Markov chain Monte Carlo

Probabilistic Models of Cognition, 2011

http://www.ipam.ucla.edu/programs/gss2011/

Roadmap:

— Some practicalities

— What can we prove?

— Building better chains:

  — Auxiliary variables

— Normalizing constants

— References

Iain Murray

http://homepages.inf.ed.ac.uk/imurray2/

tinyurl.com/murray-ipam

Iain Murray

Lecturer in Machine Learning, ANC,
School of Informatics, University of Edinburgh.

Email: i.murray@ed.ac.uk

Currently highlighting:

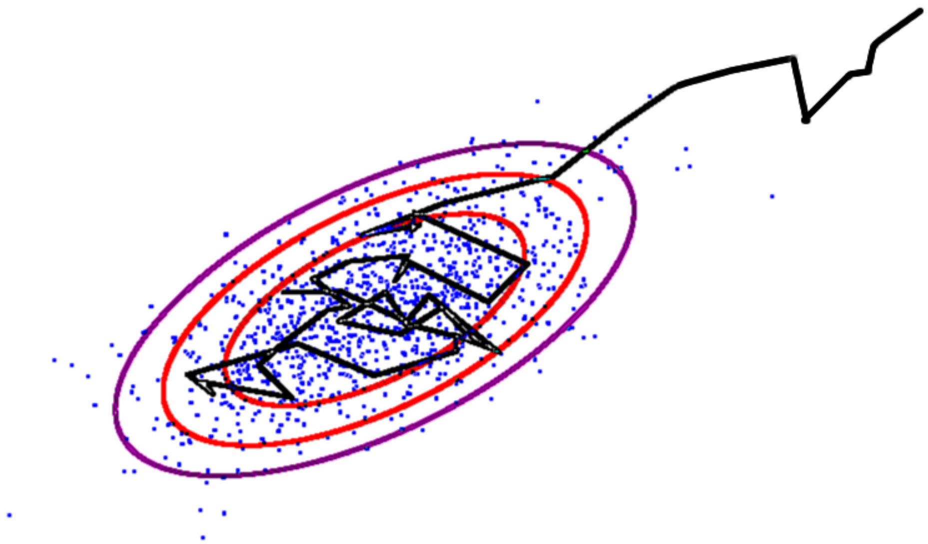
- *Density estimation*: Neural Autoregressive Distribution Estimator.
- *Latent Gaussians*: simulating variables and hyperparameters (video).
- *Teaching*: IPAM slides, Octave/Matlab; Intro. to Machine Learning, MCMC.

Main content

- Publications
- Teaching
- Code

Quick review

Construct a biased random walk that explores a target dist.



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

MCMC gives approximate, correlated samples

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

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

“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:

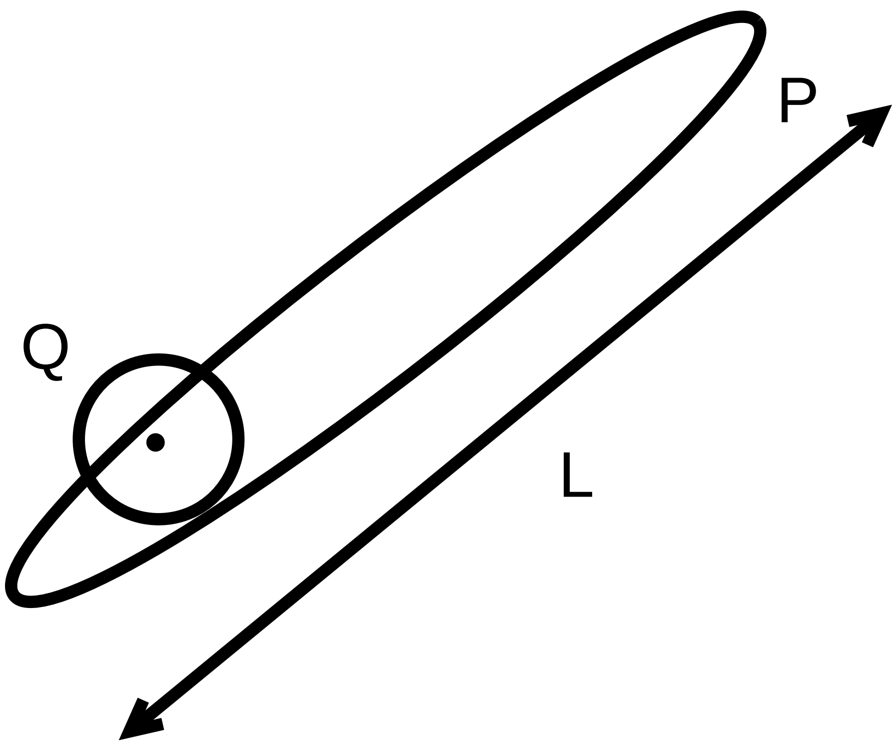
$$P(x_i|\mathbf{x}_{j\neq i}) \;\propto\; P(x_i, \mathbf{x}_{j\neq i})$$

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

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

`WinBUGS`, `OpenBUGS`, `JAGS` and others use these tricks

Diffusion time



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

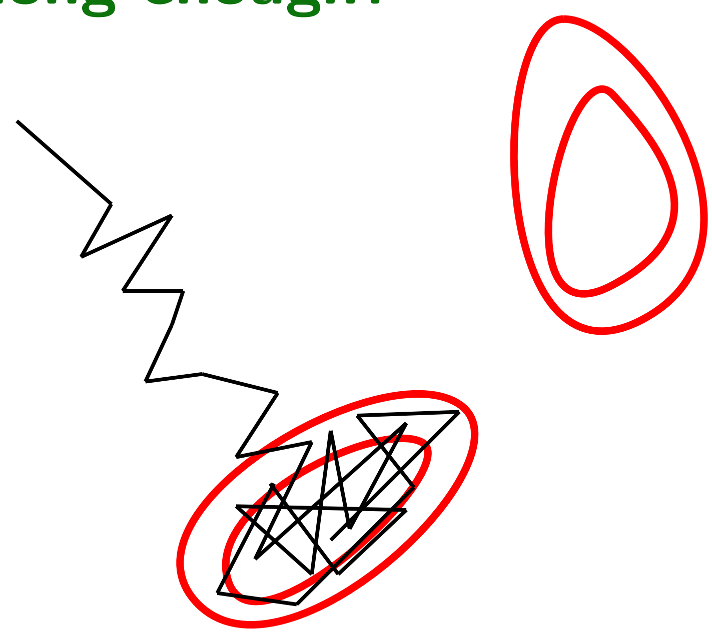
σ large $\rightarrow$ many rejections

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

Adapted from MacKay (2003)

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

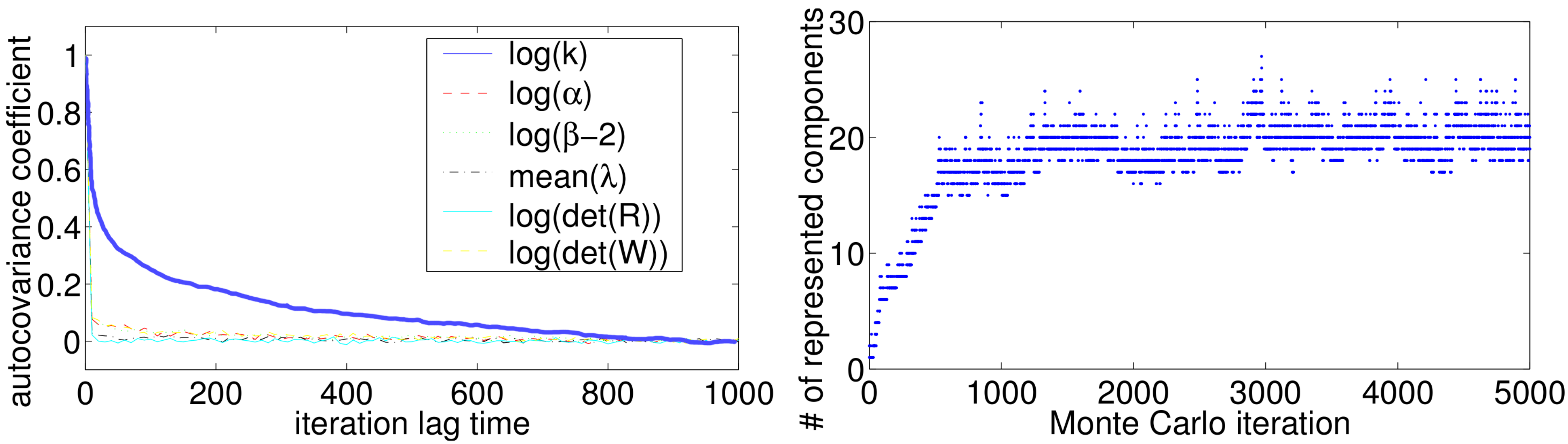
— unbiased if the chain has "burned in"

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

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

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

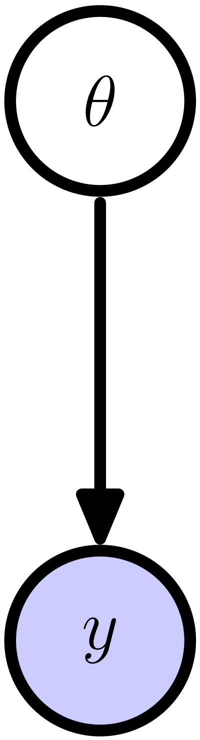
Do I get the right answer on tiny versions of my problem?

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.

Posterior Model checking: Gelman et al. Bayesian Data Analysis textbook and papers.

Getting it right



We write MCMC code to update $\theta \,|\, y$

Idea: also write code to sample $y \,|\, \theta$

Both codes leave $P(\theta, y)$ invariant

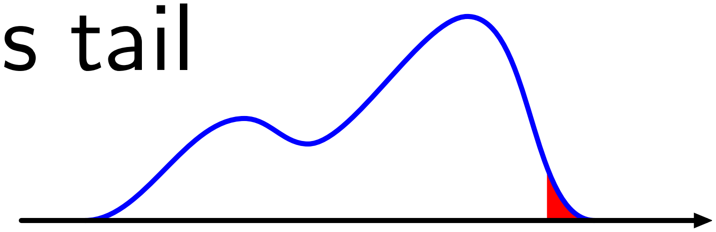
Run codes alternately. Check θ's match prior

Doing some analytic math

Collapsed sampler: marginalize some variables

Is the standard estimator too noisy?

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



Maybe we can use samples better

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

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

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

A more general form of “Rao-Blackwellization”.

Summary so far

- MCMC is general and often easy to implement
- Running it *is* a bit messy. . .
  . . . but there are some established procedures.
- There can be a choice of estimators

Can we prove anything?

It's usually hard to have many guarantees.

Sometimes convergence theory can be practical:

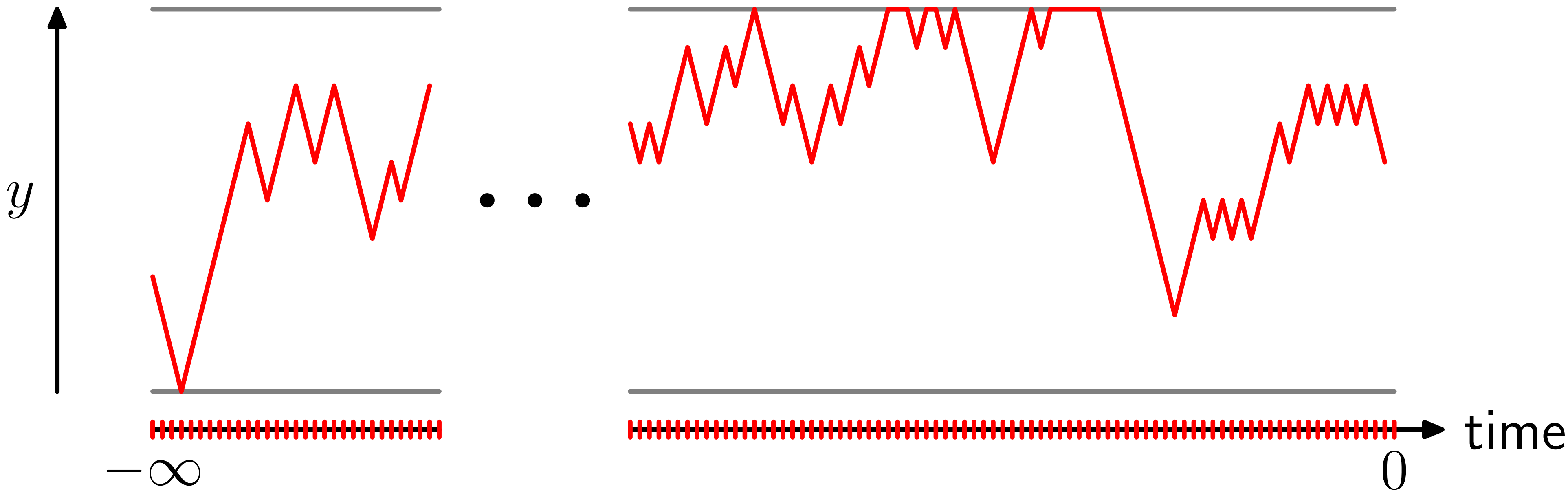
Markov chain Monte Carlo algorithms: theory and practice
Jeffrey S. Rosenthal
http://probability.ca/jeff/ftpdir/mcqmcproc.pdf

Text with more math than I give:

Monte Carlo Statistical Methods
Christian P. Robert, George Casella

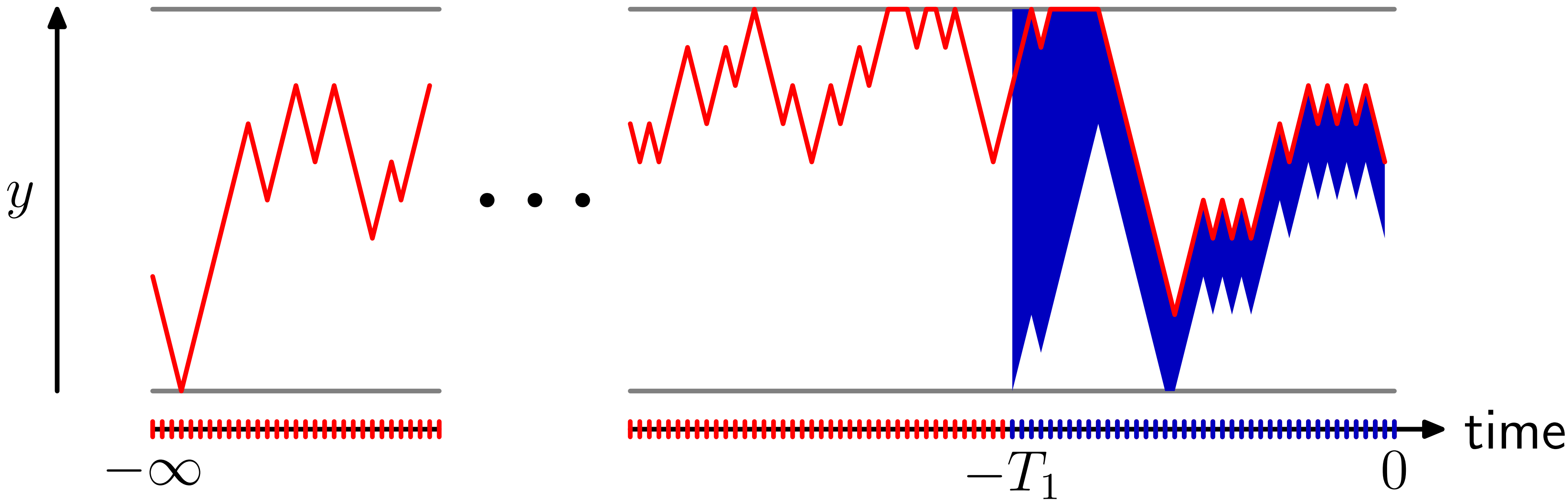
Exact sampling — *amazing* when it works

Exact sampling with MCMC



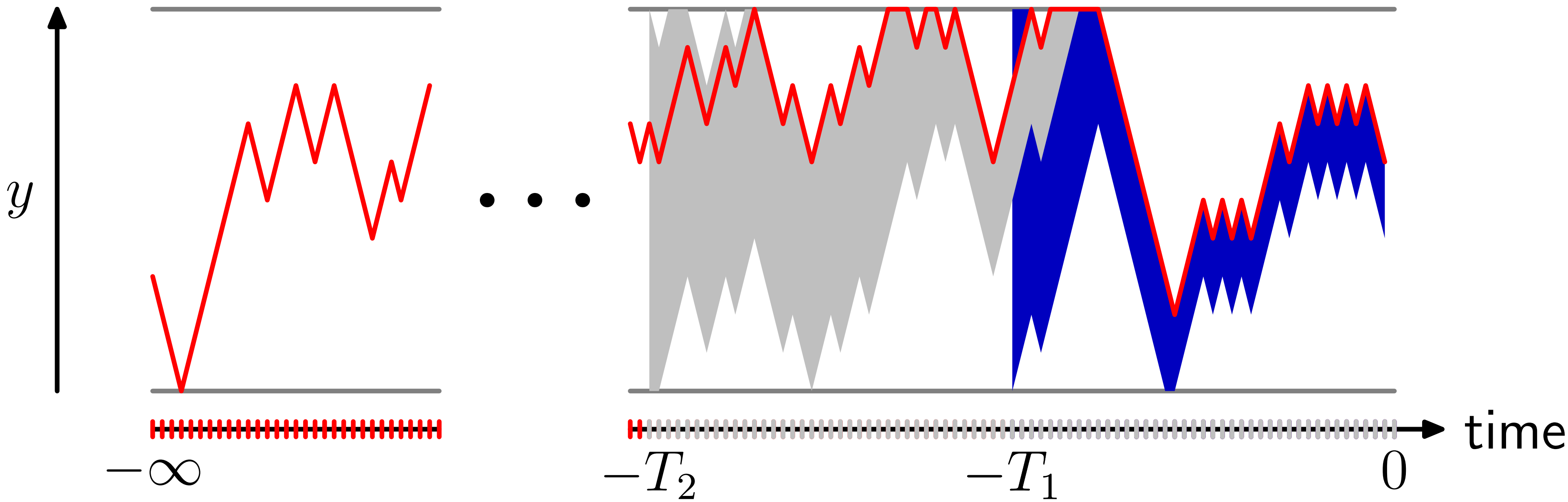
A chain that has run for ever

Exact sampling with MCMC



Try to find final state with finite number of random numbers

Exact sampling with MCMC



Takes a random amount of time.

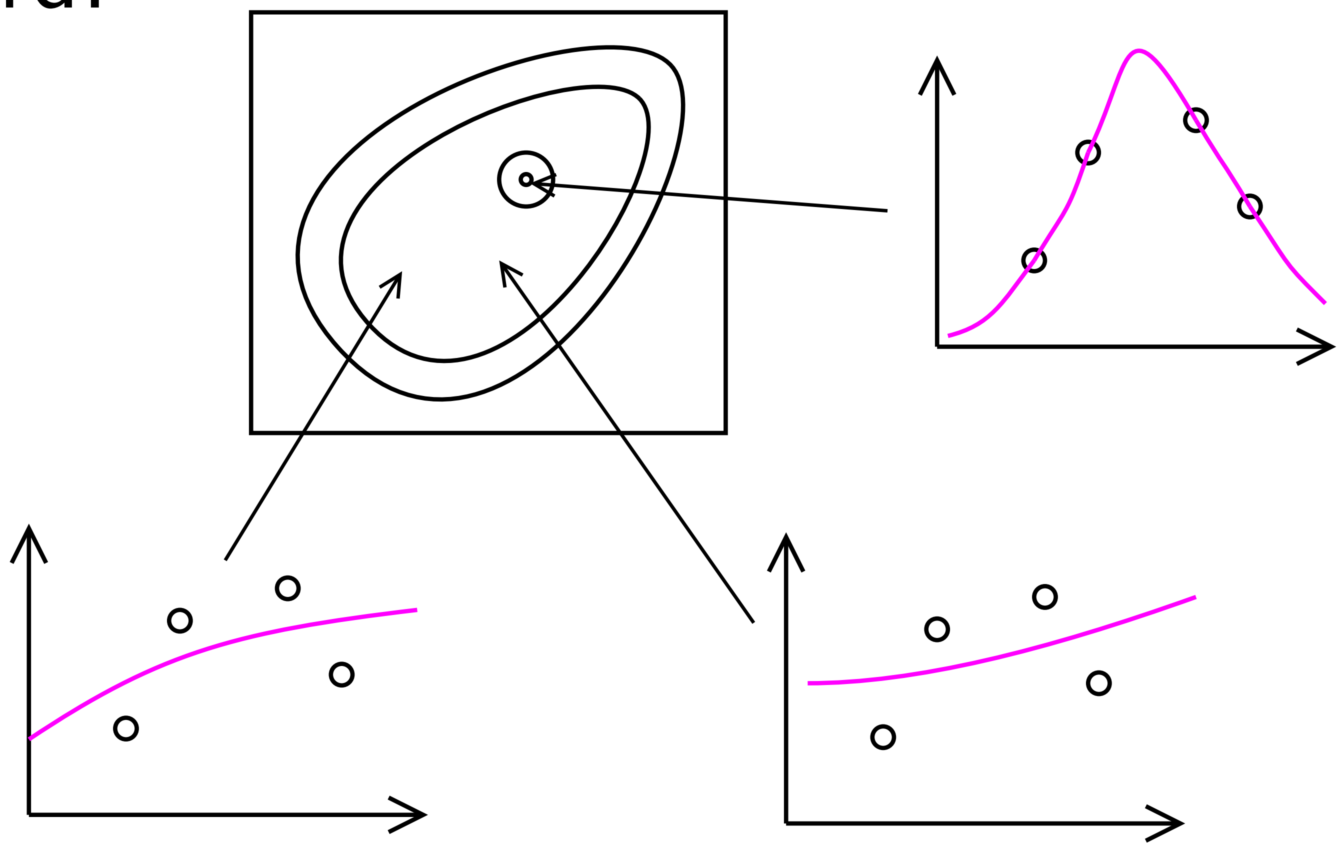
See `http://dbwilson.com/exact/`

(Google: "exact sampling" or "perfect sampling")

Building better chains

Come up with better proposals, Q?

Can be hard!



Many MCMC methods take a surprising approach. . .

Auxiliary variables

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

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

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

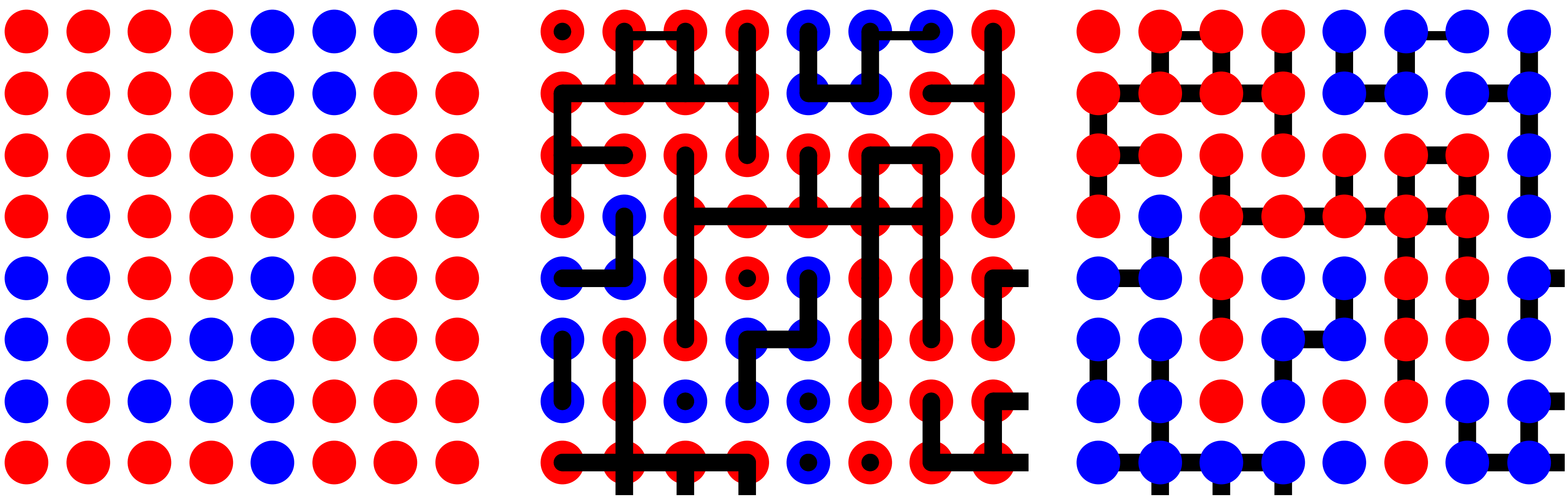
We might want to introduce v if:

- $P(x|v)$ and $P(v|x)$ are simple

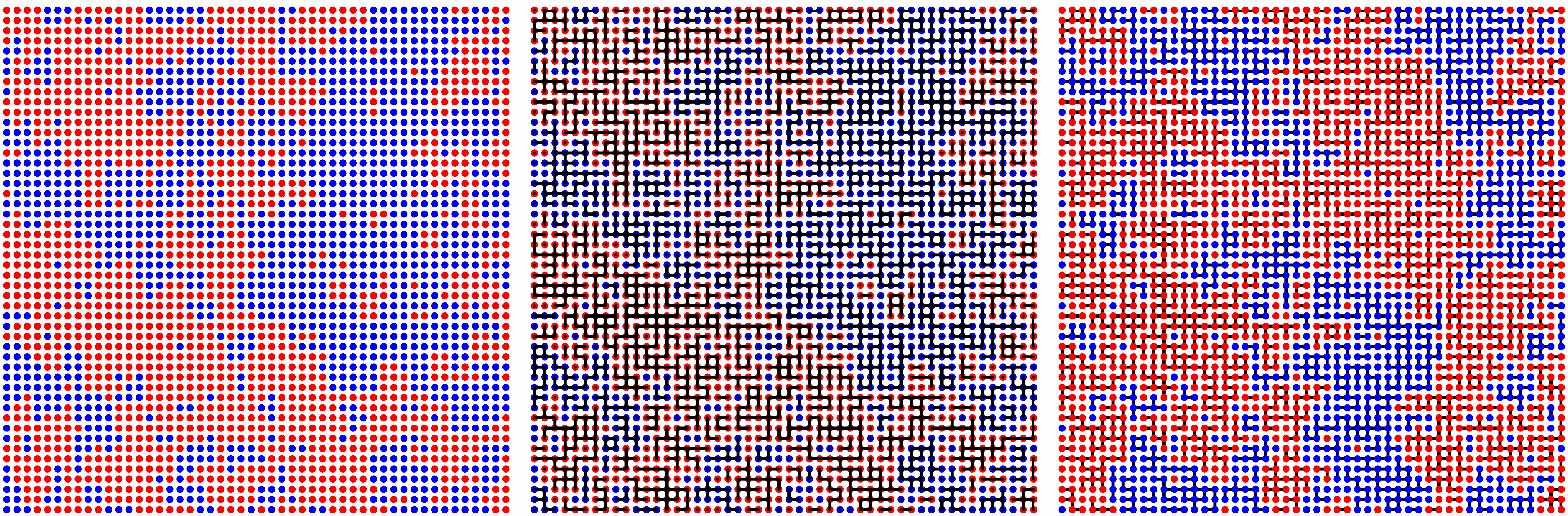
- $P(x,v)$ is otherwise easier to navigate

Swendsen–Wang (1987)

Seminal algorithm using auxiliary variables



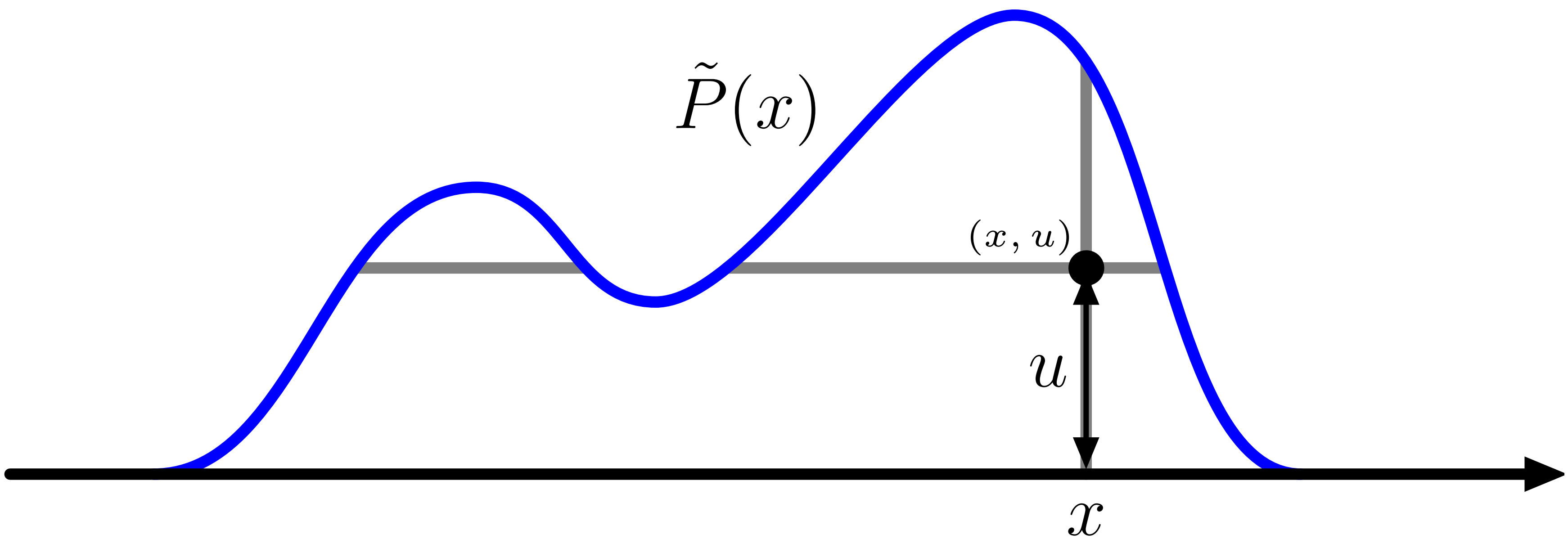
Swendsen–Wang (1987)



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 $\tilde{P}(x) \propto P(x)$

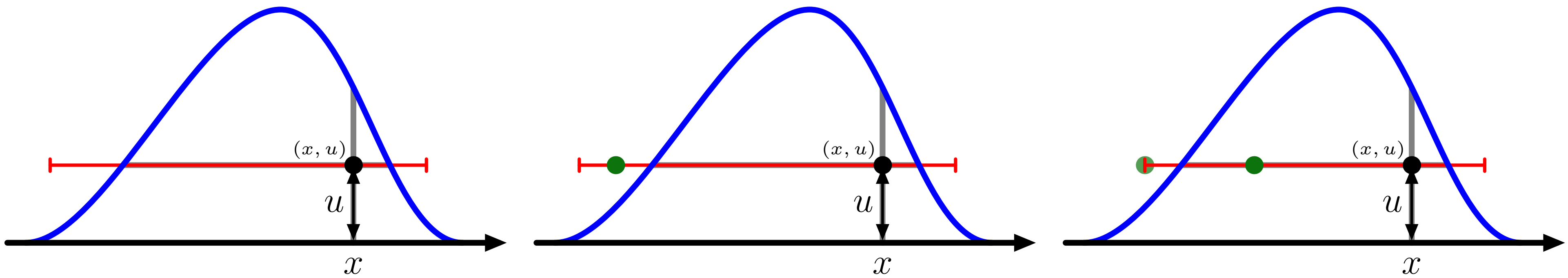


$$p(u|x) = \text{Uniform}[0, \tilde{P}(x)]$$

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

Slice sampling

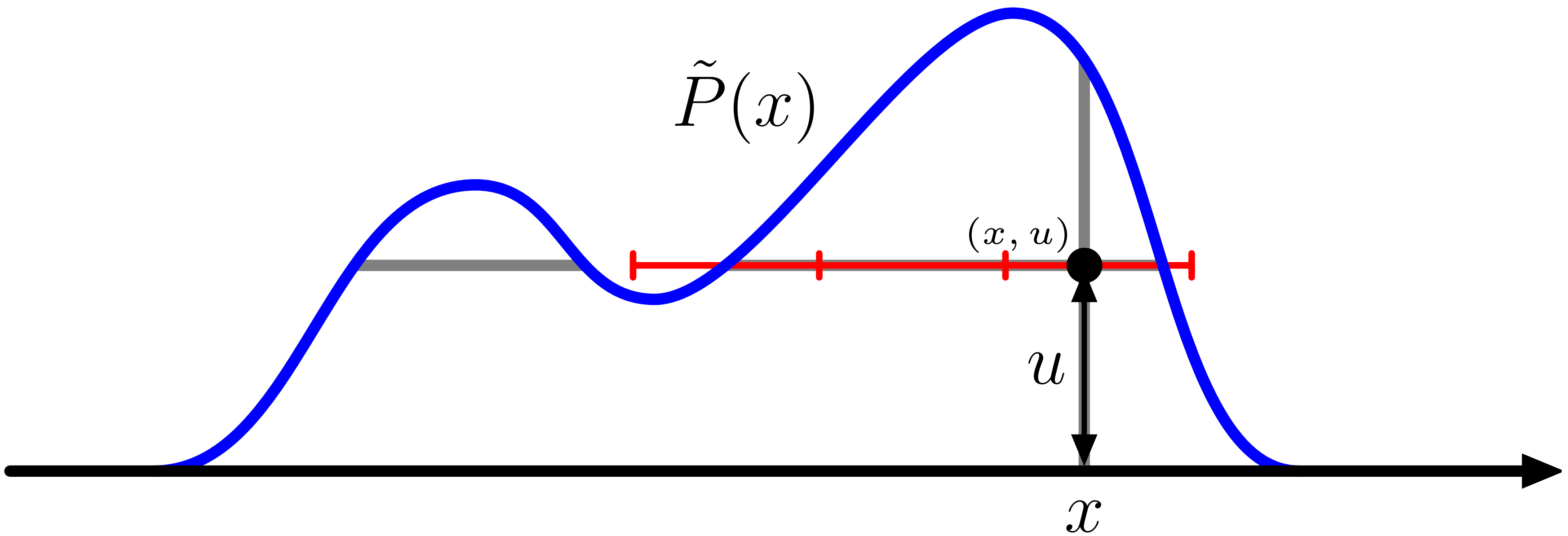
Unimodal conditionals



- 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
- Tweak params less important than Metropolis

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

Hamiltonian dynamics

Construct a landscape

Gravitational potential energy, E(x):

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

Roll a ball with velocity v

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

Recommended reading:

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

Summary of auxiliary variables

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

Some of my auxiliary representation work:

Doubly-intractable distributions

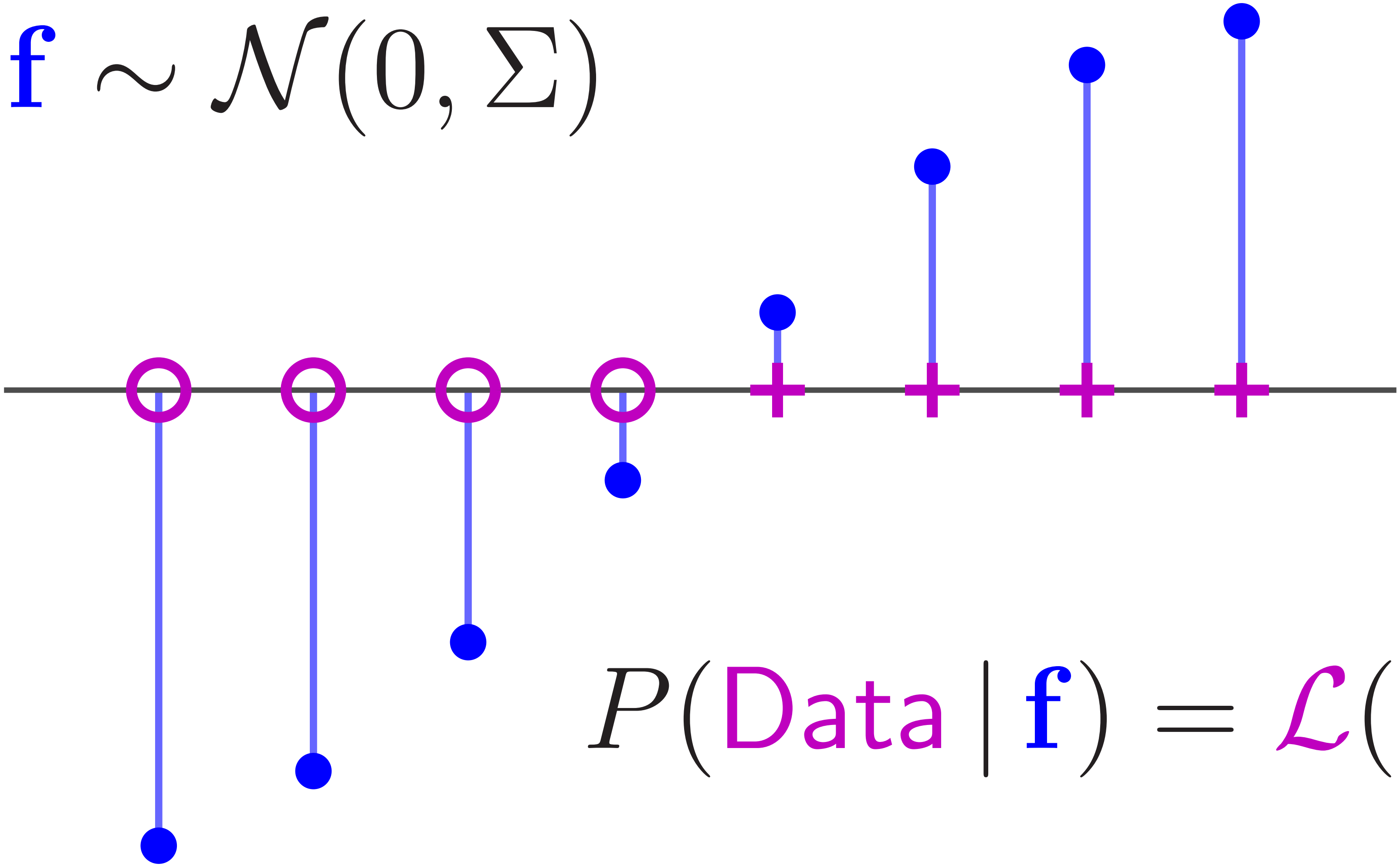
Population methods for better mixing (on parallel hardware)

Being robust to bad random number generators

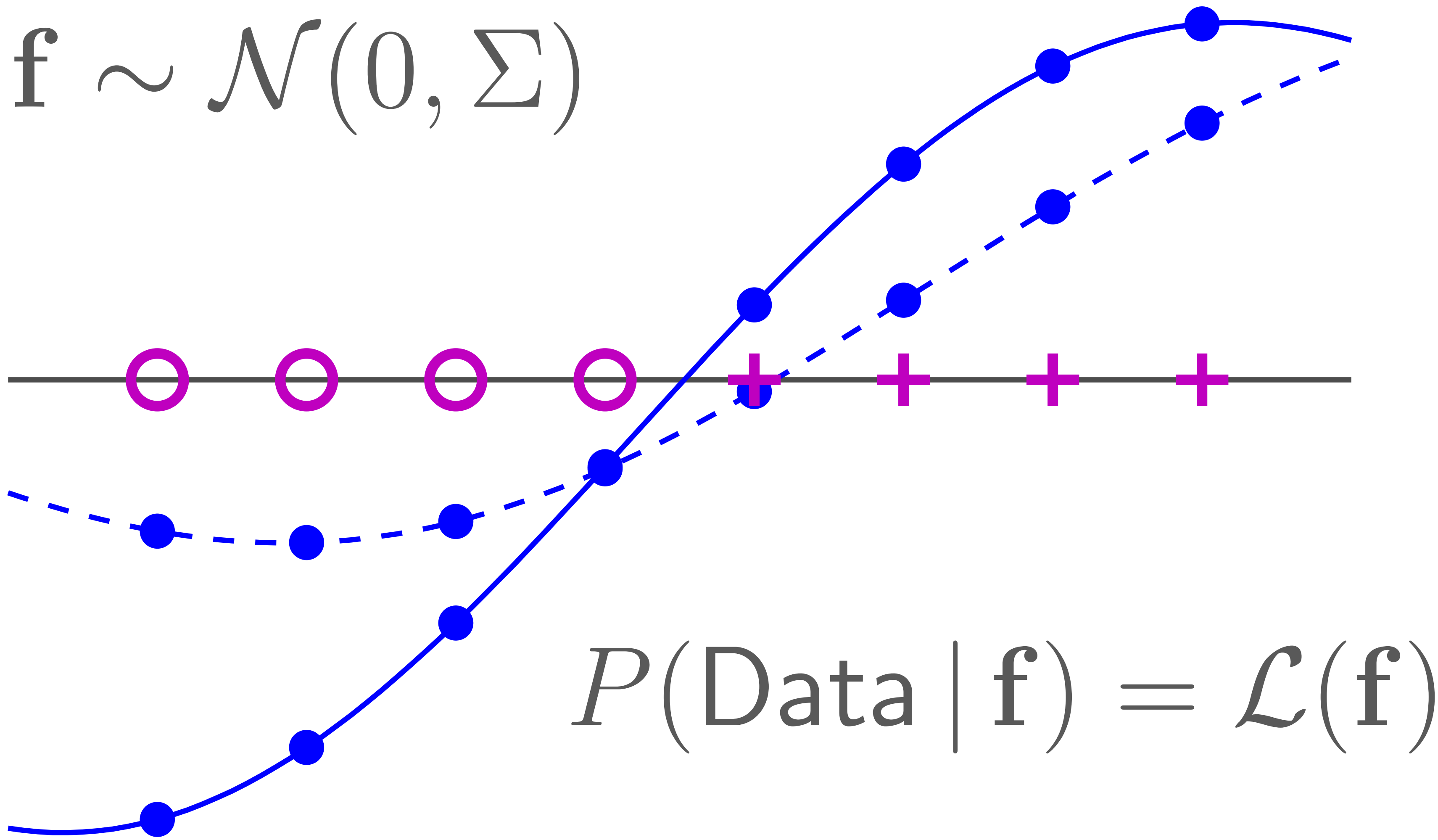
Recent slice-sampling work

Data

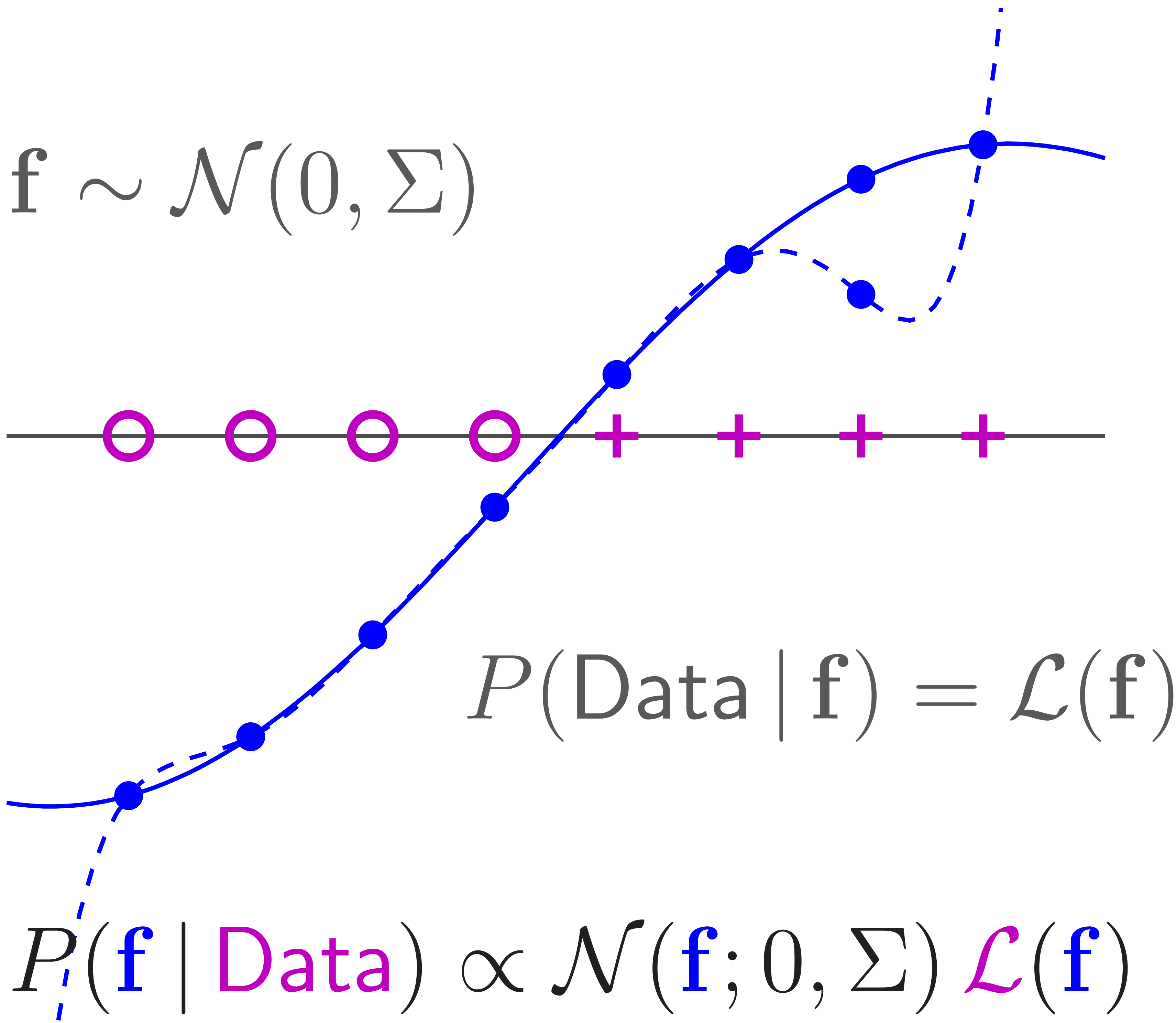
$$\mathbf{f} \sim \mathcal{N}(0, \Sigma)$$



$$P(\mathsf{Data} \,|\, \mathbf{f}) = \mathcal{L}(\mathbf{f})$$



$$P(\mathbf{f} \mid \text{Data}) \propto \mathcal{N}(\mathbf{f}; 0, \Sigma)\, \mathcal{L}(\mathbf{f})$$



$\mathbf{f} \sim \mathcal{N}(0, \Sigma)$
$P(\mathsf{Data} \mid \mathbf{f}) = \mathcal{L}(\mathbf{f})$
$P(\mathbf{f} \mid \mathsf{Data}) \propto \mathcal{N}(\mathbf{f}; 0, \Sigma)\, \mathcal{L}(\mathbf{f})$

An update for Gaussian priors

Target to leave invariant: $P^\star(\mathbf{f}) \propto \mathcal{N}(0, \Sigma)\, L(\mathbf{f})$

Propose:

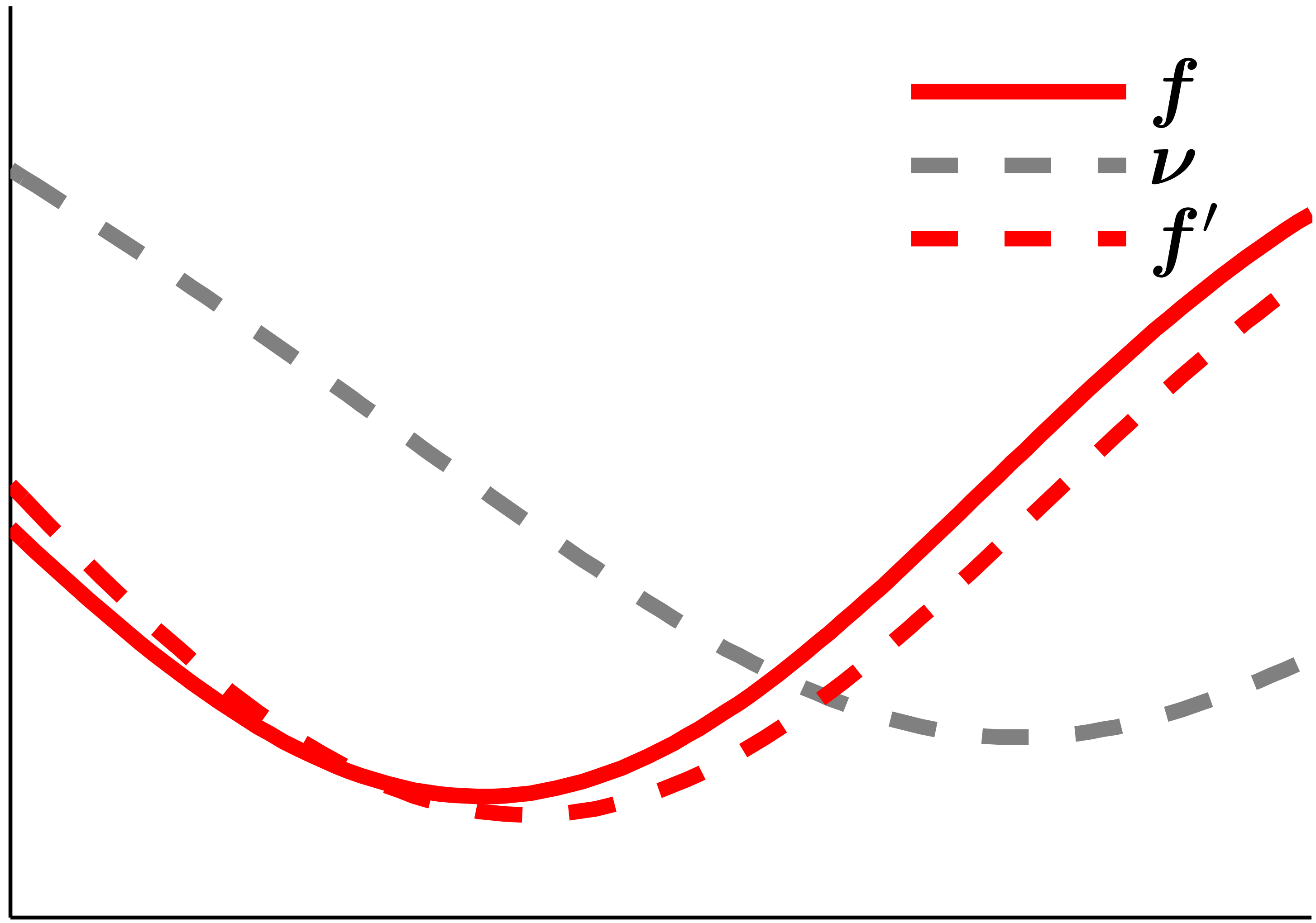
$$\mathbf{f}' \leftarrow \alpha\, \mathbf{f} + \sqrt{1-\alpha^2}\, \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim \mathcal{N}(0, \Sigma)$$

Accept/Reject:

Accept $\mathbf{f}'$ with probability $\min\!\left(1,\ \frac{L(\mathbf{f}')}{L(\mathbf{f})}\right)$

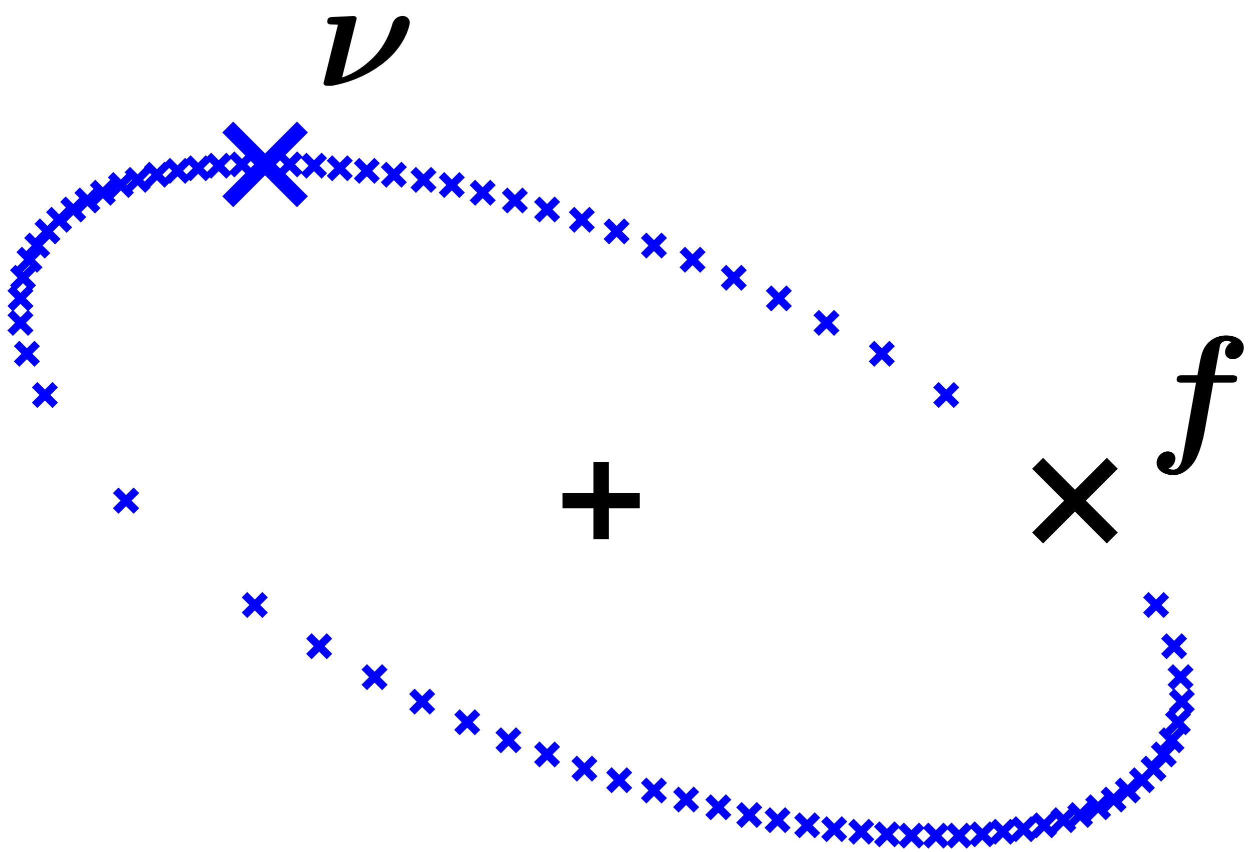
Neal (1999 Valencia paper)

Update for GP functions



$$\mathbf{f}' \leftarrow \alpha\, \mathbf{f} + \sqrt{1-\alpha^2}\, \boldsymbol{\nu}, \quad \boldsymbol{\nu} \sim \mathcal{N}(0, \Sigma)$$

Ellipse of combinations

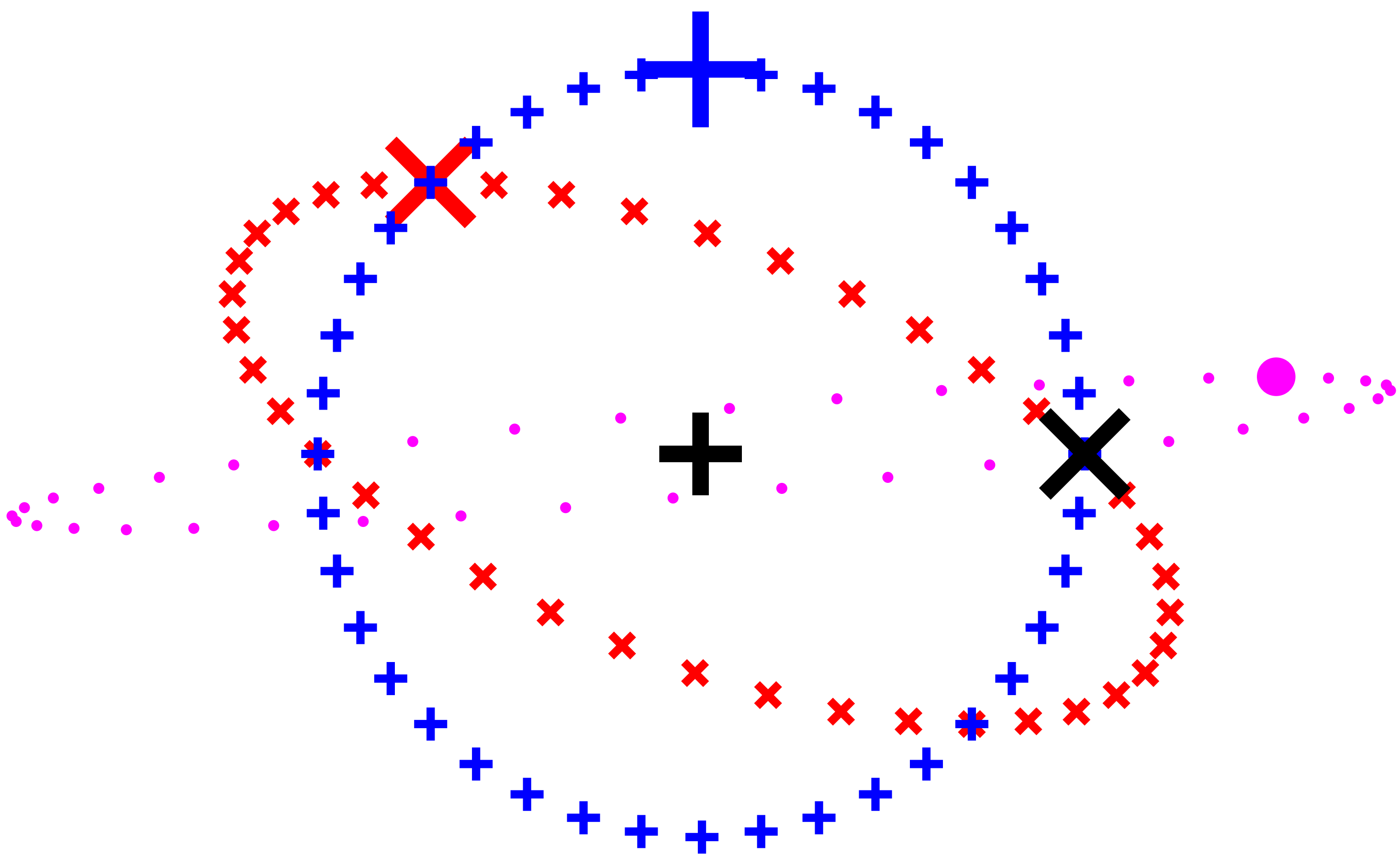


$$\mathbf{f}' \leftarrow \alpha\,\mathbf{f} \pm \sqrt{1-\alpha^2}\,\boldsymbol{\nu}, \quad \alpha \in [-1, 1]$$

Angular parameterization

Locus of points with correct marginal covariance:

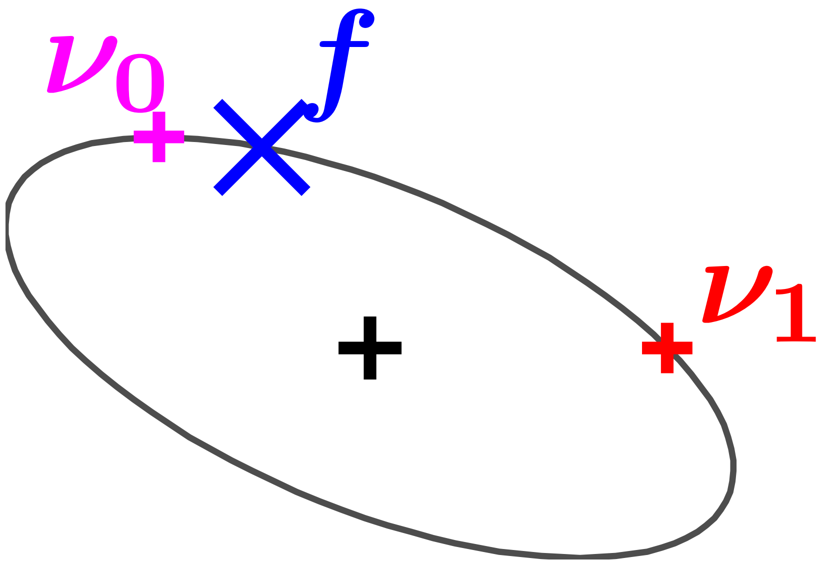
$$\mathbf{f}' = \mathbf{f}\cos\beta + \boldsymbol{\nu}\sin\beta$$



Auxiliary variable model

Prior:

$$\boldsymbol{\nu}_0 \sim \mathcal{N}(0, \Sigma)$$
$$\boldsymbol{\nu}_1 \sim \mathcal{N}(0, \Sigma)$$
$$\beta \sim \mathrm{Uniform}[0, 2\pi]$$
$$\mathbf{f} = \boldsymbol{\nu}_0 \sin\beta + \boldsymbol{\nu}_1 \cos\beta$$



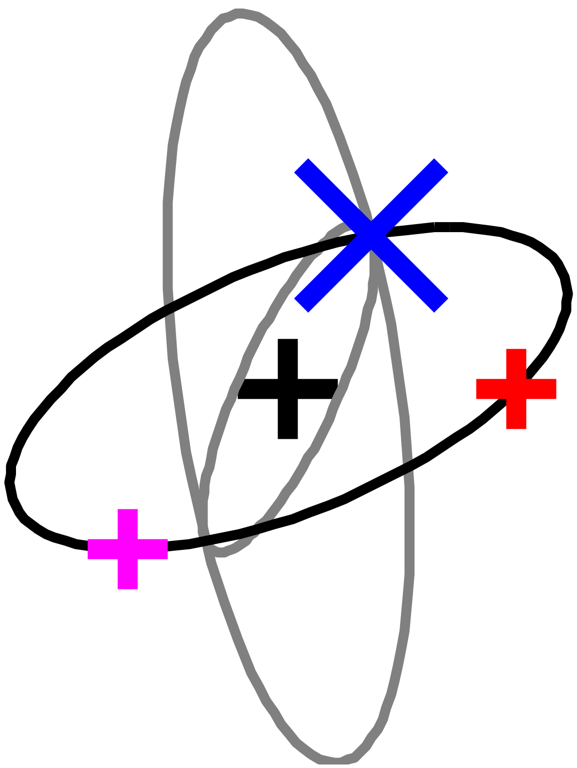
Likelihood: $L(\mathbf{f}(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \beta))$

Posterior: $P^\star(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \beta) \propto \mathcal{N}(\boldsymbol{\nu}_0; 0, \Sigma)\, \mathcal{N}(\boldsymbol{\nu}_1; 0, \Sigma)\, L(\mathbf{f}(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \beta))$

MCMC in Auxiliary model

Operator 1: resample $\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \beta \,|\, \mathbf{f} \;\sim\; P(\beta \,|\, \mathbf{f})\, P(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1 \,|\, \beta, \mathbf{f})$:

$$\begin{aligned} \beta &\sim \mathrm{Uniform}[0, 2\pi] \\ \boldsymbol{\nu} &\sim \mathcal{N}(0, \Sigma) \\ \boldsymbol{\nu}_0 &\leftarrow \mathbf{f}\sin\beta + \boldsymbol{\nu}\cos\beta \\ \boldsymbol{\nu}_1 &\leftarrow \mathbf{f}\cos\beta - \boldsymbol{\nu}\sin\beta \end{aligned}$$



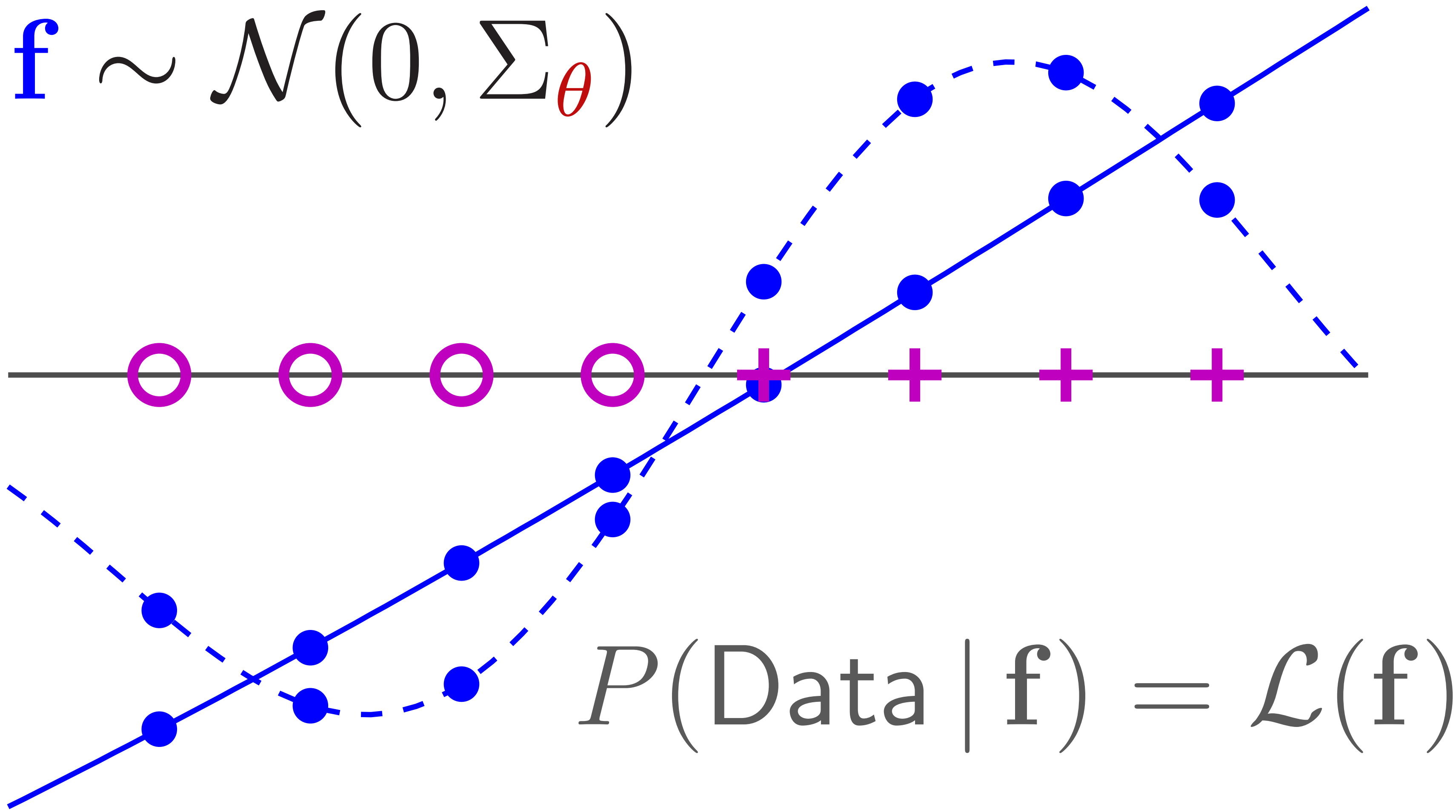
Operator 2: slice sample β for fixed $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}_1$.

Both operators leave the target distribution stationary:

$$P^\star(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \beta) \propto \mathcal{N}(\boldsymbol{\nu}_0; 0, \Sigma)\, \mathcal{N}(\boldsymbol{\nu}_1; 0, \Sigma)\, L(\mathbf{f}(\boldsymbol{\nu}_0, \boldsymbol{\nu}_1, \beta))$$

$$\theta \sim p_h$$

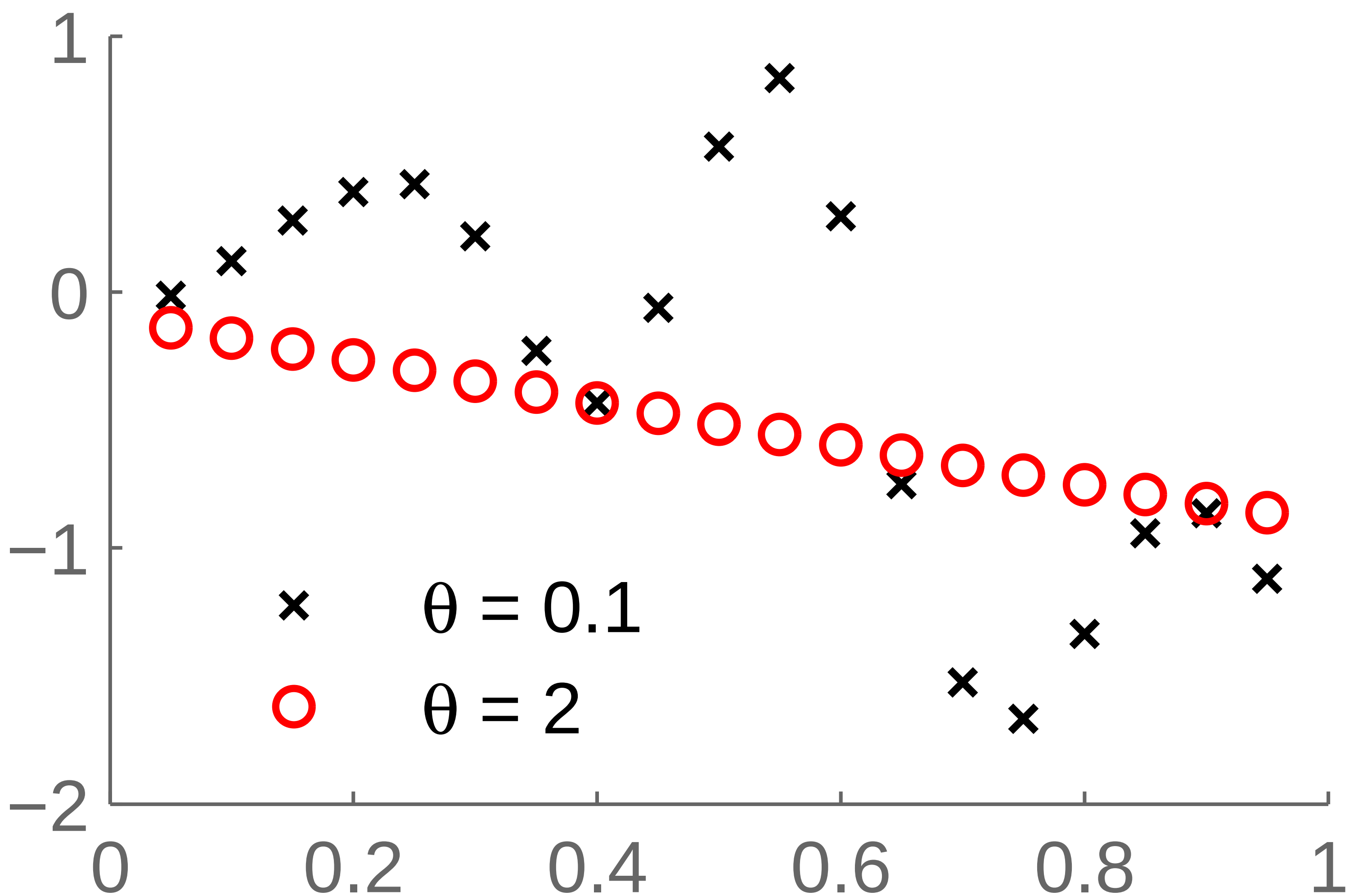
$$\mathbf{f} \sim \mathcal{N}(0, \Sigma_\theta)$$



$$P(\mathbf{f}, \theta \mid \mathsf{D}) \propto p(\theta)\, \mathcal{N}(\mathbf{f}; 0, \Sigma_\theta)\, \mathcal{L}(\mathbf{f})$$

We're not mode-searching

Start at **Red** values. Propose short scale $\theta = 0.1$.



Red values are $> 500\times$ more probable than **Black**

#include

`http://videolectures.net/nips2010_murray_ssc/`

(a talk on sampling hyper-parameters in Gaussian processes)

Summary

Please be careful running MCMC

Try Gibbs or simple Metropolis, then:

— Try to find a better Q, e.g., data-driven MCMC

— Try to find a better representation

— Auxiliary variables often useful

Remember operators can be concatenated

(Mix in simple updates with fancy ones)

Combining operators

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

$$
\begin{aligned}
&x_0 \sim P^\star(x) \\
&x_1 \sim T_a(x_1 \leftarrow x_0) \qquad && P(x_1) \;=\; \textstyle\sum_{x_0} T_a(x_1 \leftarrow x_0) P^\star(x_0) \;=\; P^\star(x_1) \\
&x_2 \sim T_b(x_2 \leftarrow x_1) \qquad && P(x_2) \;=\; \textstyle\sum_{x_1} T_b(x_2 \leftarrow x_1) P^\star(x_1) \;=\; P^\star(x_2) \\
&x_3 \sim T_c(x_3 \leftarrow x_2) \qquad && P(x_3) \;=\; \textstyle\sum_{x_1} T_c(x_3 \leftarrow x_2) P^\star(x_2) \;=\; P^\star(x_3) \\
&\quad \ldots && \quad \ldots
\end{aligned}
$$

— 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

Finding normalizers is hard

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

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

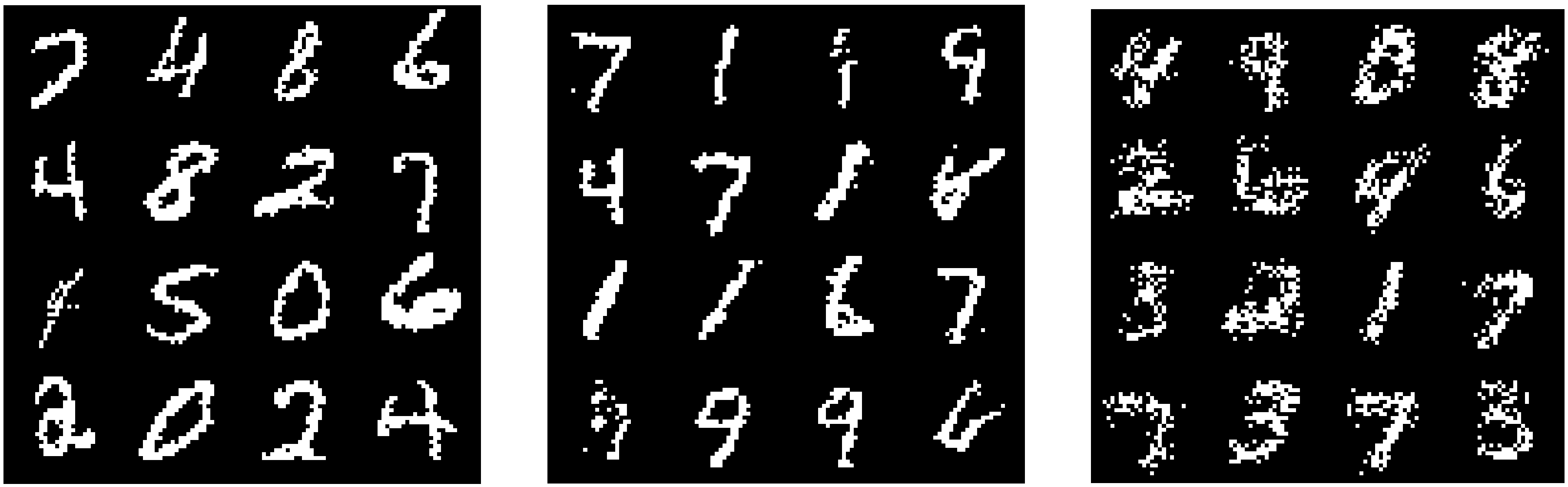
. . . usually has huge variance

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:

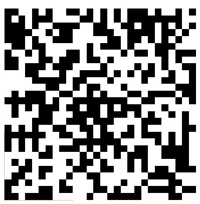
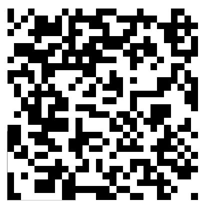
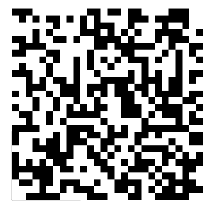
— $28 \times 28 = 784$ binary visible variables

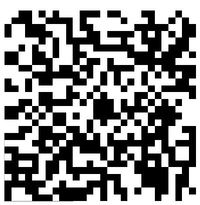
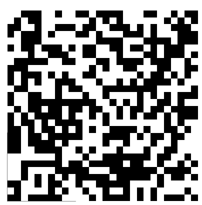
— 500 binary hidden variables

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

Simple Importance Sampling

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

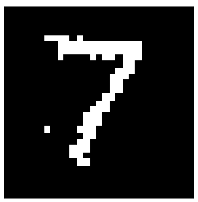
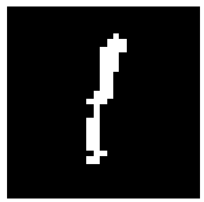
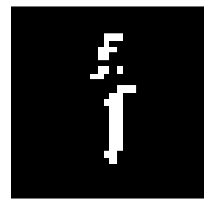
$\mathbf{x}^{(1)} =$ , $\mathbf{x}^{(2)} =$ , $\mathbf{x}^{(3)} =$ ,

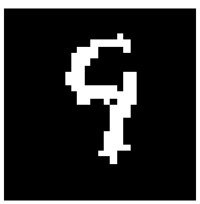
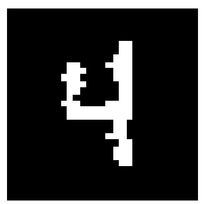
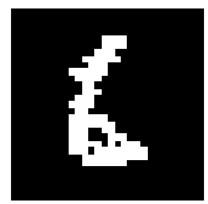
$\mathbf{x}^{(4)} =$ , $\mathbf{x}^{(5)} =$ , $\mathbf{x}^{(6)} =$ , ...

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

"Posterior" Sampling

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

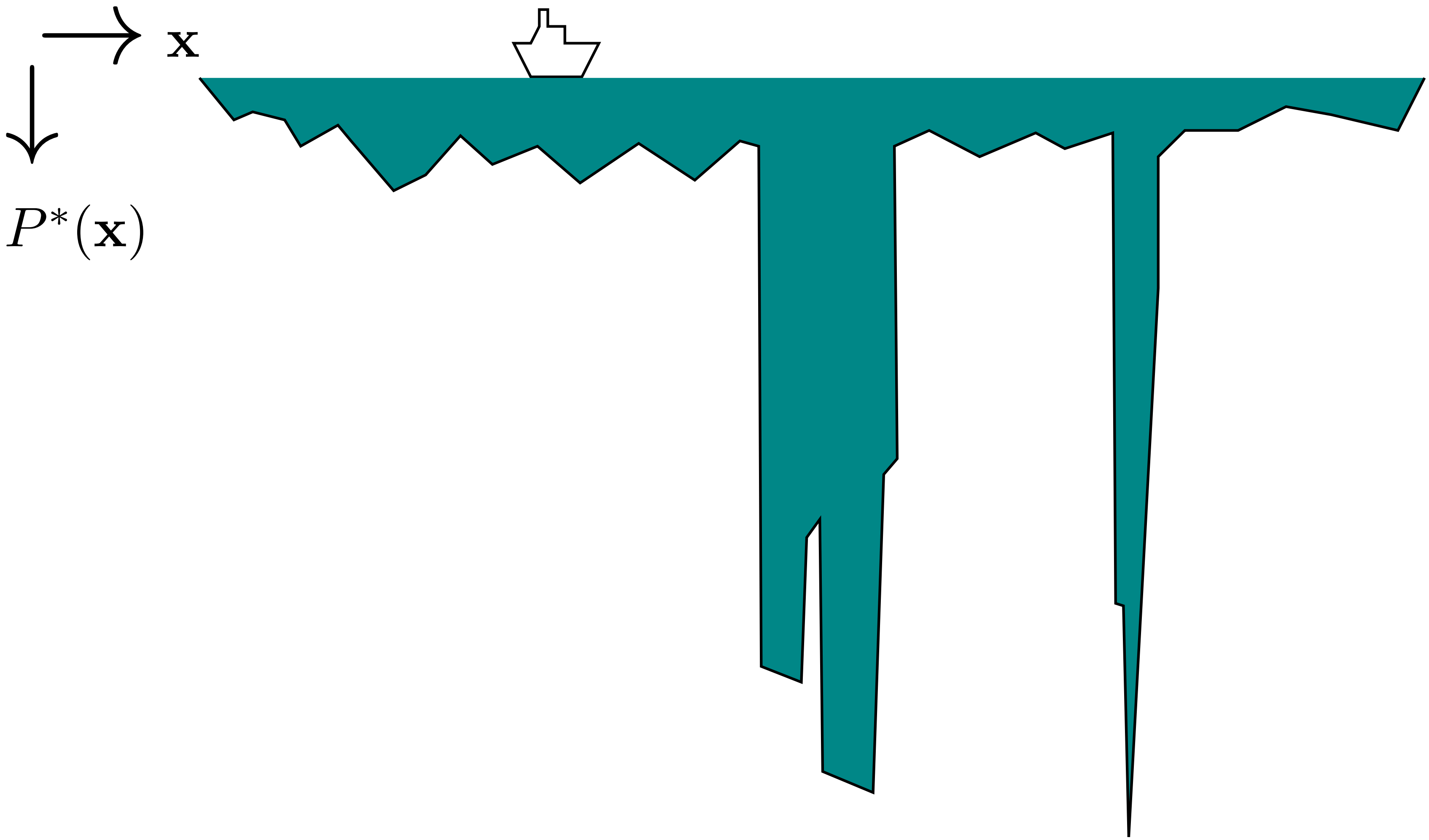
$\mathbf{x}^{(1)} =$ , $\mathbf{x}^{(2)} =$ , $\mathbf{x}^{(3)} =$ ,

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

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

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

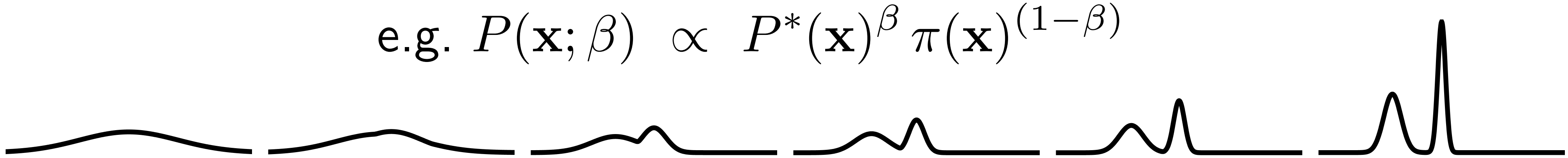
Finding a Volume



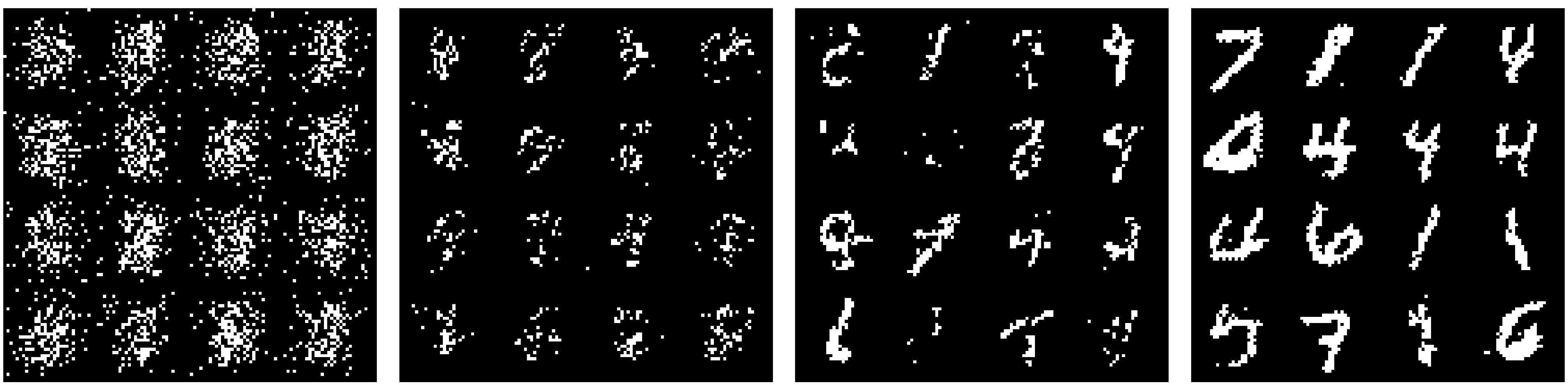
Lake analogy and figure from MacKay textbook (2003)

Annealing / Tempering

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



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

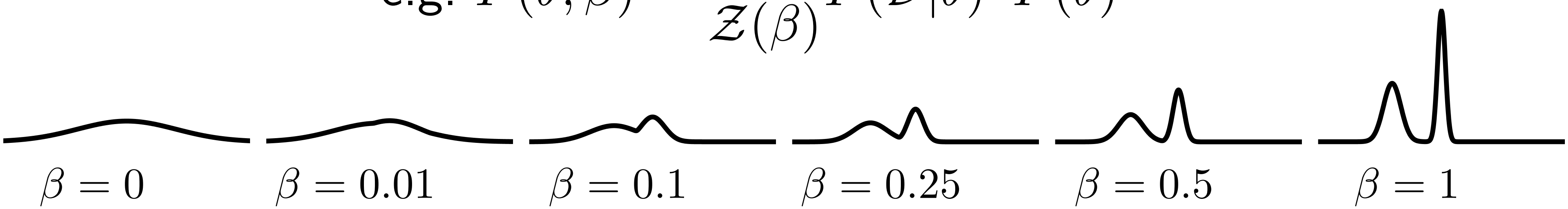


$1/\beta \;=\;$ "temperature"

Using other distributions

Chain between posterior and prior:

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



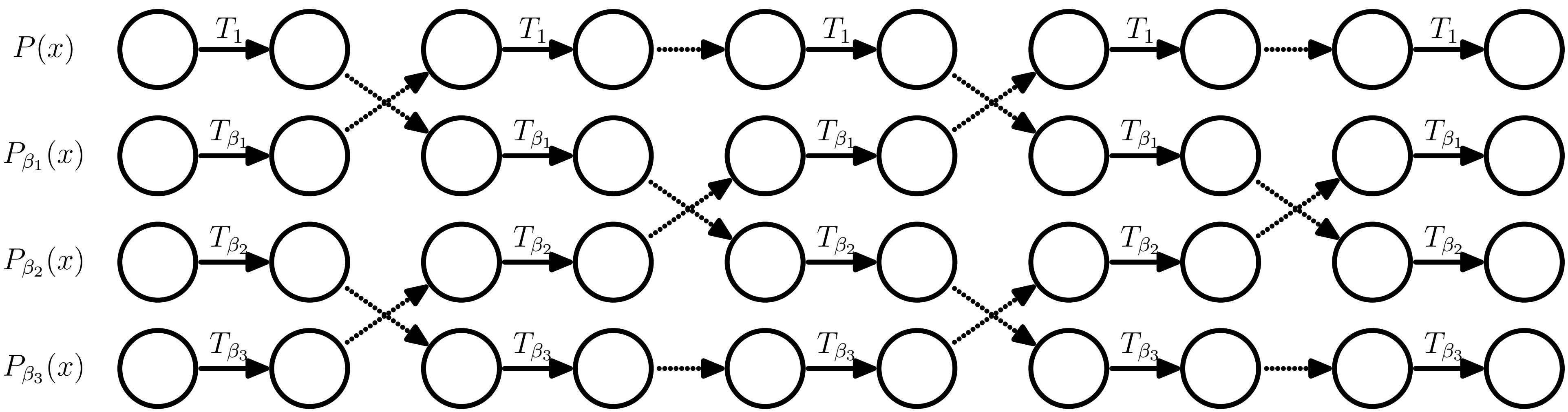
Advantages:

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

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

Parallel tempering

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

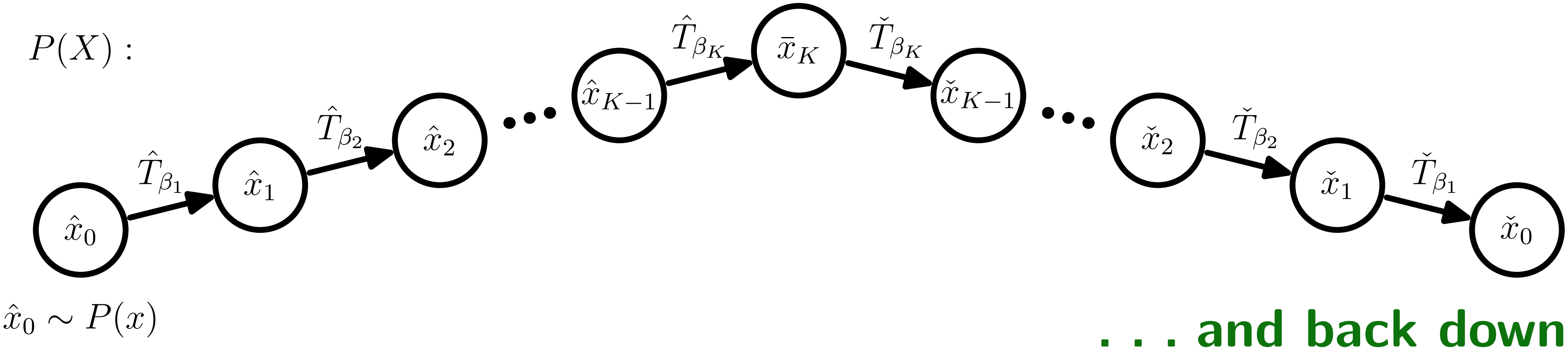


Problems / trade-offs:

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

Tempered transitions

Drive temperature up. . .



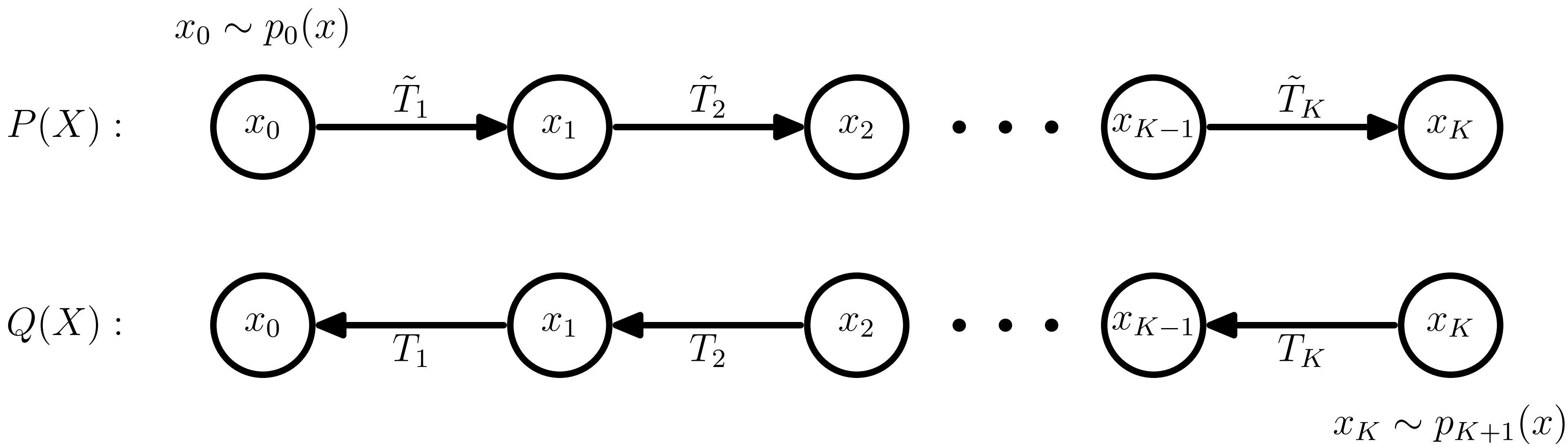
. . . and back down

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

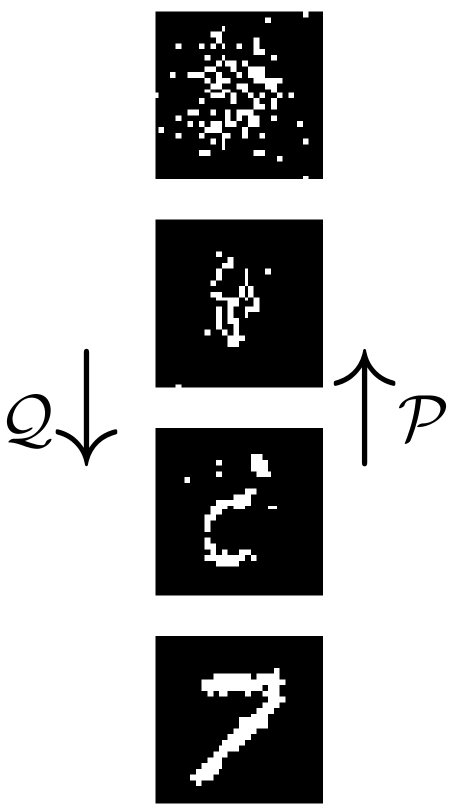


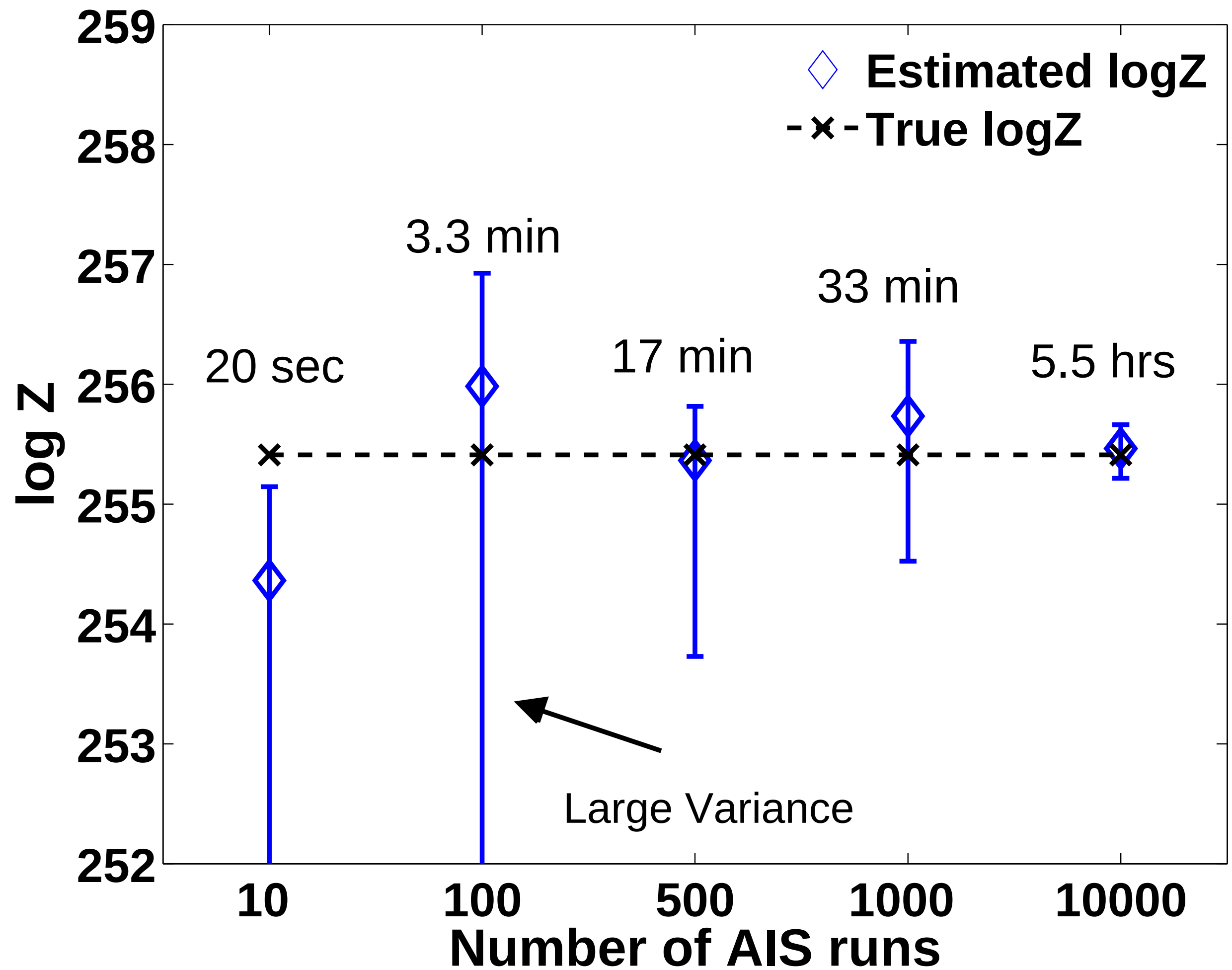
$$\mathcal{P}(X) = \frac{P^*(\mathbf{x}_K)}{\mathcal{Z}} \prod_{k=1}^{K} \widetilde{T}_k(\mathbf{x}_{k-1}; \mathbf{x}_k), \qquad \mathcal{Q}(X) = \pi(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k; \mathbf{x}_{k-1})$$

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 some estimators of $\mathcal{Z}$

Methods must be *good* at exploring the distribution

So watch these approaches for general use on the hardest problems.

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/` `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`