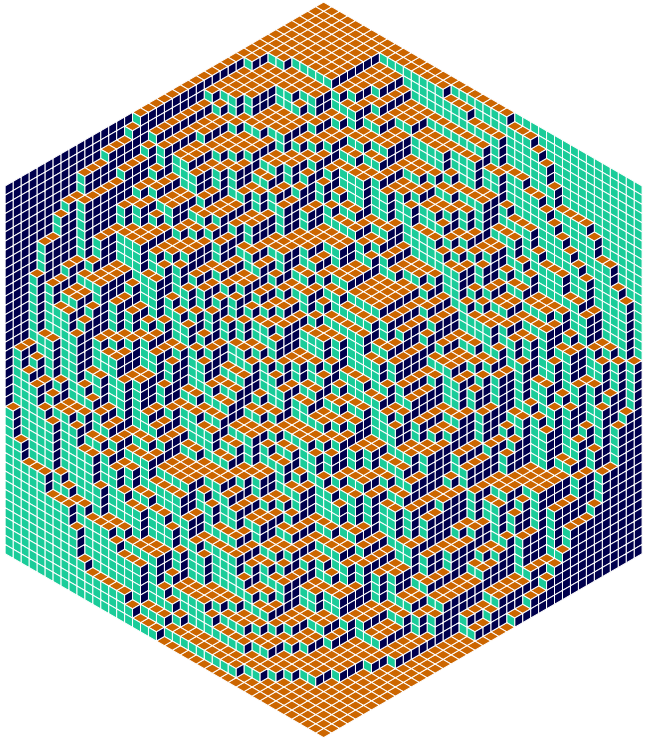
Gives π to 6 dp's in 108 evaluations, machine precision in 2598.

(NB Matlab's `quadl` fails at zero tolerance)

Other lecturers are covering alternatives for higher dimensions.

No approx. integration method always works. Sometimes Monte Carlo is the best.

Eye-balling samples

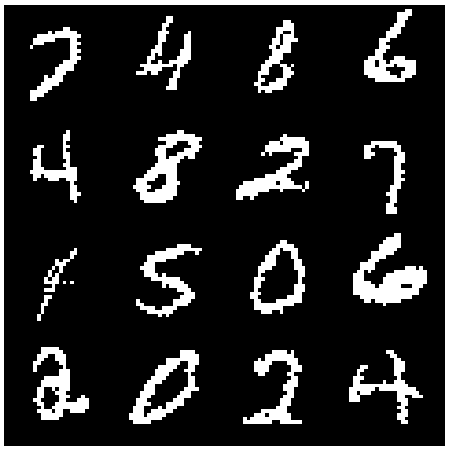


Sometimes samples are pleasing to look at:

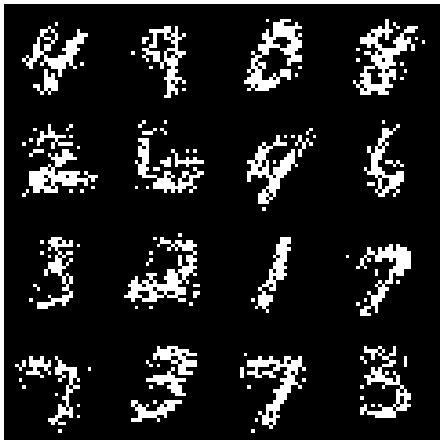
(if you're into geometrical combinatorics)

Figure by Propp and Wilson. Source: MacKay textbook.

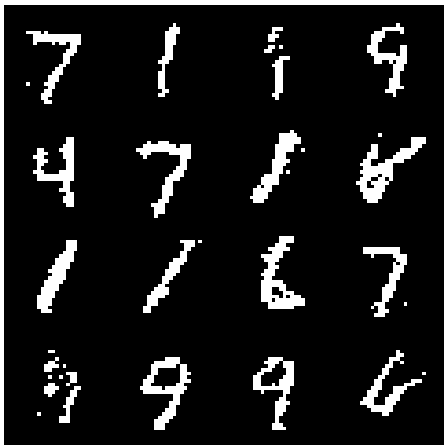
Sanity check probabilistic modelling assumptions:



Data samples

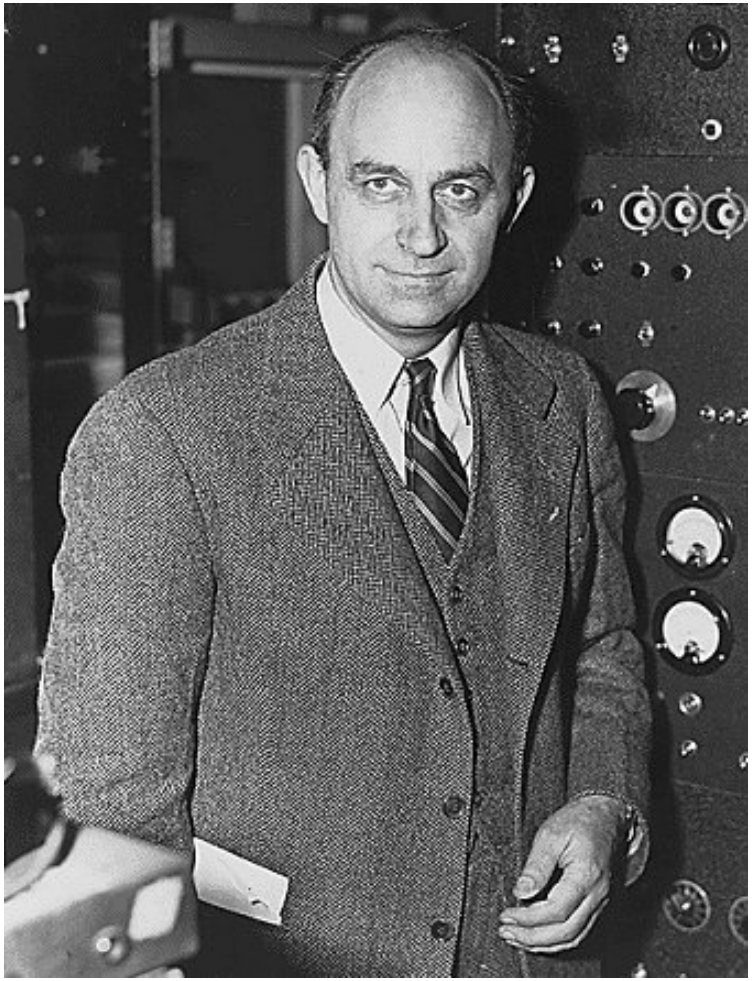


MoB samples



RBM samples

Monte Carlo and Insomnia



Enrico Fermi (1901–1954) took great delight in astonishing his colleagues with his remakably accurate predictions of experimental results. . . he revealed that his "guesses" were really derived from the statistical sampling techniques that he used to calculate with whenever insomnia struck in the wee morning hours!

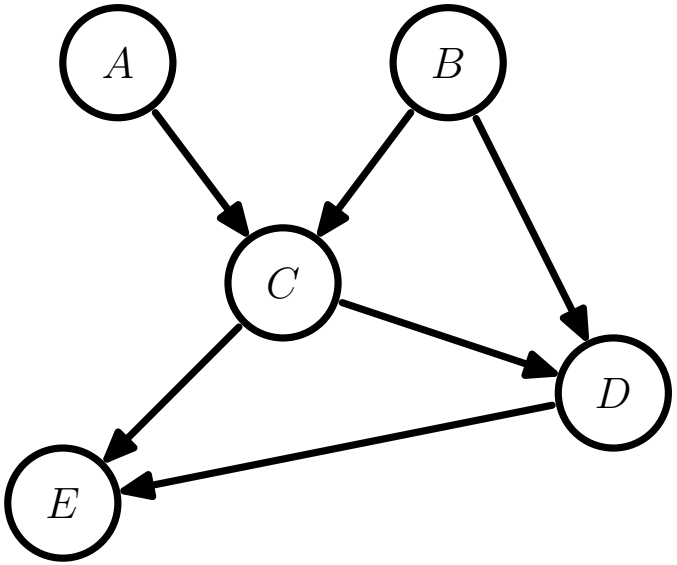
—*The beginning of the Monte Carlo method*,
N. Metropolis

Sampling from a Bayes net

Ancestral pass for directed graphical models:

— sample each top level variable from its marginal

— sample each other node from its conditional once its parents have been sampled

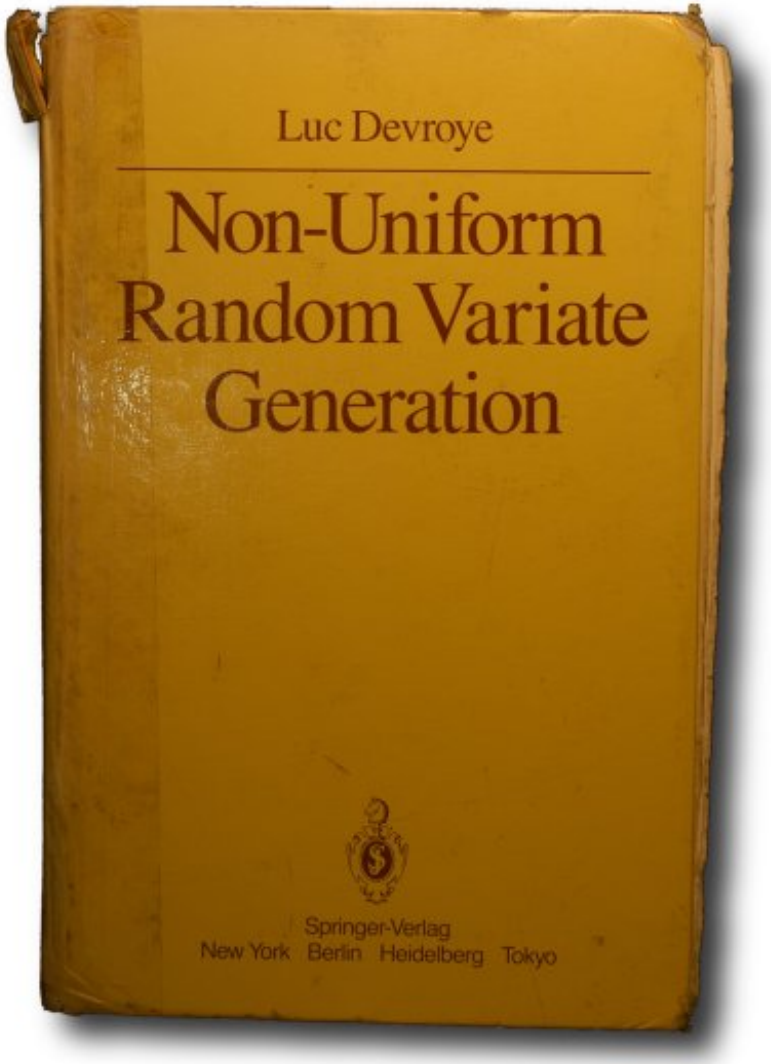


Sample:

$$A \sim P(A)$$
$$B \sim P(B)$$
$$C \sim P(C \mid A, B)$$
$$D \sim P(D \mid B, C)$$
$$E \sim P(D \mid C, D)$$

$$P(A, B, C, D, E) = P(A)\, P(B)\, P(C \mid A, B)\, P(D \mid B, C)\, P(E \mid C, D)$$

Sampling the conditionals



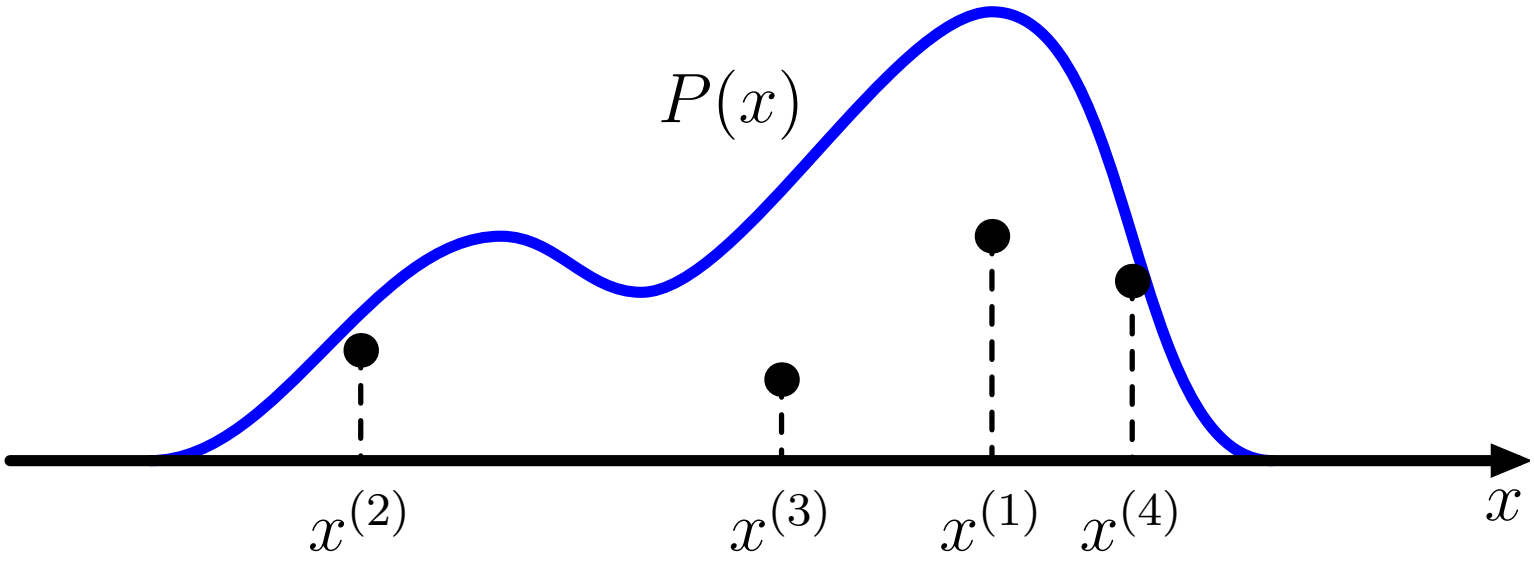
Use library routines for univariate distributions
(and some other special cases)

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

`http://cg.scs.carleton.ca/~luc/rnbookindex.html`

Sampling from distributions

Draw points uniformly under the curve:



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

Sampling from distributions

How to convert samples from a Uniform[0,1] generator:

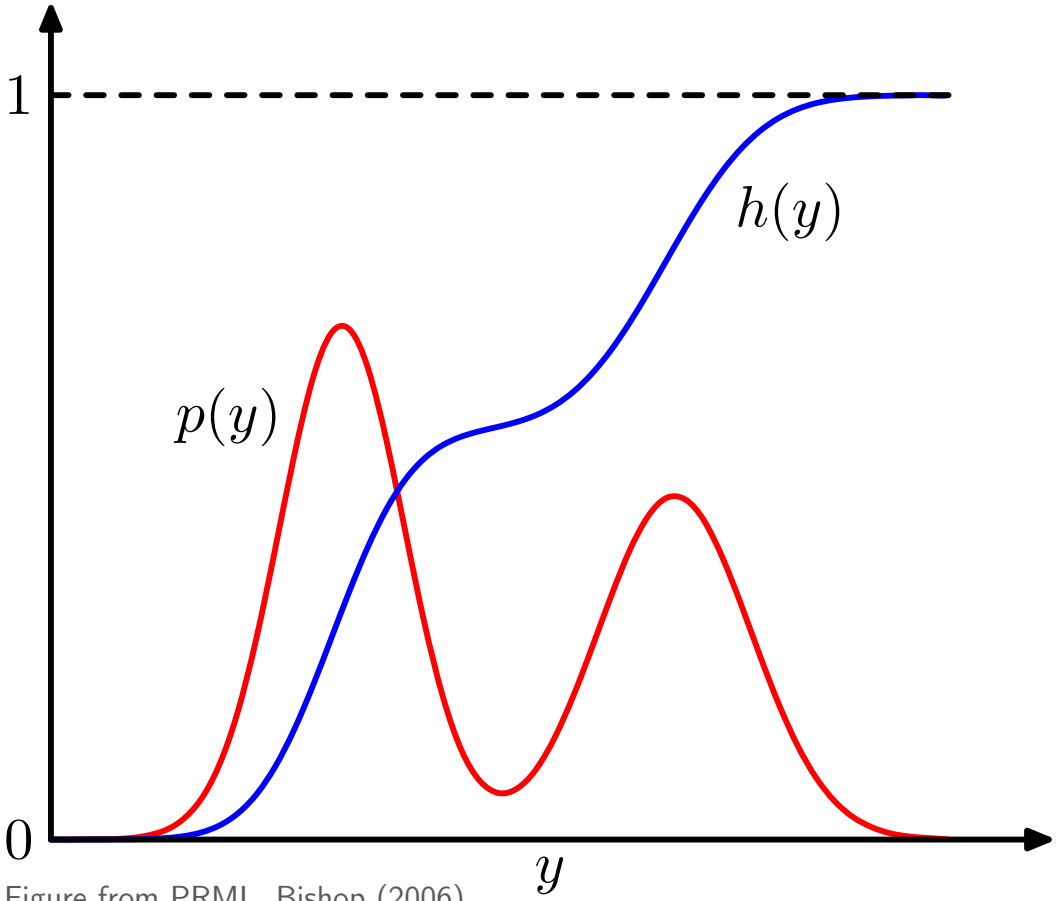


Figure from PRML, Bishop (2006)

$$h(y) = \int_{-\infty}^{y} p(y') \, \mathrm{d}y'$$

Draw mass to left of point:

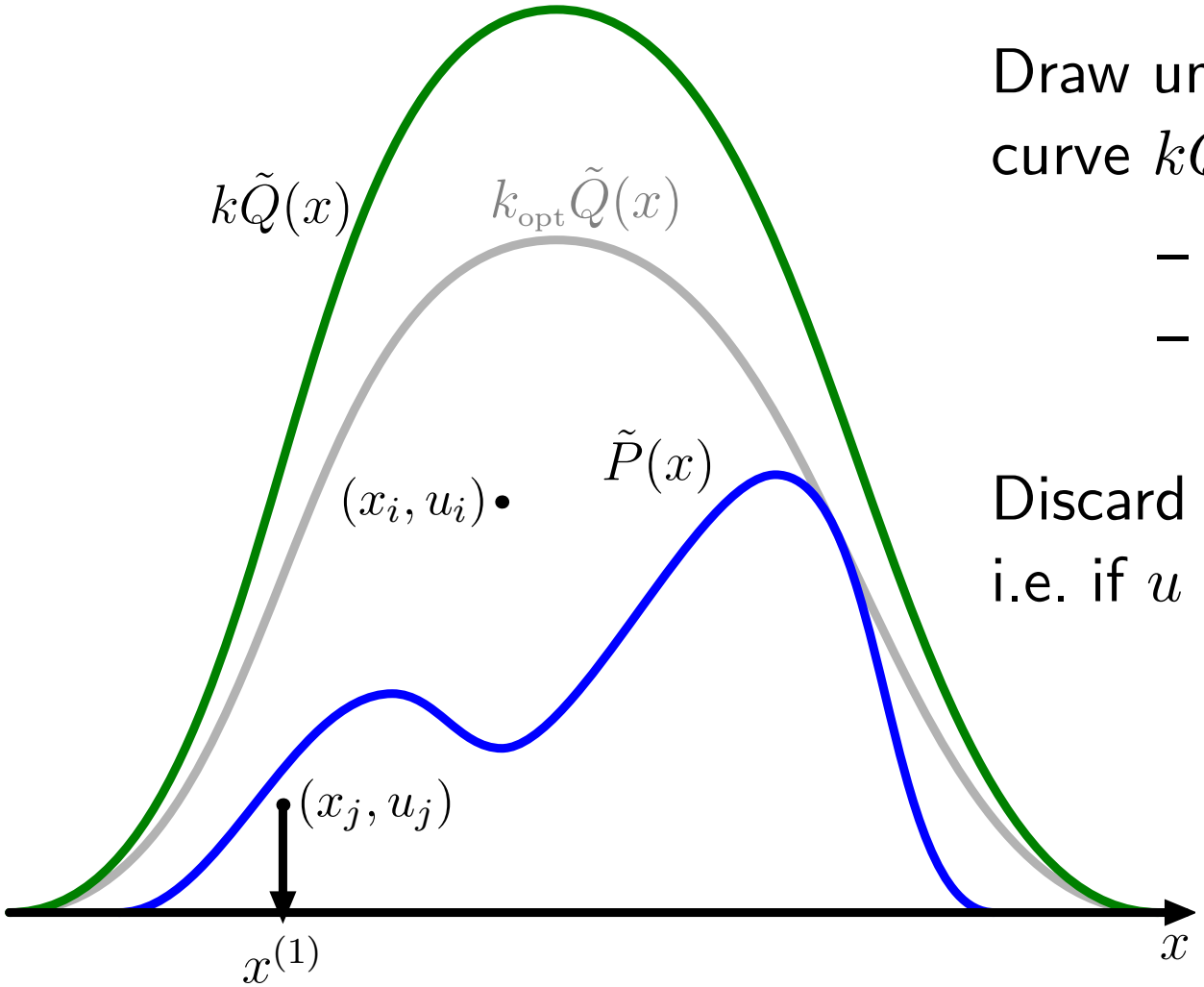
$u \sim$ Uniform[0,1]

Sample, $y(u) = h^{-1}(u)$

Although we can't always compute and invert $h(y)$

Rejection sampling

Sampling underneath a $\tilde{P}(x) \propto P(x)$ curve is also valid



Draw underneath a simple curve $k\tilde{Q}(x) \geq \tilde{P}(x)$:

- Draw $x \sim Q(x)$
- height $u \sim \text{Uniform}[0, k\tilde{Q}(x)]$

Discard the point if above $\tilde{P}$, i.e. if $u > \tilde{P}(x)$

Importance sampling

Computing $\tilde{P}(x)$ and $\tilde{Q}(x)$, then *throwing* x *away* seems wasteful
Instead rewrite the integral as an expectation under Q:

$$\int f(x)P(x)\,\mathrm{d}x = \int f(x)\frac{P(x)}{Q(x)}Q(x)\,\mathrm{d}x, \qquad (Q(x) > 0 \text{ if } P(x) > 0)$$

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

This is just simple Monte Carlo again, so it is unbiased.

Importance sampling applies when the integral is not an expectation.
Divide and multiply any integrand by a convenient distribution.

Importance sampling (2)

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

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

$$\approx \frac{1/\cancel{S}}{1/\cancel{S}\sum_{s'}\tilde{r}^{(s')}}\sum_{s=1}^{S} f(x^{(s)})\tilde{r}^{(s)} \equiv \sum_{s=1}^{S} f(x^{(s)})w^{(s)}$$

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

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

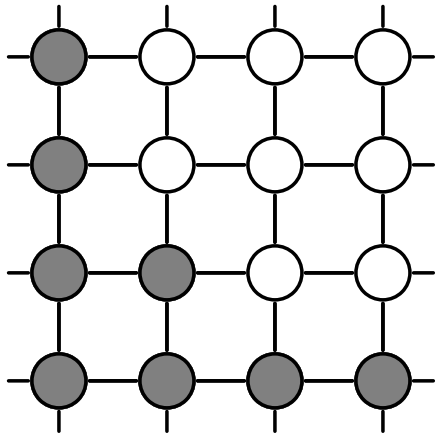
Summary so far

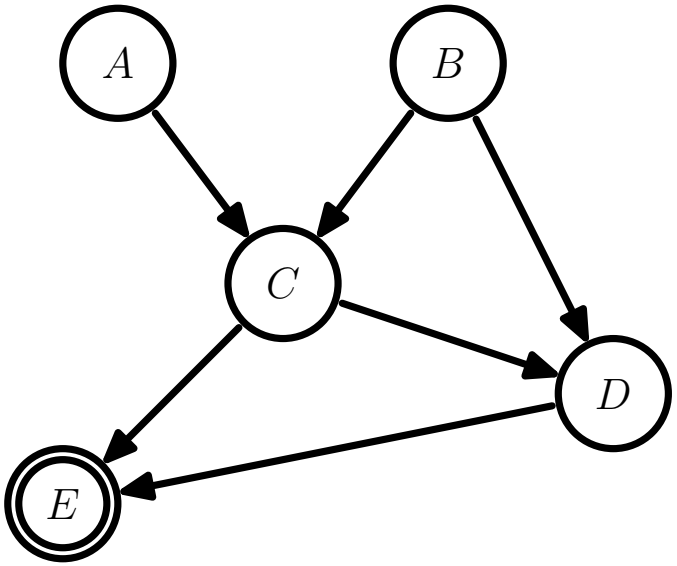
- Sums and integrals, often expectations, occur frequently in statistics
- **Monte Carlo** approximates expectations with a sample average
- **Rejection sampling** draws samples from complex distributions
- **Importance sampling** applies Monte Carlo to 'any' sum/integral

Application to large problems

We often can't decompose $P(X)$ into low-dimensional conditionals

Undirected graphical models: $P(x) = \frac{1}{\mathcal{Z}} \prod_i f_i(x)$





Posterior of a directed graphical model

$$P(A, B, C, D \,|\, E) = \frac{P(A, B, C, D, E)}{P(E)}$$

We often don't know $\mathcal{Z}$ or $P(E)$

Application to large problems

Rejection & importance sampling scale badly with dimensionality

Example:

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

Rejection sampling:

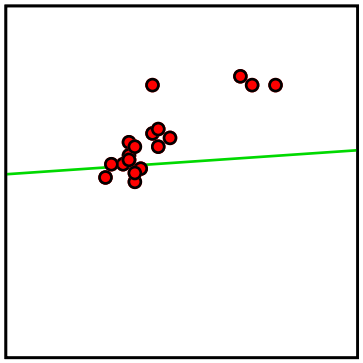
Requires $\sigma \geq 1$. Fraction of proposals accepted $= \sigma^{-D}$

Importance sampling:

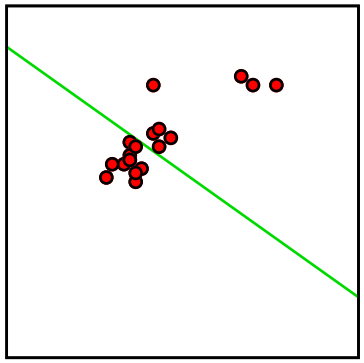
Variance of importance weights $= \left(\frac{\sigma^2}{2 - 1/\sigma^2} \right)^{D/2} - 1$

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

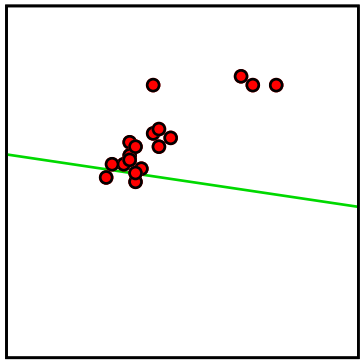
Importance sampling weights



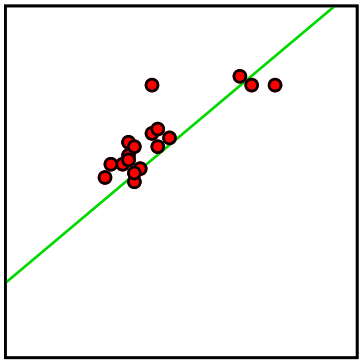
w =0.00548



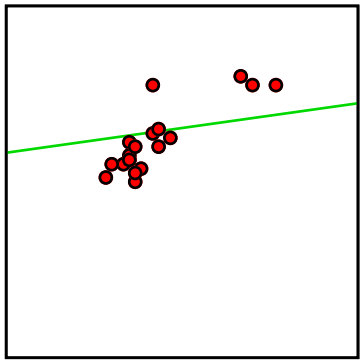
w =1.59e-08



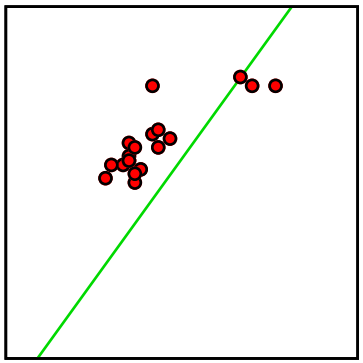
w =9.65e-06



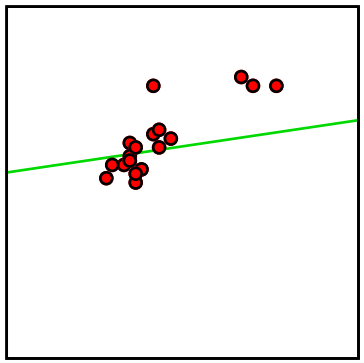
w =0.371



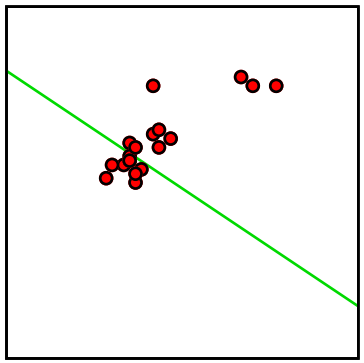
w =0.103



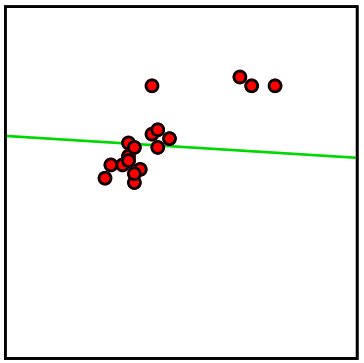
w =1.01e-08



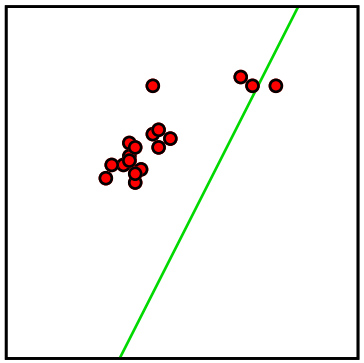
w =0.111



w =1.92e-09

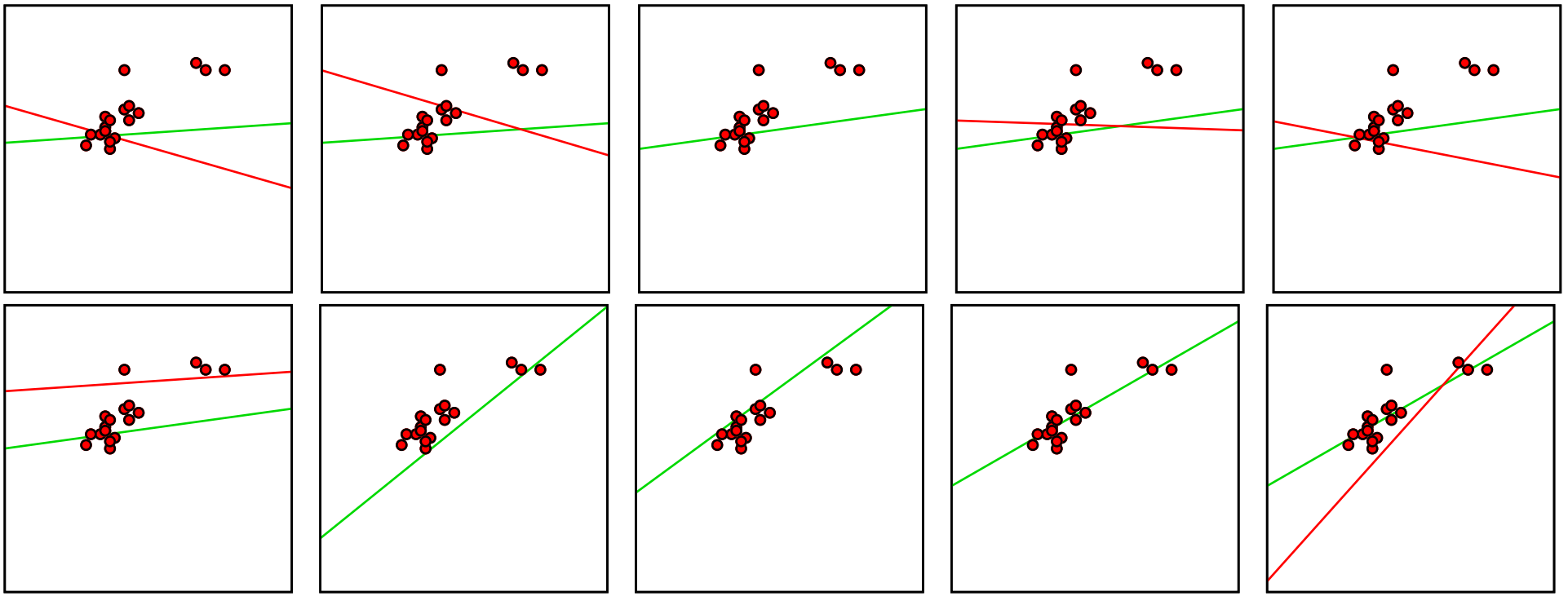


w =0.0126

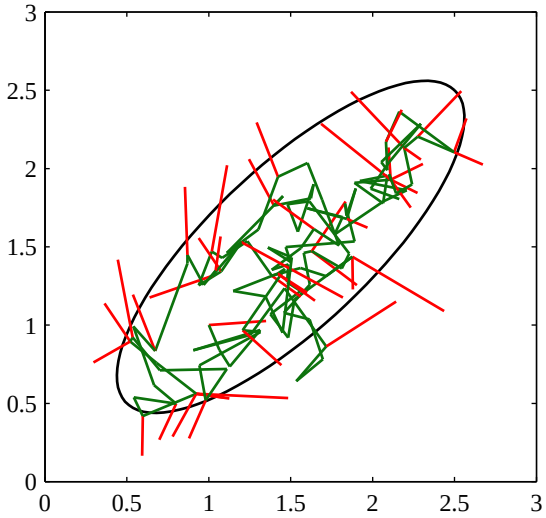


w =1.1e-51

Metropolis algorithm



- Perturb parameters: $Q(\theta'; \theta)$, e.g. $\mathcal{N}(\theta, \sigma^2)$
- Accept with probability $\min\left(1, \dfrac{\tilde{P}(\theta'|\mathcal{D})}{\tilde{P}(\theta|\mathcal{D})}\right)$
- Otherwise **keep old parameters**



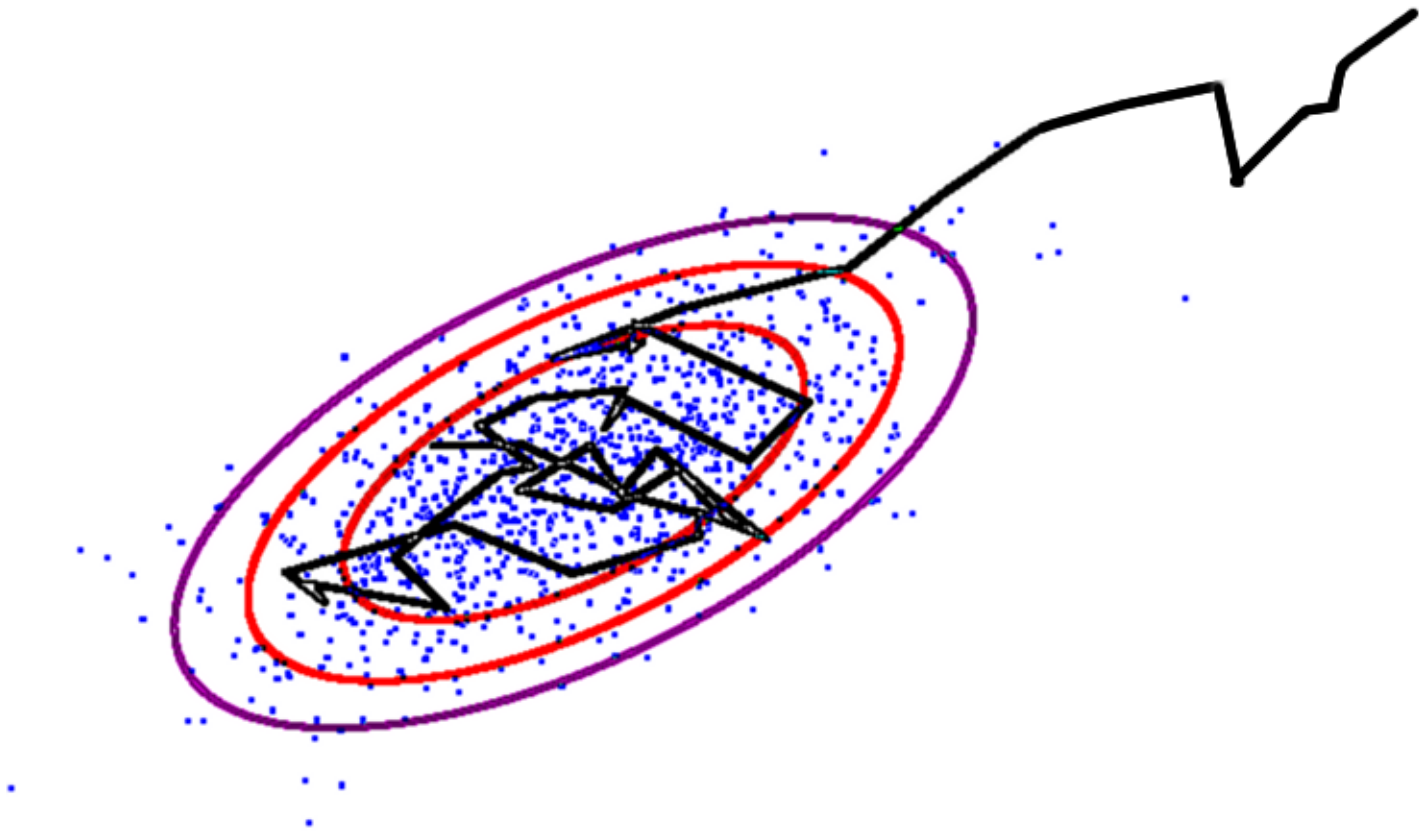
This subfigure from PRML, Bishop (2006)

Detail: Metropolis, as stated, requires $Q(\theta'; \theta) = Q(\theta; \theta')$

Markov chain Monte Carlo

Construct a biased random walk that explores target dist $P^\star(x)$

Markov steps, $x_t \sim T(x_t \leftarrow x_{t-1})$



MCMC gives approximate, correlated samples from $P^\star(x)$

Transition operators

Discrete example

$$P^\star = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} \qquad T = \begin{pmatrix} 2/3 & 1/2 & 1/2 \\ 1/6 & 0 & 1/2 \\ 1/6 & 1/2 & 0 \end{pmatrix} \qquad T_{ij} = T(x_i \leftarrow x_j)$$

$P^\star$ is an **invariant distribution** of T because $TP^\star = P^\star$, i.e.

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

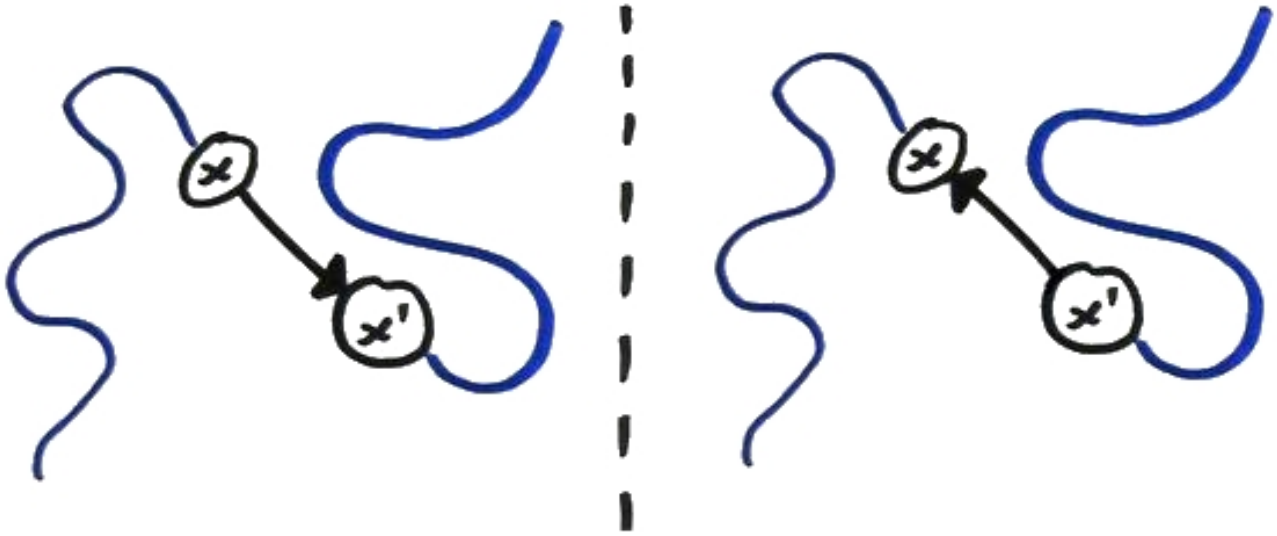
Also $P^\star$ is *the* **equilibrium distribution** of T:

To machine precision: $T^{100} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3/5 \\ 1/5 \\ 1/5 \end{pmatrix} = P^\star$

Ergodicity requires: $T^K(x' \leftarrow x) > 0$ for all $x' : P^\star(x') > 0$, for some K

Detailed Balance

Detailed balance means $\rightarrow x \rightarrow x'$ and $\rightarrow x' \rightarrow x$ are equally probable:



$$T(x' \leftarrow x)P^\star(x) = T(x \leftarrow x')P^\star(x')$$

Detailed balance implies the invariant condition:

$$\sum_x T(x' \leftarrow x)P^\star(x) \;=\; P^\star(x') \sum_x T(x \leftarrow x') \;\; 1$$

Enforcing detailed balance is easy: it only involves isolated pairs

Reverse operators

If T satisfies stationarity, we can define a *reverse operator*

$$\widetilde{T}(x \leftarrow x') \propto T(x' \leftarrow x)\, P^\star(x) = \frac{T(x' \leftarrow x)\, P^\star(x)}{\sum_x T(x' \leftarrow x)\, P^\star(x)} = \frac{T(x' \leftarrow x)\, P^\star(x)}{P^\star(x')}$$

Generalized balance condition:

$$T(x' \leftarrow x) P^\star(x) \;=\; \widetilde{T}(x \leftarrow x') P^\star(x')$$

also implies the invariant condition *and is necessary*.

Operators satisfying detailed balance are their own reverse operator.

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 $\tilde{P} \propto P(x)$; normalizer cancels in acceptance ratio
- Satisfies detailed balance (shown below)
- Q must be chosen to fulfill the other technical requirements

---

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

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

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
```

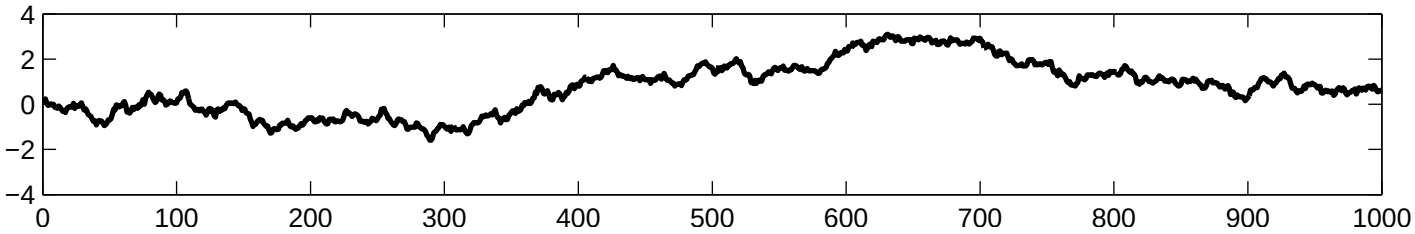
Step-size demo

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

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

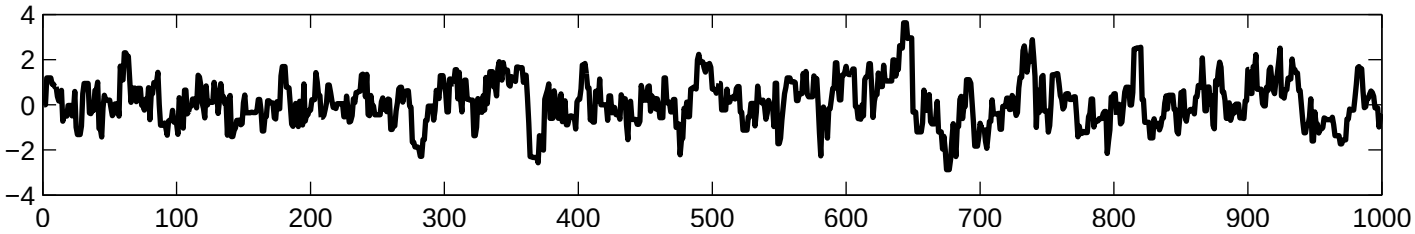
`sigma(0.1)`

99.8% accepts



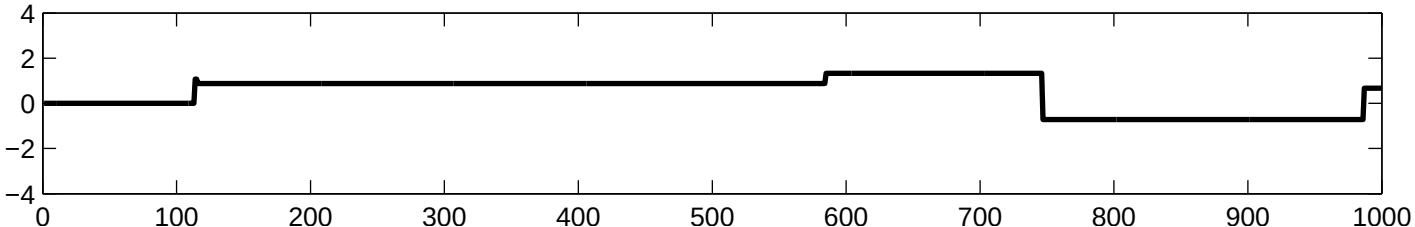
`sigma(1)`

68.4% accepts

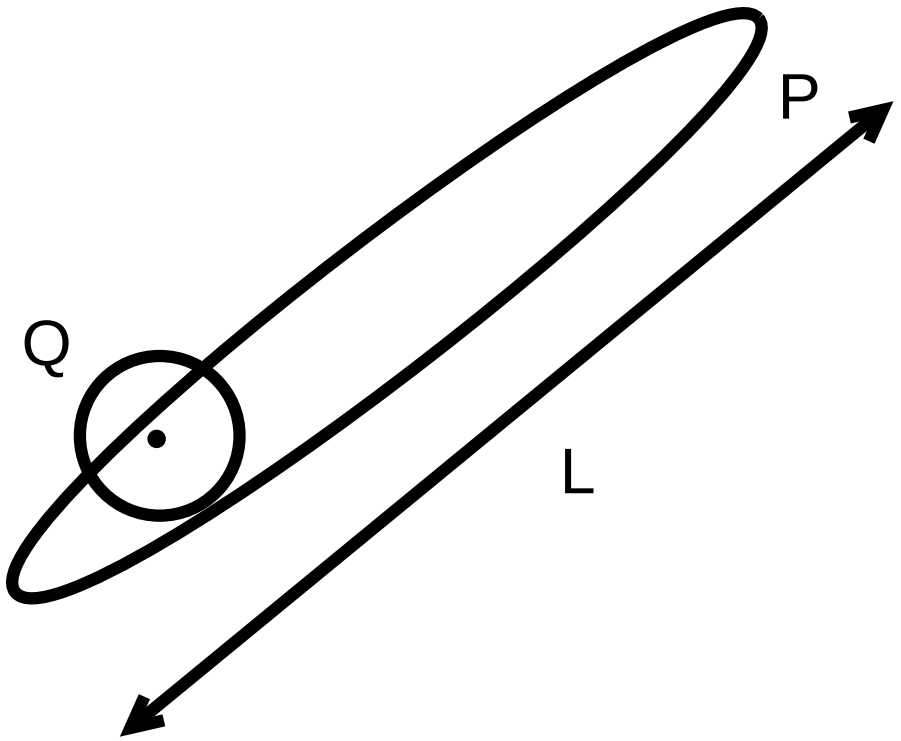


`sigma(100)`

0.5% accepts



Metropolis limitations



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

Combining operators

A sequence of operators, each with $P^\star$ invariant:

$$
\begin{array}{lll}
x_0 \sim P^\star(x) & & \\
x_1 \sim T_a(x_1 \leftarrow x_0) & P(x_1) = \sum_{x_0} T_a(x_1 \leftarrow x_0) P^\star(x_0) & = P^\star(x_1) \\
x_2 \sim T_b(x_2 \leftarrow x_1) & P(x_2) = \sum_{x_1} T_b(x_2 \leftarrow x_1) P^\star(x_1) & = P^\star(x_2) \\
x_3 \sim T_c(x_3 \leftarrow x_2) & P(x_3) = \sum_{x_1} T_c(x_3 \leftarrow x_2) P^\star(x_2) & = P^\star(x_3) \\
\ldots & \ldots &
\end{array}
$$

— Combination $T_c T_b T_a$ leaves $P^\star$ invariant

— If they can reach any x, $T_c T_b T_a$ is a valid MCMC operator

— Individually T_c, T_b and T_a need not be ergodic

Gibbs sampling

A method with no rejections:

- Initialize $\mathbf{x}$ to some value
- Pick each variable in turn or randomly and resample $P(x_i|\mathbf{x}_{j\neq i})$

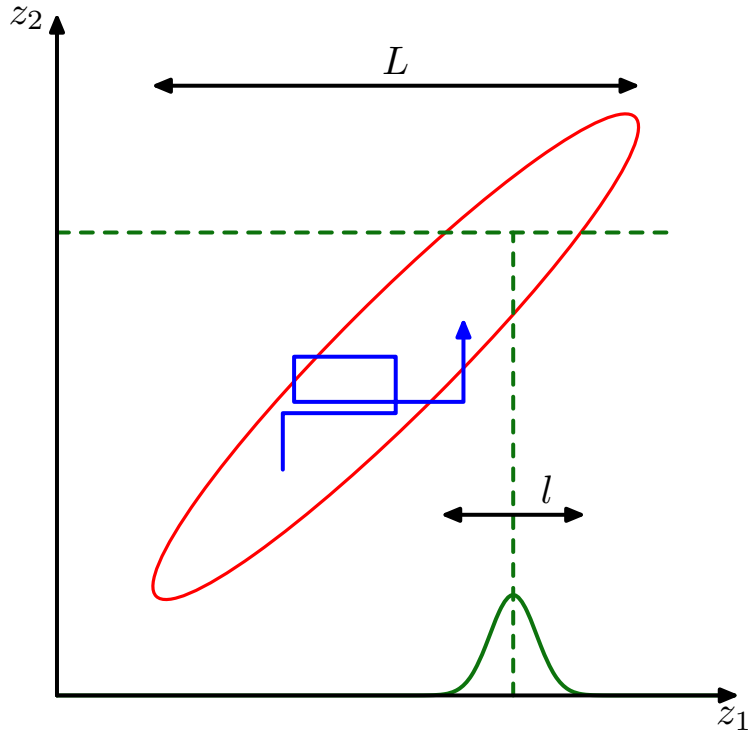


Figure from PRML, Bishop (2006)

Proof of validity: **a)** check detailed balance for component update.
b) Metropolis–Hastings 'proposals' $P(x_i|\mathbf{x}_{j\neq i}) \Rightarrow$ accept with prob. 1
Apply a series of these operators. Don't need to check acceptance.

Gibbs sampling

Alternative explanation:

Chain is currently at $\mathbf{x}$

At equilibrium can assume $\mathbf{x} \sim P(\mathbf{x})$

Consistent with $\mathbf{x}_{j \neq i} \sim P(\mathbf{x}_{j \neq i}), \quad x_i \sim P(x_i \,|\, \mathbf{x}_{j \neq i})$

Pretend x_i was never sampled and do it again.

This view may be useful later for non-parametric applications

"Routine" Gibbs sampling

Gibbs sampling benefits from few free choices and convenient features of conditional distributions:

- Conditionals with a few discrete settings can be explicitly normalized:

$$P(x_i|\mathbf{x}_{j\neq i}) \propto P(x_i, \mathbf{x}_{j\neq i})$$
$$= \frac{P(x_i, \mathbf{x}_{j\neq i})}{\sum_{x_i'} P(x_i', \mathbf{x}_{j\neq i})} \leftarrow \text{this sum is small and easy}$$

- Continuous conditionals only univariate
  $\Rightarrow$ amenable to standard sampling methods.

`WinBUGS` and `OpenBUGS` sample graphical models using these tricks

Summary so far

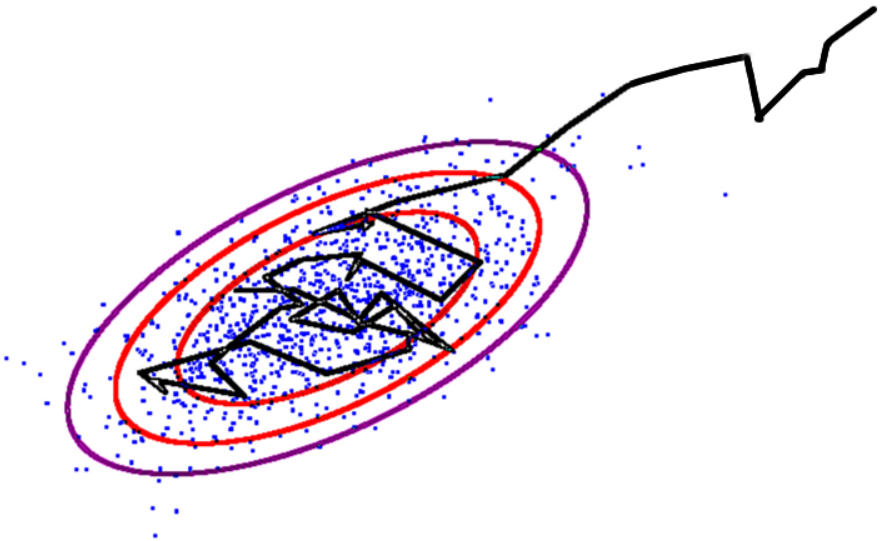
- We need approximate methods to solve sums/integrals
- Monte Carlo does not explicitly depend on dimension, although simple methods work only in low dimensions
- Markov chain Monte Carlo (MCMC) can make local moves. By assuming less, it's more applicable to higher dimensions
- simple computations $\Rightarrow$ "easy" to implement (harder to diagnose).

How do we use these MCMC samples?

End of Lecture 1

Quick review

Construct a biased random walk that explores a target dist.



Markov steps, $x^{(s)} \sim T\left(x^{(s)} \leftarrow x^{(s-1)}\right)$

MCMC gives approximate, correlated samples

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

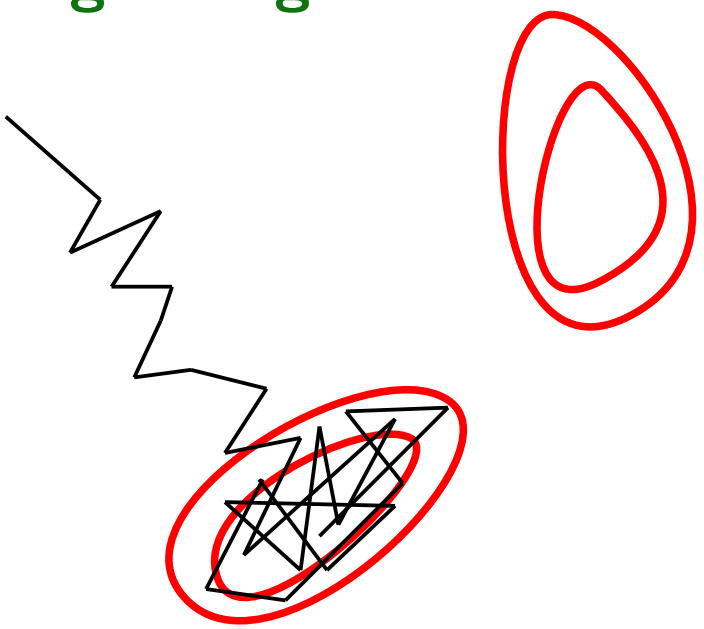
Example transitions:

Metropolis–Hastings: $T(x' \leftarrow x) = Q(x'; x)\, \min\left(1,\, \frac{P(x')\, Q(x; x')}{P(x)\, Q(x'; x)}\right)$

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

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

Approximately independent samples can be obtained by *thinning*.
However, **all the samples can be used.**

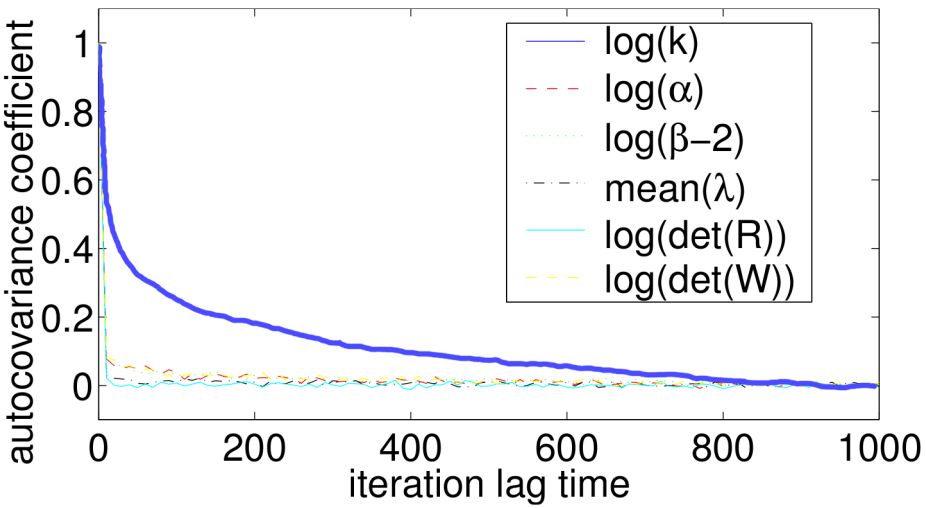
Use the simple Monte Carlo estimator on MCMC samples. It is:

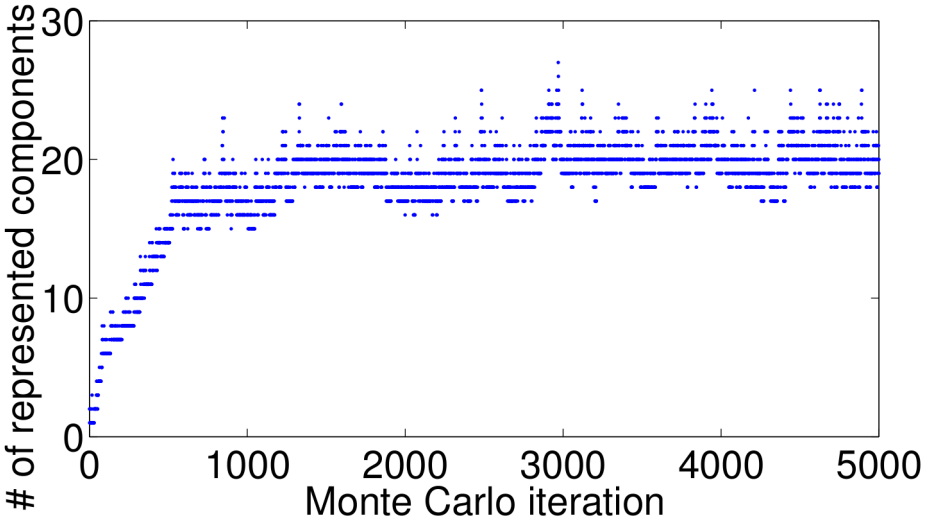
— consistent

— unbiased if the chain has "burned in"

The correct motivation to thin: if computing $f(\mathbf{x}^{(s)})$ is expensive

Empirical diagnostics





Rasmussen (2000)

Recommendations

For diagnostics:
Standard software packages like `R-CODA`

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

Practical Markov chain Monte Carlo
Charles J. Geyer, *Statistical Science*. 7(4):473–483, 1992.
`http://www.jstor.org/stable/2246094`

Consistency checks

Do I get the right answer on tiny versions of my problem?

Can I make good inferences about synthetic data drawn from my model?

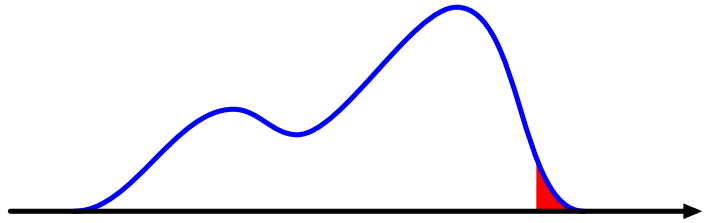
Getting it right: joint distribution tests of posterior simulators, John Geweke, $JASA$, 99(467):799–804, 2004.

[next: using the samples]

Making good use of samples

Is the standard estimator too noisy?

e.g. need many samples from a distribution to estimate its tail



We can often do *some* analytic calculations

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 $P(x_i=1|\mathbf{x}_{\setminus i})$

$$P(x_i=1) = \sum_{\mathbf{x}_{\setminus i}} P(x_i=1|\mathbf{x}_{\setminus i})P(\mathbf{x}_{\setminus i})$$

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

Example of "Rao-Blackwellization". See also "waste recycling".

Processing samples

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

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

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

A more general form of "Rao-Blackwellization".

Summary so far

- MCMC algorithms are general and often easy to implement
- Running them *is* a bit messy. . .
  . . . but there are some established procedures.
- Given the samples there might be a choice of estimators

Next question:
Is MCMC research all about finding a good $Q(\mathbf{x})$?

Auxiliary variables

The point of MCMC is to marginalize out variables, but one can introduce more variables:

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

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

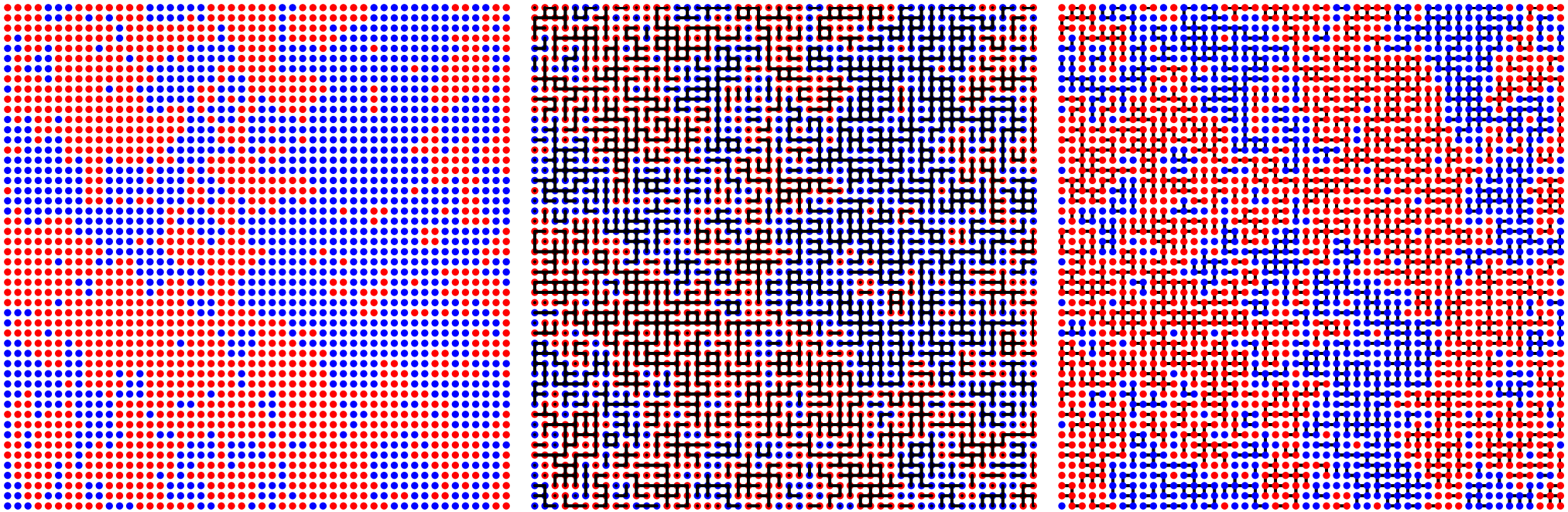
We might want to do this if

- $P(x|v)$ and $P(v|x)$ are simple
- $P(x,v)$ is otherwise easier to navigate

Swendsen–Wang (1987)

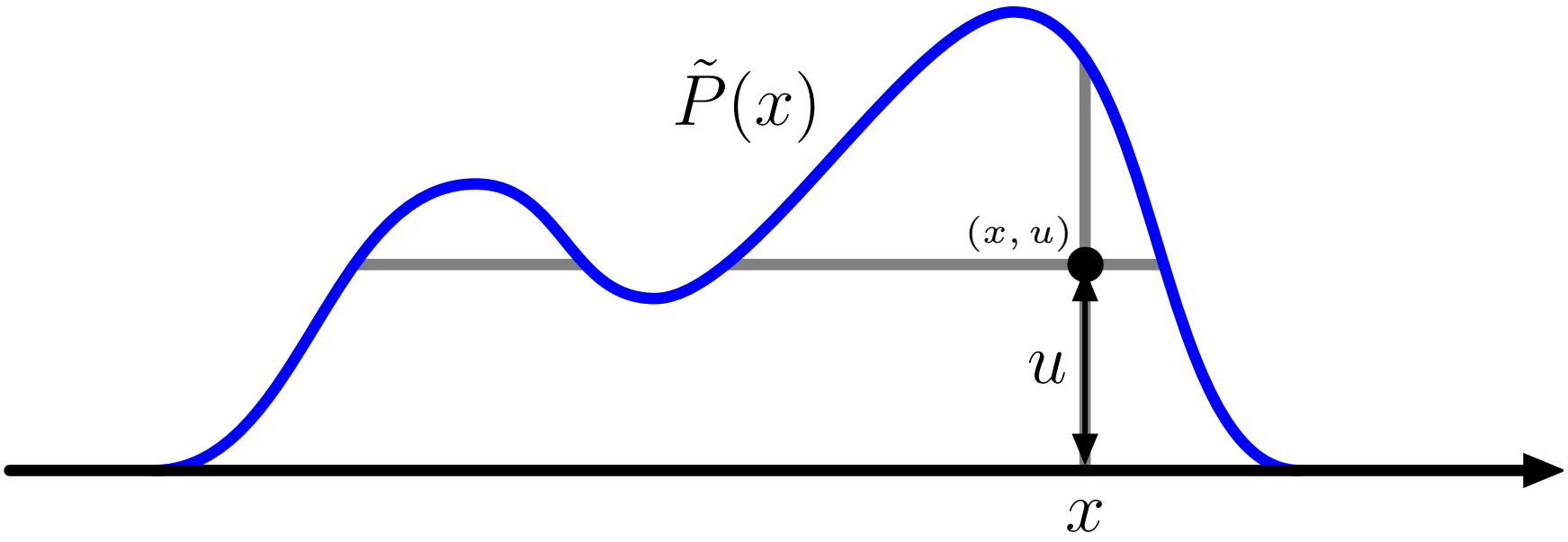
Seminal algorithm using auxiliary variables

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



Slice sampling idea

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

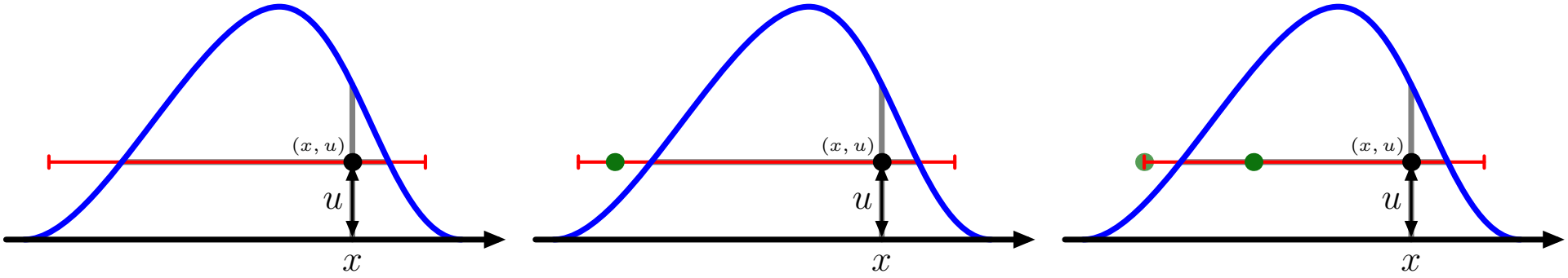


$$p(u|x) = \text{Uniform}[0, \tilde{P}(x)]$$

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

Slice sampling

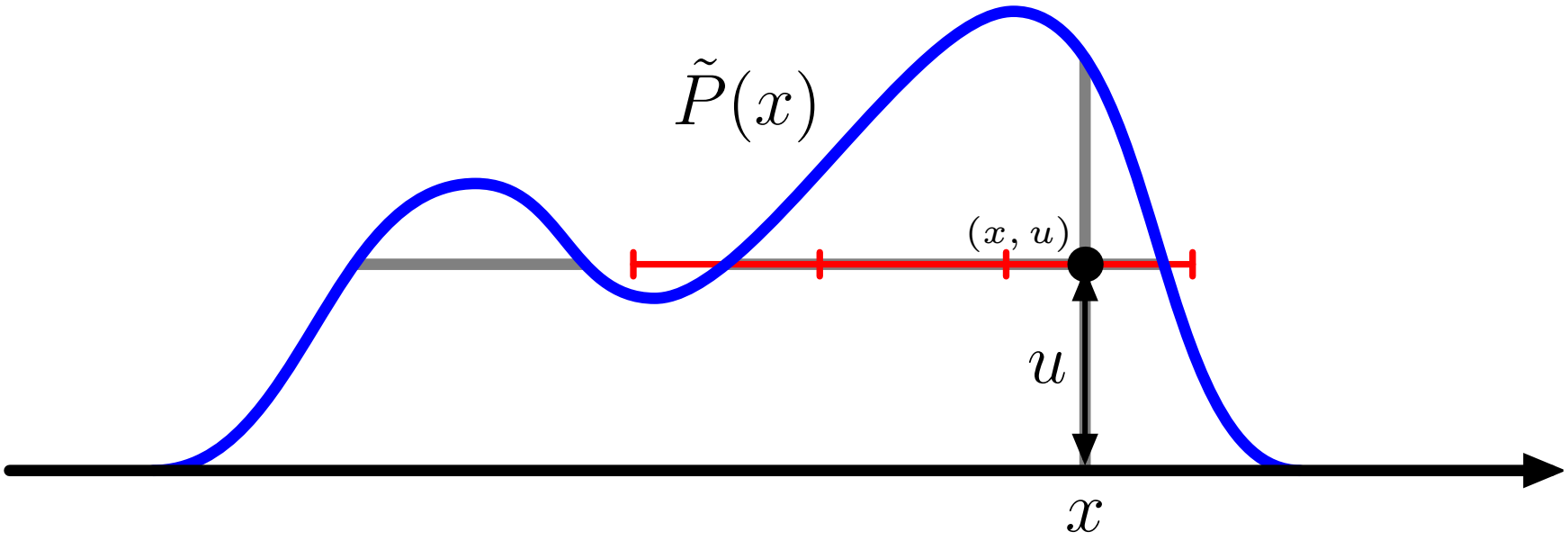
Unimodal conditionals



- 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 $\tilde{P}(x) \propto P(x)$ pointwise
- No rejections
- Step-size parameters less important than Metropolis

More advanced versions of slice sampling have been developed. Neal (2003) contains *many* ideas.

Hamiltonian dynamics

Construct a landscape with gravitational potential energy, E(x):

$$P(x) \propto e^{-E(x)}, \qquad E(x) = -\log P^*(x)$$

Introduce velocity v carrying kinetic energy $K(v) = v^\top v/2$

Some physics:

- Total energy or Hamiltonian, $H = E(x) + K(v)$
- Frictionless ball rolling $(x, v) \rightarrow (x', v')$ satisfies $H(x', v') = H(x, v)$
- Ideal Hamiltonian dynamics are time reversible:
  - reverse v and the ball will return to its start point

Hamiltonian Monte Carlo

Define a joint distribution:

- $P(x, v) \propto e^{-E(x)} e^{-K(v)} = e^{-E(x)-K(v)} = e^{-H(x,v)}$
- Velocity is independent of position and Gaussian distributed

Markov chain operators

- Gibbs sample velocity
- Simulate Hamiltonian dynamics then flip sign of velocity
  - Hamiltonian 'proposal' is deterministic and reversible $q(x', v'; x, v) = q(x, v; x', v') = 1$
  - Conservation of energy means $P(x, v) = P(x', v')$
  - Metropolis acceptance probability is 1

Except we can't simulate Hamiltonian dynamics exactly

Leap-frog dynamics

a discrete approximation to Hamiltonian dynamics:

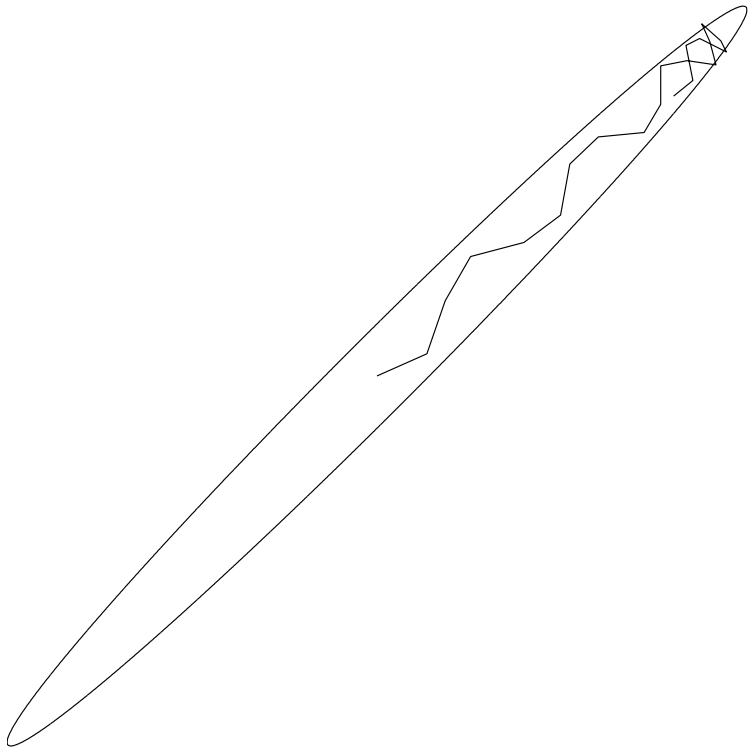
$$
\begin{aligned}
v_i(t + \tfrac{\epsilon}{2}) &= v_i(t) - \frac{\epsilon}{2} \frac{\partial E(x(t))}{\partial x_i} \\
x_i(t + \epsilon) &= x_i(t) + \epsilon\, v_i(t + \tfrac{\epsilon}{2}) \\
p_i(t + \epsilon) &= v_i(t + \tfrac{\epsilon}{2}) - \frac{\epsilon}{2} \frac{\partial E(x(t + \epsilon))}{\partial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

- H is not conserved
- dynamics are still deterministic and reversible
- 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 Leapfrog dynamics for L steps
- Accept new position 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 on which it was based.

Summary of auxiliary variables

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

A fair amount of my research (not covered in this tutorial) has been finding the right auxiliary representation on which to run standard MCMC updates.

Example benefits:

Population methods to give better mixing and exploit parallel hardware

Being robust to bad random number generators

Removing step-size parameters when slice sample doesn't really apply

Finding normalizers is hard

Prior sampling: **like finding fraction of needles in a hay-stack**

$$P(\mathcal{D}|\mathcal{M}) = \int P(\mathcal{D}|\theta, \mathcal{M}) P(\theta|\mathcal{M}) \, \mathrm{d}\theta$$

$$= \frac{1}{S} \sum_{s=1}^{S} P(\mathcal{D}|\theta^{(s)}, \mathcal{M}), \quad \theta^{(s)} \sim P(\theta|\mathcal{M})$$

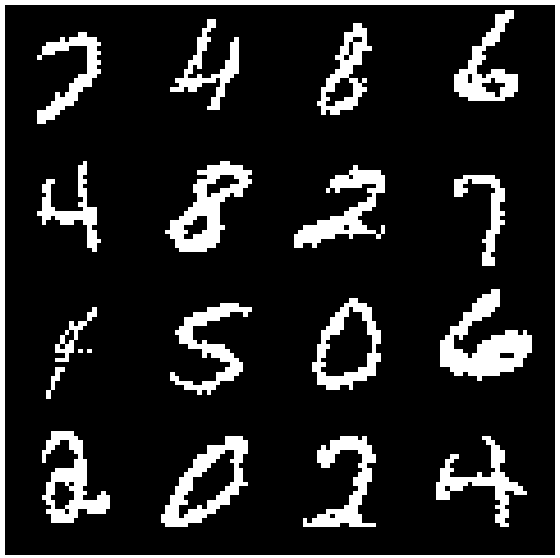
. . . usually has huge variance

Similarly for undirected graphs:

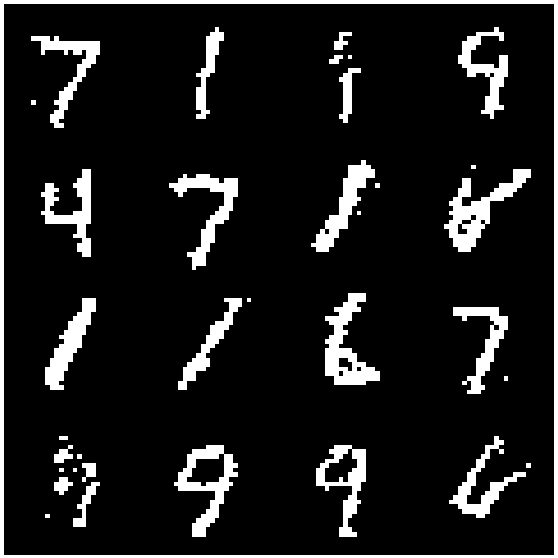
$$P(\mathbf{x}) = \frac{P^*(\mathbf{x})}{\mathcal{Z}}, \qquad \mathcal{Z} = \sum_{\mathbf{x}} P^*(\mathbf{x})$$

I will use this as an easy-to-illustrate case-study

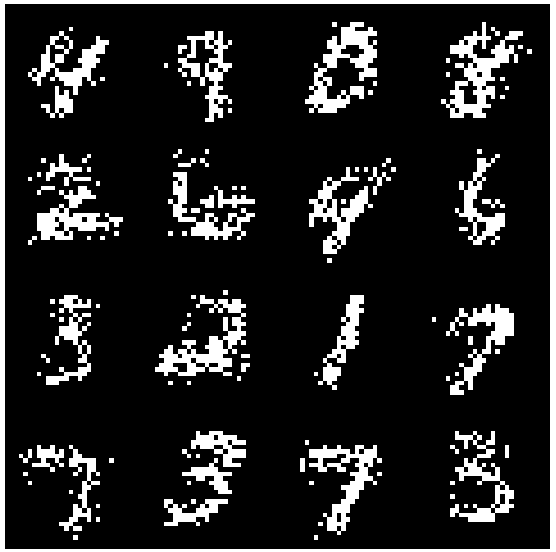
Benchmark experiment



Training set



RBM samples



MoB samples

RBM setup:

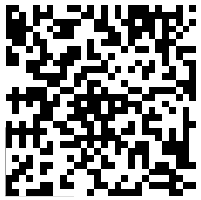
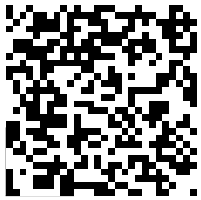
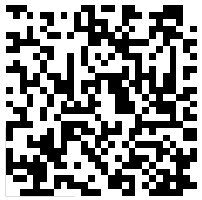
— $28 \times 28 = 784$ binary visible variables

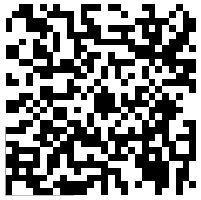
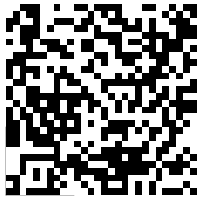
— 500 binary hidden variables

Goal: Compare $P(\mathbf{x})$ on test set, $(P_{\text{RBM}}(\mathbf{x}) = P^*(\mathbf{x})/\mathcal{Z})$

Simple Importance Sampling

$$\mathcal{Z} = \sum_{\mathbf{x}} \frac{P^*(\mathbf{x})}{Q(\mathbf{x})} Q(\mathbf{x}) \approx \frac{1}{S} \sum_{s=1}^{S} \frac{P^*(\mathbf{x}^{(s)})}{Q(\mathbf{x})}, \quad \mathbf{x}^{(s)} \sim Q(\mathbf{x})$$

$\mathbf{x}^{(1)} =$  , $\mathbf{x}^{(2)} =$  , $\mathbf{x}^{(3)} =$  ,

$\mathbf{x}^{(4)} =$  , $\mathbf{x}^{(5)} =$  , $\mathbf{x}^{(6)} =$  , . . .

$$\mathcal{Z} = 2^D \sum_{\mathbf{x}} \frac{1}{2^D} P^*(\mathbf{x}) \approx \frac{2^D}{S} \sum_{s=1}^{S} P^*(\mathbf{x}^{(s)}), \quad \mathbf{x}^{(s)} \sim \text{Uniform}$$

"Posterior" Sampling

Sample from $P(\mathbf{x}) = \frac{P^*(\mathbf{x})}{\mathcal{Z}}$, $\left[\text{or } P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})}\right]$

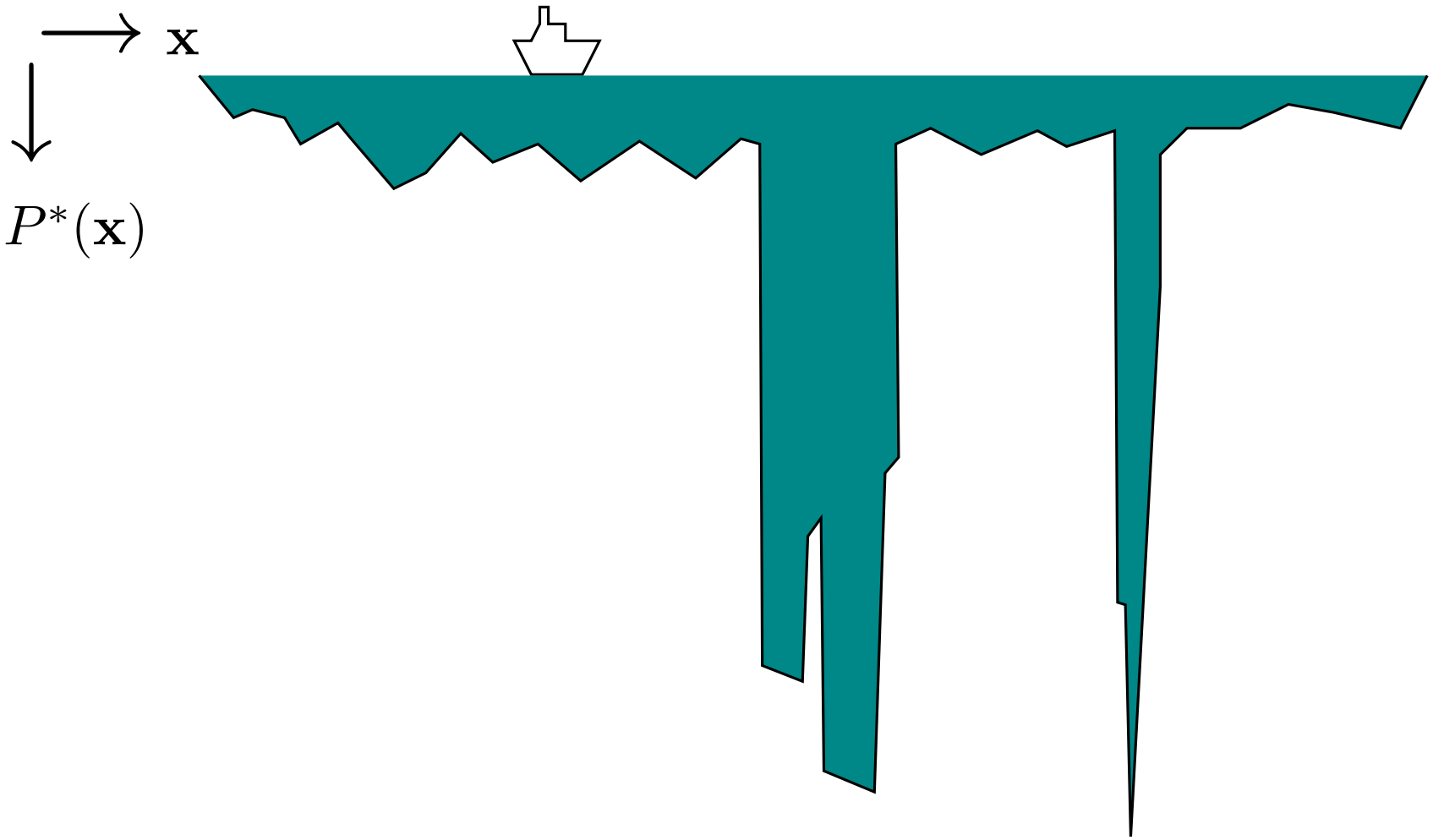
$\mathbf{x}^{(1)} =$  , $\mathbf{x}^{(2)} =$  , $\mathbf{x}^{(3)} =$  ,

$\mathbf{x}^{(4)} =$  , $\mathbf{x}^{(5)} =$  , $\mathbf{x}^{(6)} =$  , . . .

$$\mathcal{Z} = \sum_{\mathbf{x}} P^*(\mathbf{x})$$

$$\mathcal{Z} \text{ "}\approx\text{" } \frac{1}{S}\sum_{s=1}^{S}\frac{P^*(\mathbf{x})}{P(\mathbf{x})} = \mathcal{Z}$$

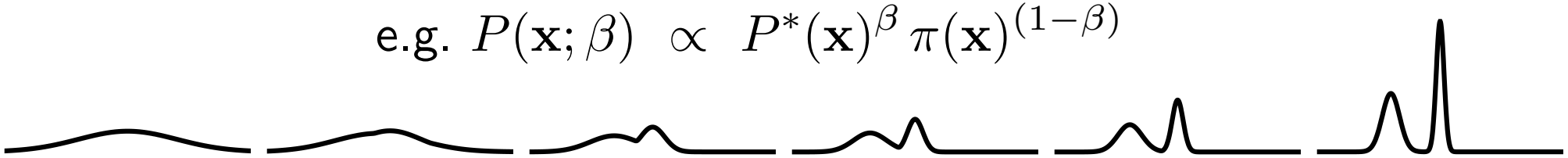
Finding a Volume



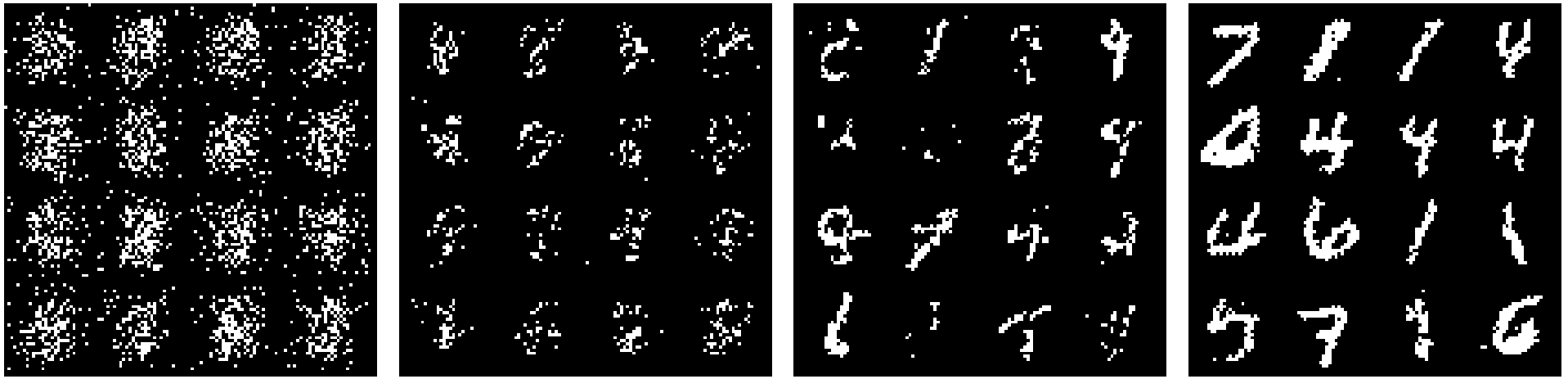
Lake analogy and figure from MacKay textbook (2003)

Annealing / Tempering

e.g. $P(\mathbf{x};\beta) \;\propto\; P^*(\mathbf{x})^{\beta}\,\pi(\mathbf{x})^{(1-\beta)}$



$\beta = 0$ $\beta = 0.01$ $\beta = 0.1$ $\beta = 0.25$ $\beta = 0.5$ $\beta = 1$

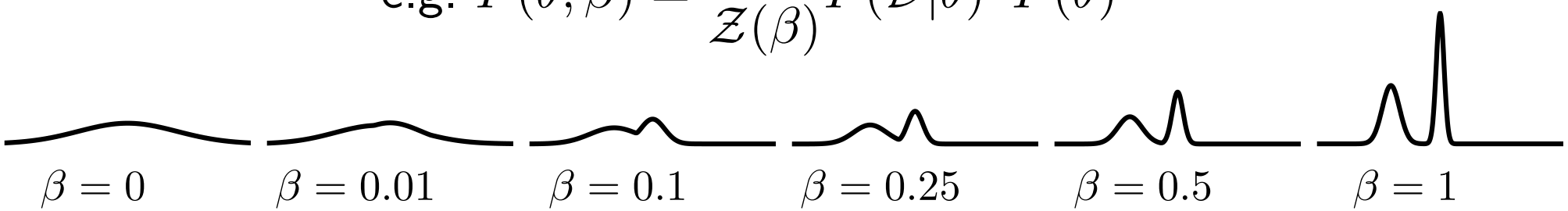


$1/\beta \;=$ "temperature"

Using other distributions

$Chain$ between posterior and prior:

$$\text{e.g. } P(\theta;\beta) = \frac{1}{\mathcal{Z}(\beta)} P(\mathcal{D}|\theta)^{\beta} P(\theta)$$



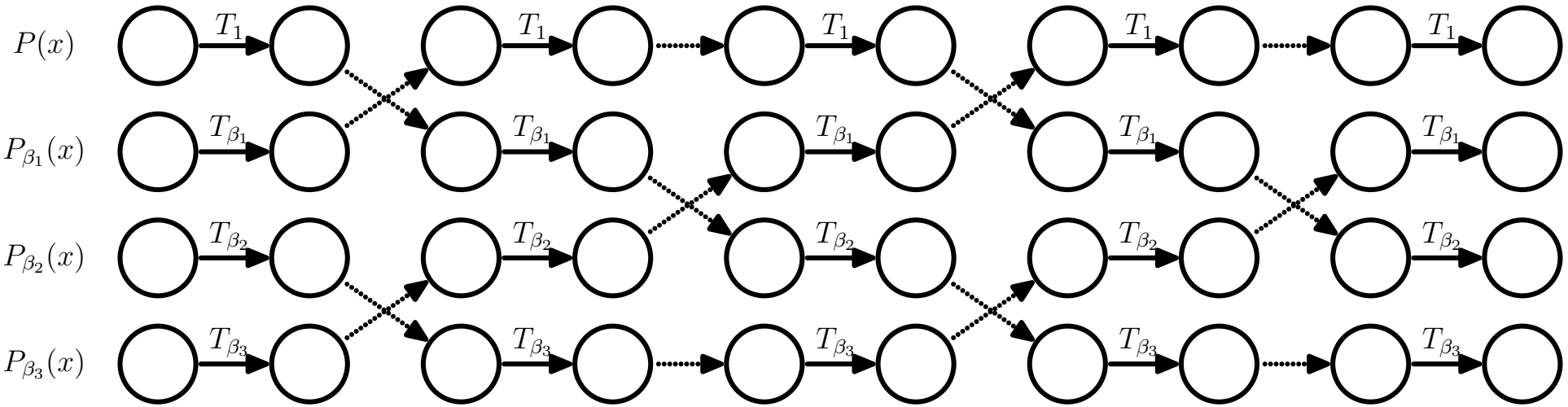
Advantages:

- mixing easier at low β, good initialization for higher β?
- $$\frac{\mathcal{Z}(1)}{\mathcal{Z}(0)} = \frac{\mathcal{Z}(\beta_1)}{\mathcal{Z}(0)} \cdot \frac{\mathcal{Z}(\beta_2)}{\mathcal{Z}(\beta_1)} \cdot \frac{\mathcal{Z}(\beta_3)}{\mathcal{Z}(\beta_2)} \cdot \frac{\mathcal{Z}(\beta_4)}{\mathcal{Z}(\beta_3)} \cdot \frac{\mathcal{Z}(1)}{\mathcal{Z}(\beta_4)}$$

Related to *annealing* or *tempering*, $1/\beta =$ "temperature"

Parallel tempering

Normal MCMC transitions + swap proposals on $P(X) = \prod_{\beta} P(X;\beta)$

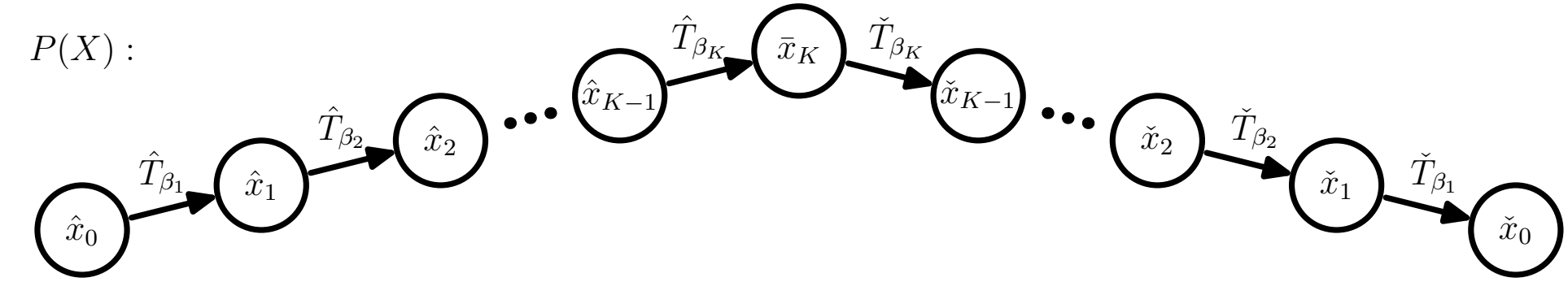


Problems / trade-offs:

- obvious space cost
- need to equilibriate larger system
- information from low β diffuses up by slow random walk

Tempered transitions

Drive temperature up. . .



$\hat{x}_0 \sim P(x)$

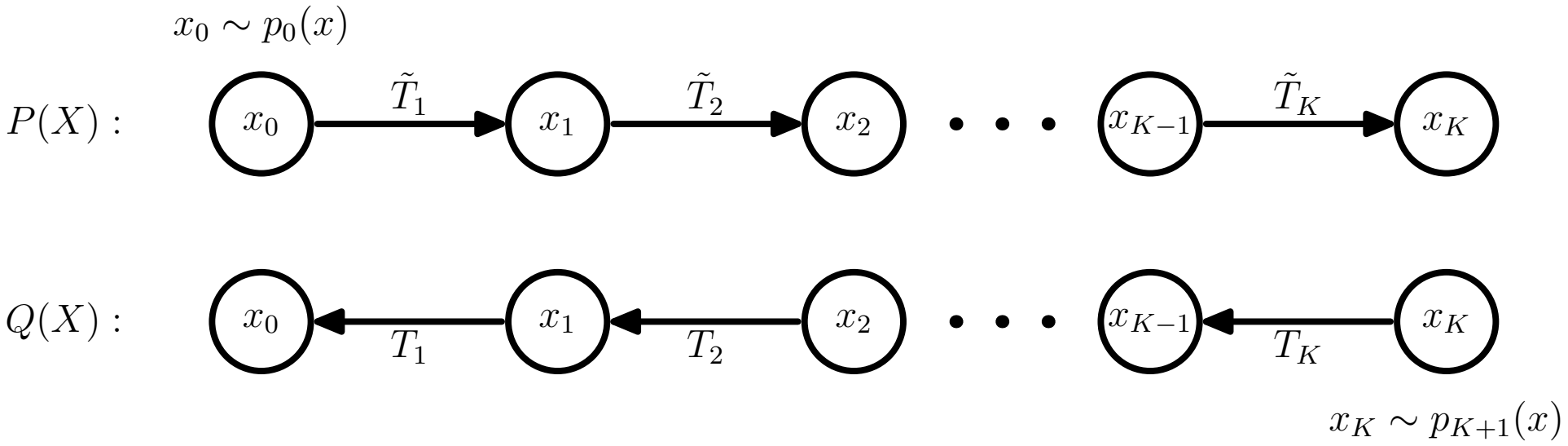
. . . and back down

Proposal: swap order of points so final point $\check{x}_0$ putatively $\sim P(x)$

Acceptance probability:

$$\min\left[1,\ \frac{P_{\beta_1}(\hat{x}_0)}{P(\hat{x}_0)}\cdots\frac{P_{\beta_K}(\hat{x}_{K-1})}{P_{\beta_{K-1}}(\hat{x}_0)}\frac{P_{\beta_{K-1}}(\check{x}_{K-1})}{P_{\beta_K}(\check{x}_{K-1})}\cdots\frac{P(\check{x}_0)}{P_{\beta_1}(\check{x}_0)}\right]$$

Annealed Importance Sampling

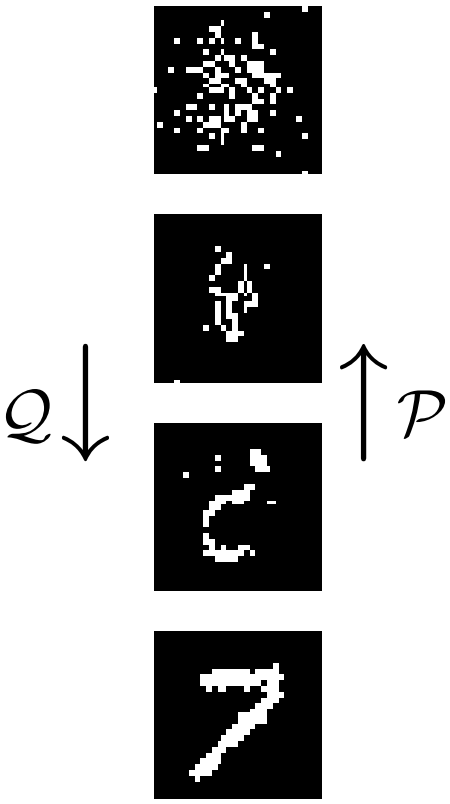


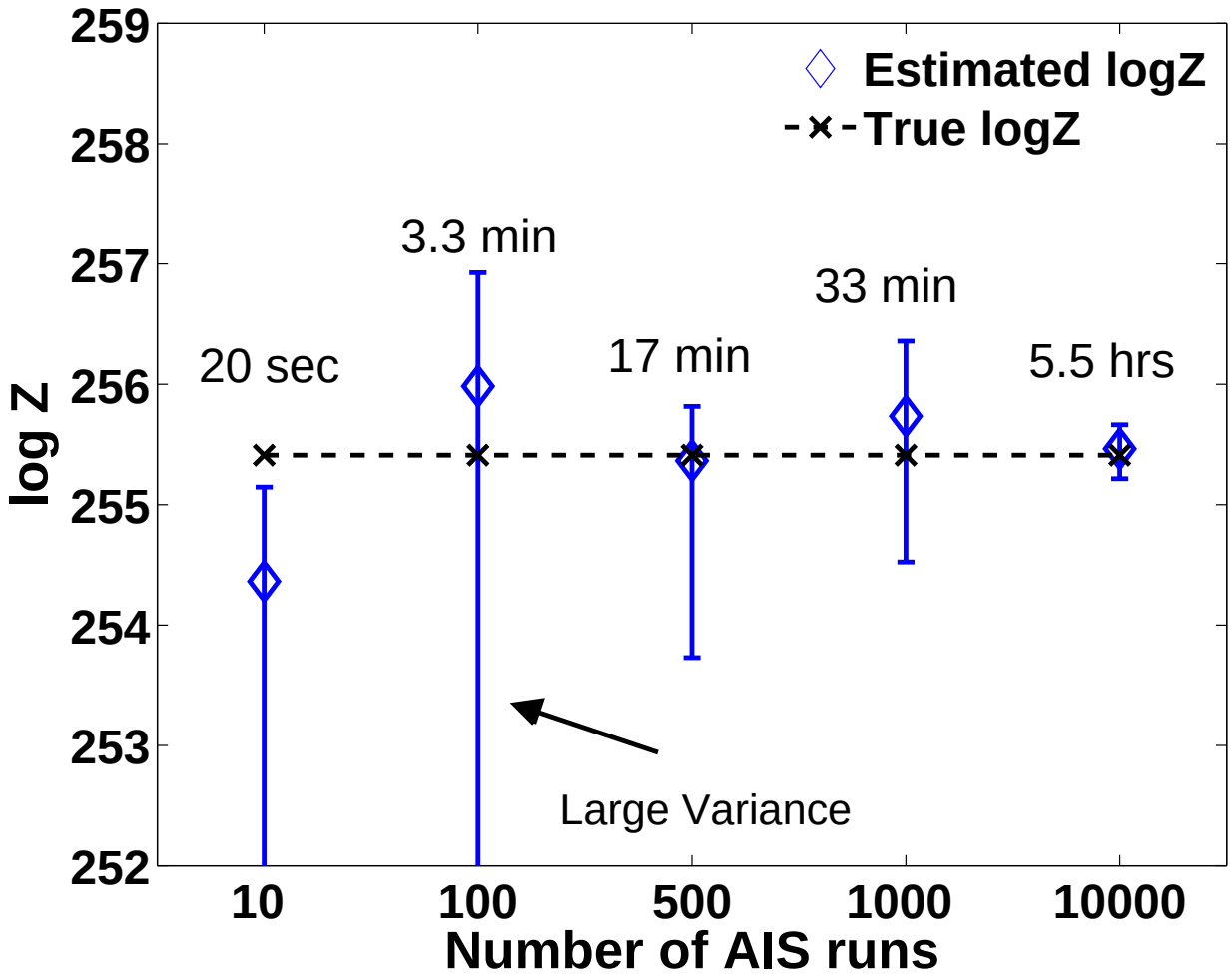
$$\mathcal{P}(X) = \frac{P^*(\mathbf{x}_K)}{\mathcal{Z}} \prod_{k=1}^{K} \widetilde{T}_k(\mathbf{x}_{k-1}; \mathbf{x}_k), \qquad \mathcal{Q}(X) = \pi(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k; \mathbf{x}_{k-1})$$

Then standard importance sampling of $\mathcal{P}(X) = \frac{\mathcal{P}^*(X)}{\mathcal{Z}}$ with $\mathcal{Q}(X)$

Annealed Importance Sampling

$$\mathcal{Z} \approx \frac{1}{S} \sum_{s=1}^{S} \frac{\mathcal{P}^*(X)}{\mathcal{Q}(X)}$$





Summary on $\mathcal{Z}$

Whirlwind tour of roughly how to find $\mathcal{Z}$ with Monte Carlo

The algorithms really have to be good at exploring the distribution

These are also the Monte Carlo approaches to watch for general use on the hardest problems.

Can be useful for optimization too.

See the references for more.

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/`
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 modelling, 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`