Markov chain Monte Carlo

Roadmap:

— Monte Carlo basics

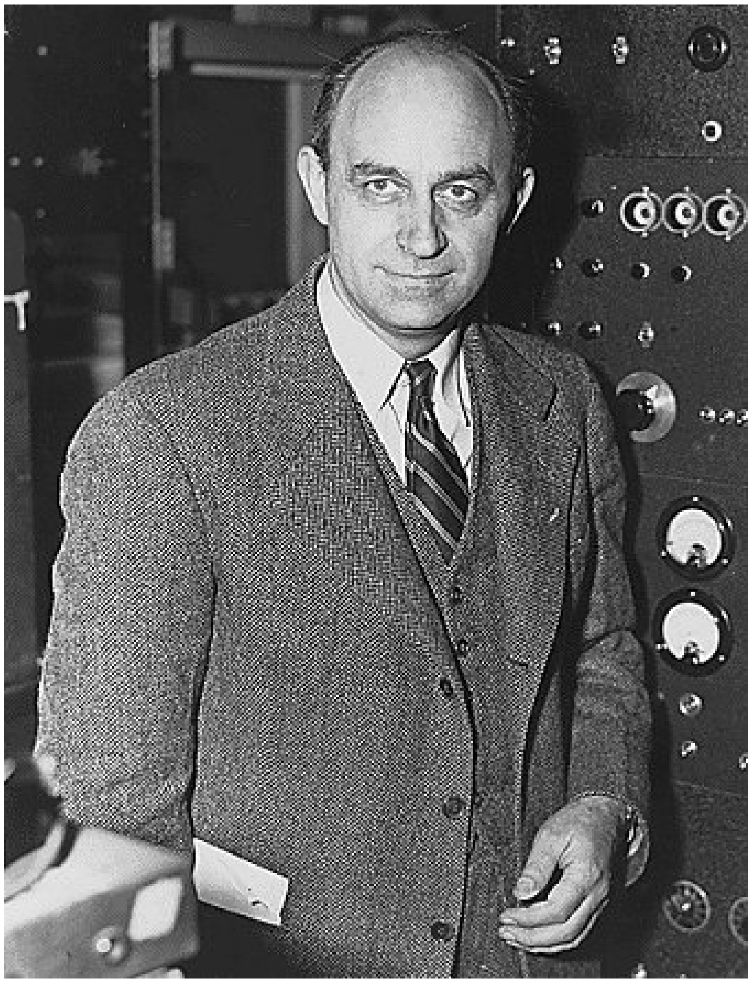
— What is MCMC?

— Gibbs and Metropolis–Hastings

Iain Murray

`http://iainmurray.net/`

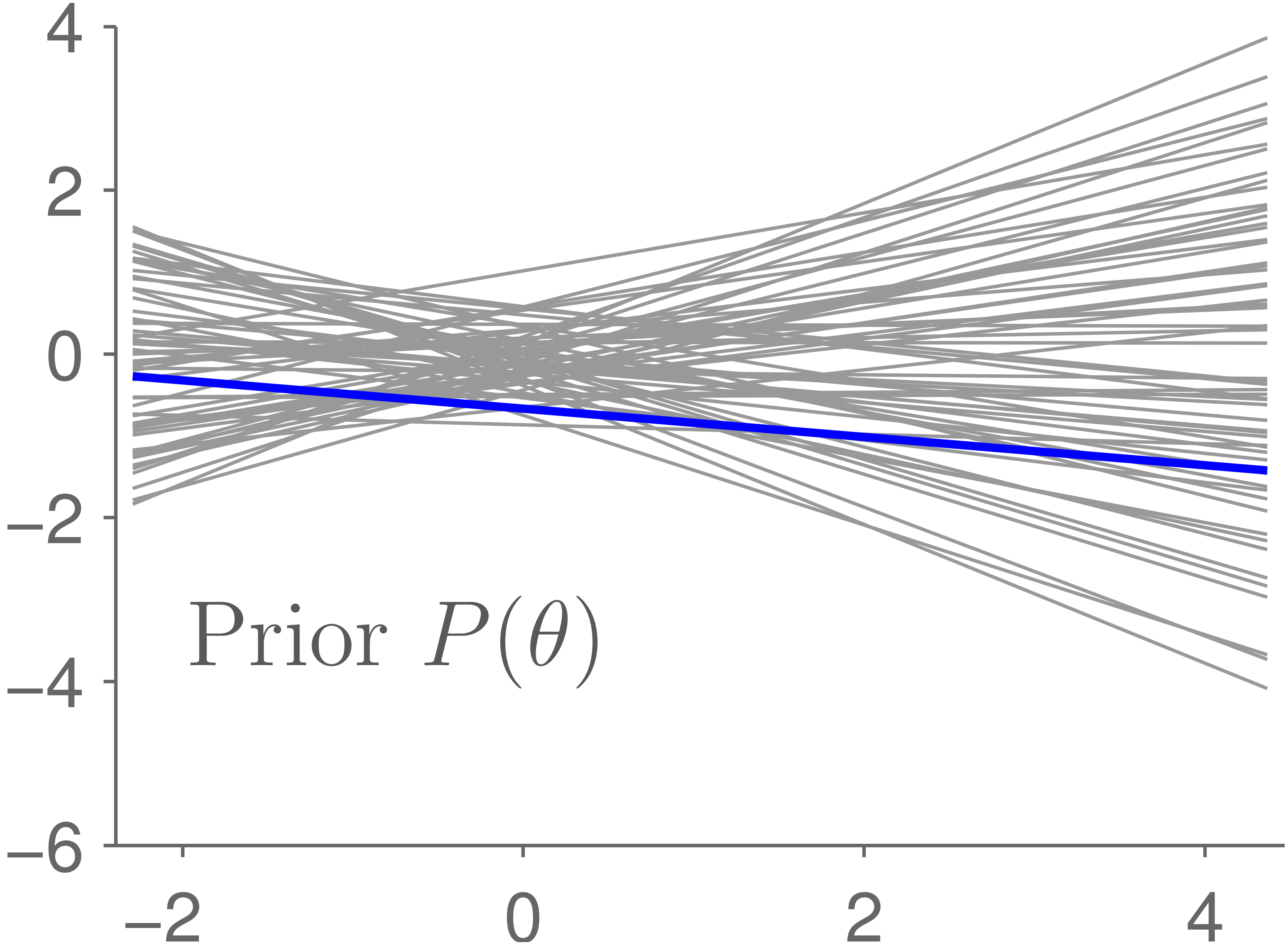
Monte Carlo and Insomnia



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!

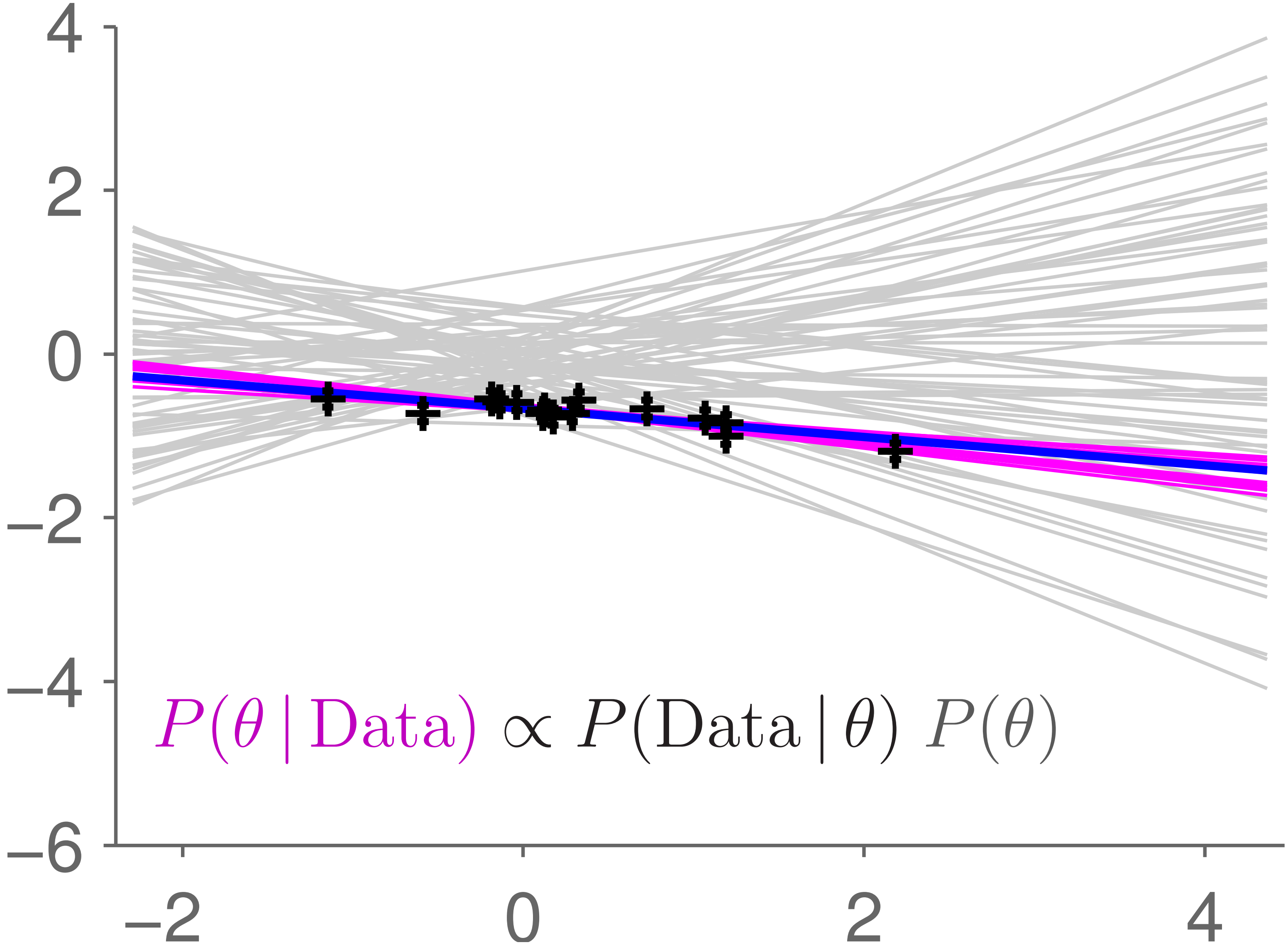
—*The beginning of the Monte Carlo method*, N. Metropolis

Linear Regression: Prior



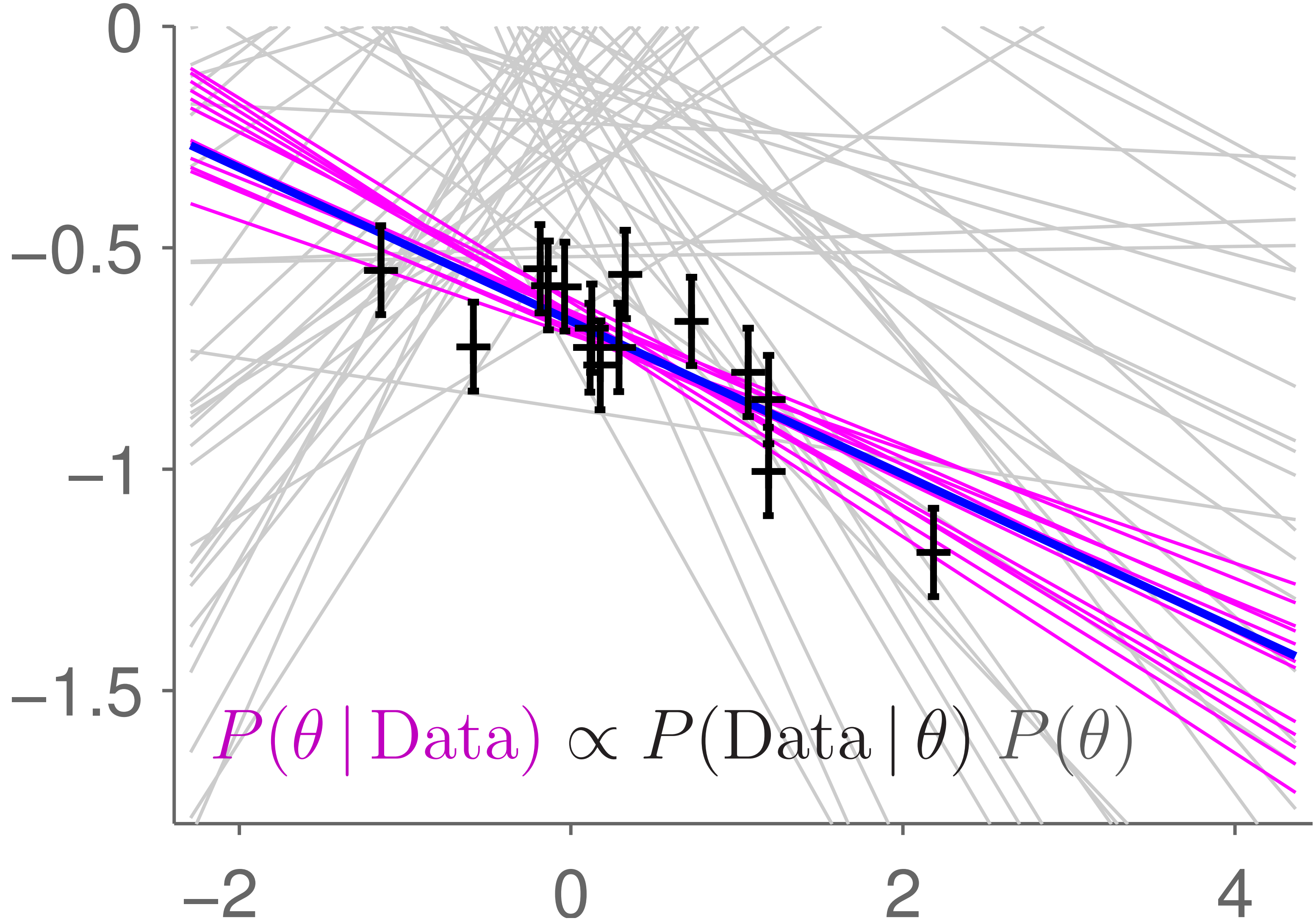
Input $\rightarrow$ output mappings considered plausible before seeing data.

Linear Regression: Posterior



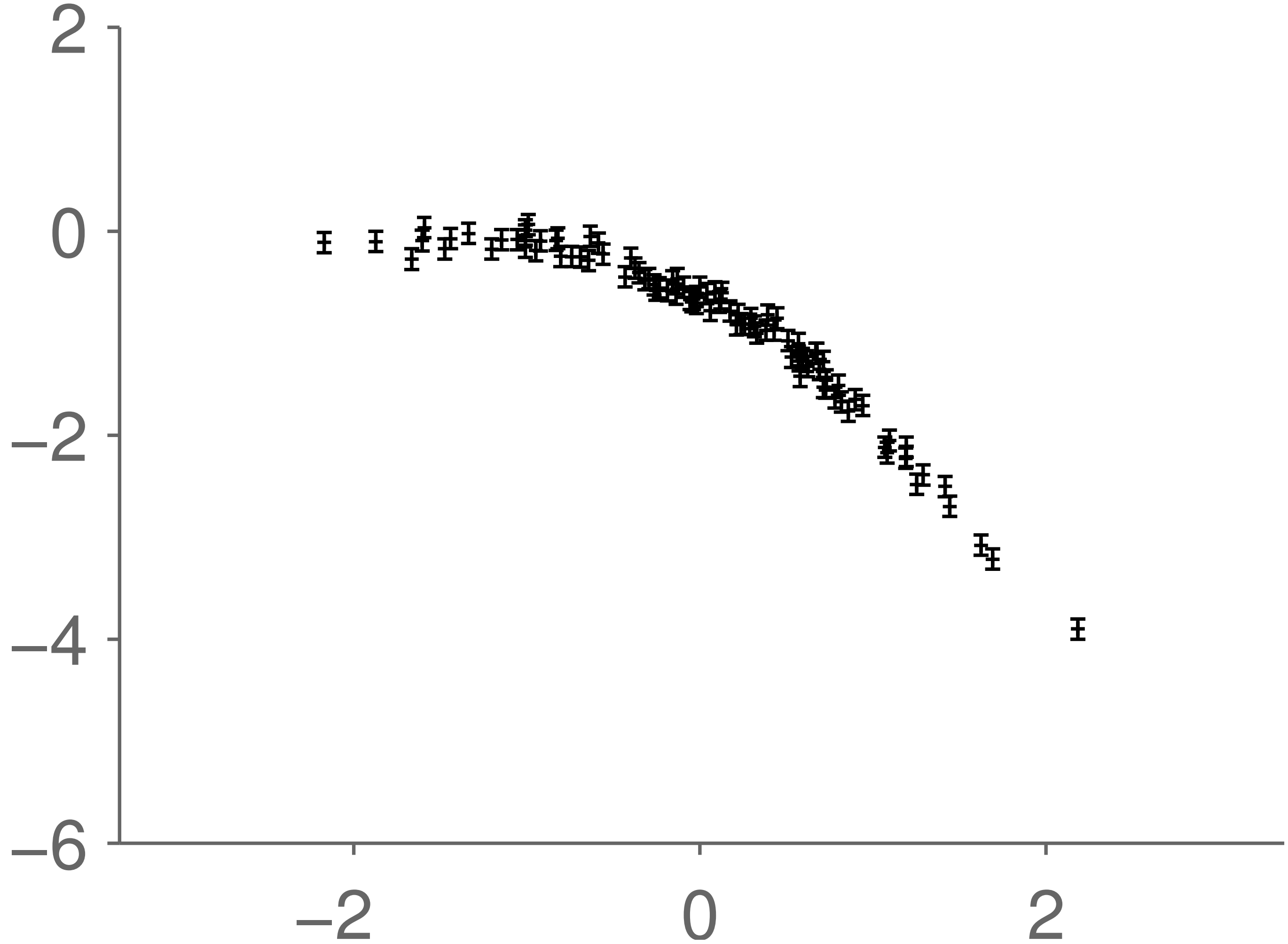
Posterior much more compact than prior.

Linear Regression: Posterior



Draws from posterior. Non-linear error envelope. Possible explanations linear.

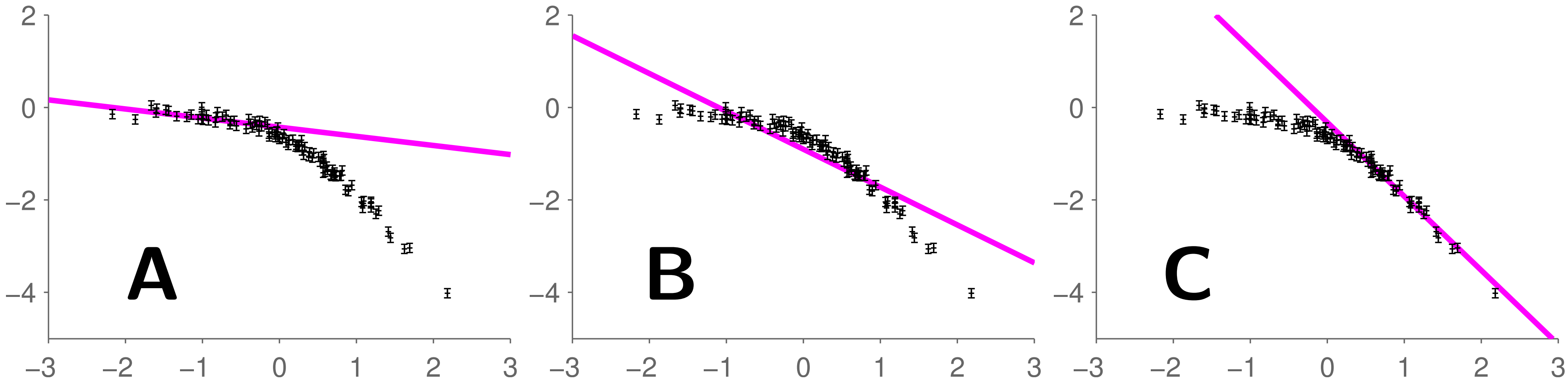
Model mismatch



What will Bayesian linear regression do?

Quiz

Given a (wrong) linear assumption, which explanations are typical of the posterior distribution?

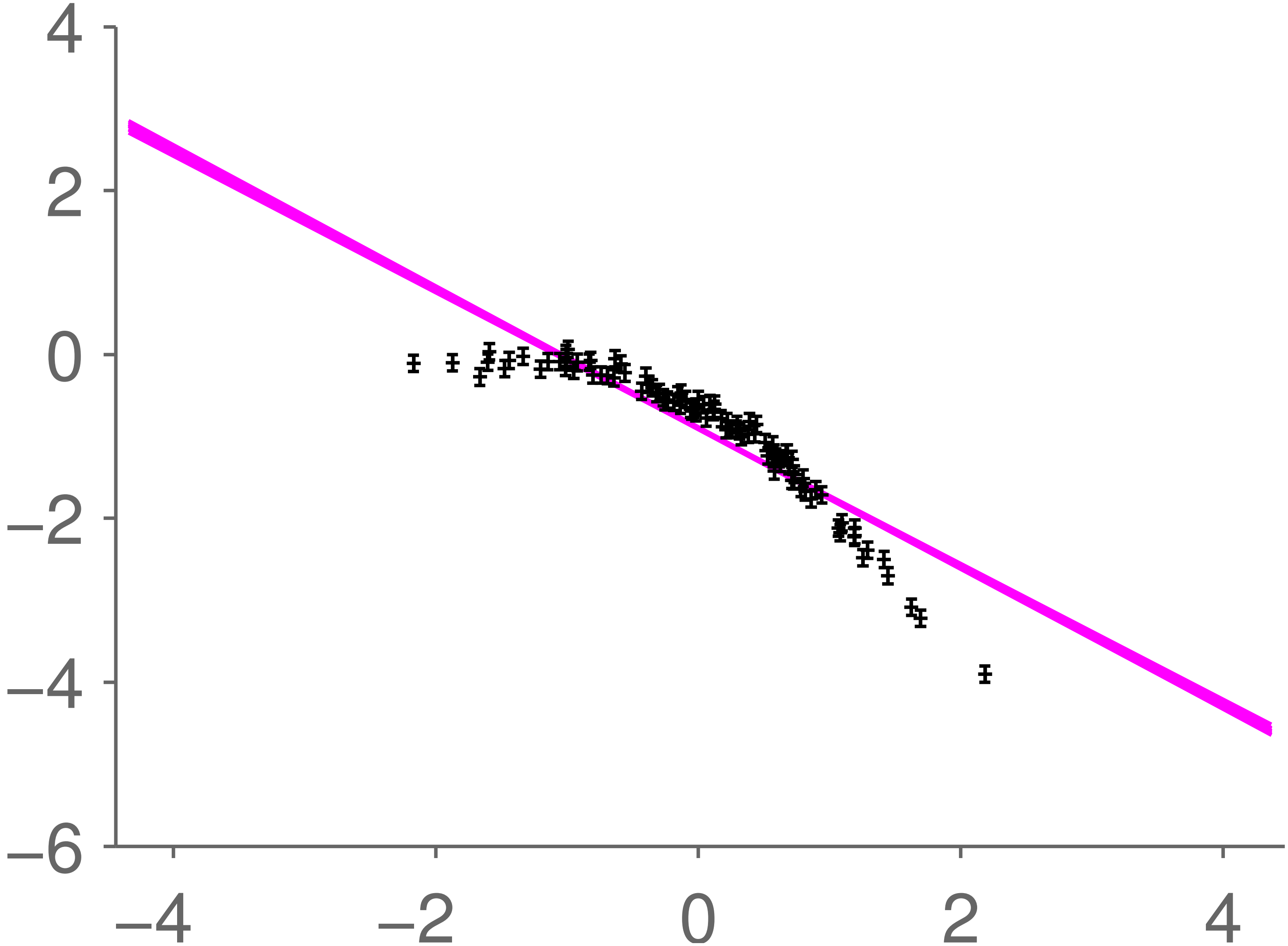


D All of the above

E None of the above

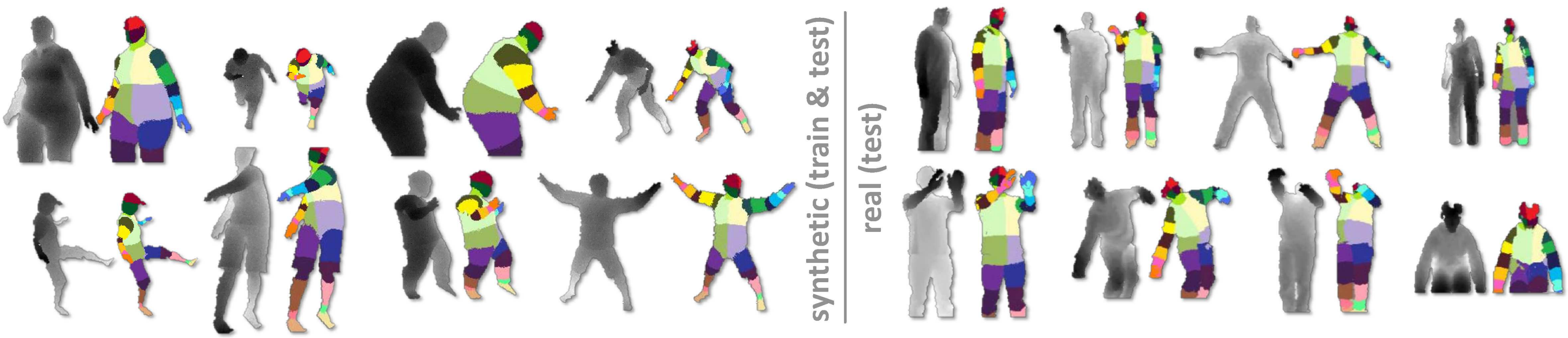
Z Not sure

‘Underfitting’



Posterior *very* certain despite blatant misfit. Prior ruled out truth.

Microsoft Kinect (Shotton et al., 2011)



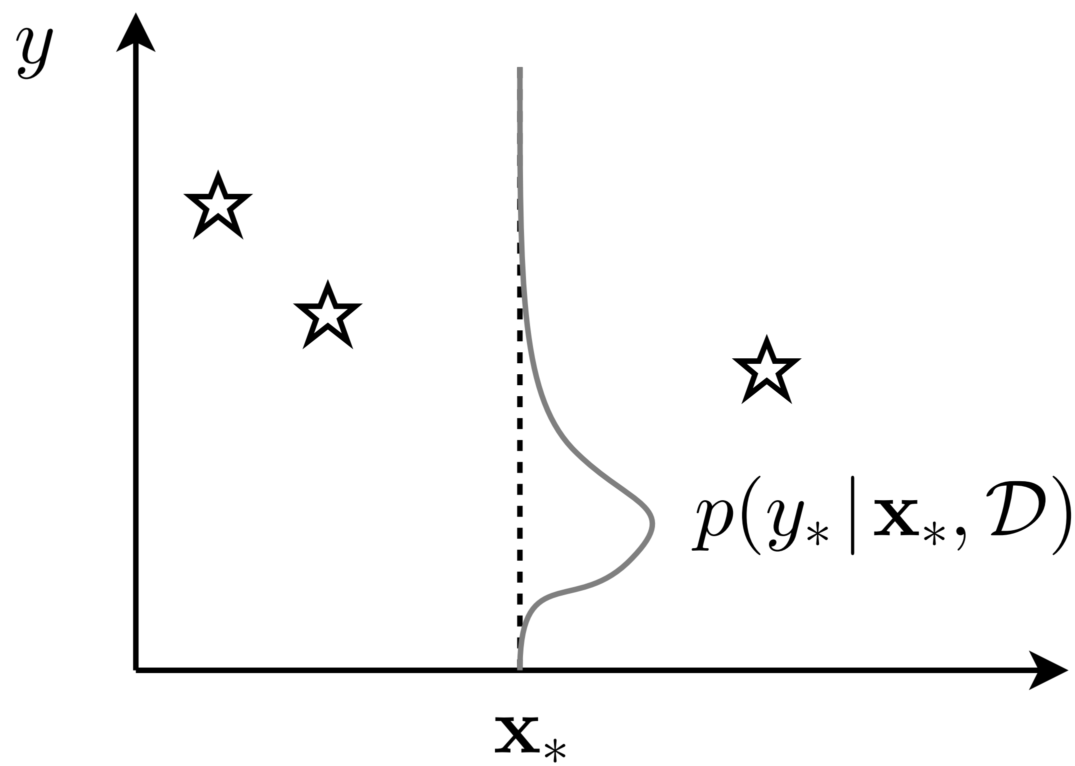
Eyeball modelling assumptions

Generate training data

Random forest applied to fantasies

The need for integrals

$$
\begin{aligned}
p(y_* \,|\, \mathbf{x}_*, \mathcal{D}) &= \int \mathrm{d}\theta \; p(y_*, \theta \,|\, \mathbf{x}_*, \mathcal{D}) \\
&= \int \mathrm{d}\theta \; p(y_* \,|\, \theta, \cancel{\mathcal{D}}) \; p(\theta \,|\, \cancel{\mathbf{x}_*}, \mathcal{D})
\end{aligned}
$$



A statistical problem

What is the average height of the people in this room?
Method: measure our heights, add them up and divide by N.

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

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

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

Surveying works for large and notionally infinite populations.

Simple Monte Carlo

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

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

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

Many other integrals appear throughout statistical 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 $\propto 1/S$:

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

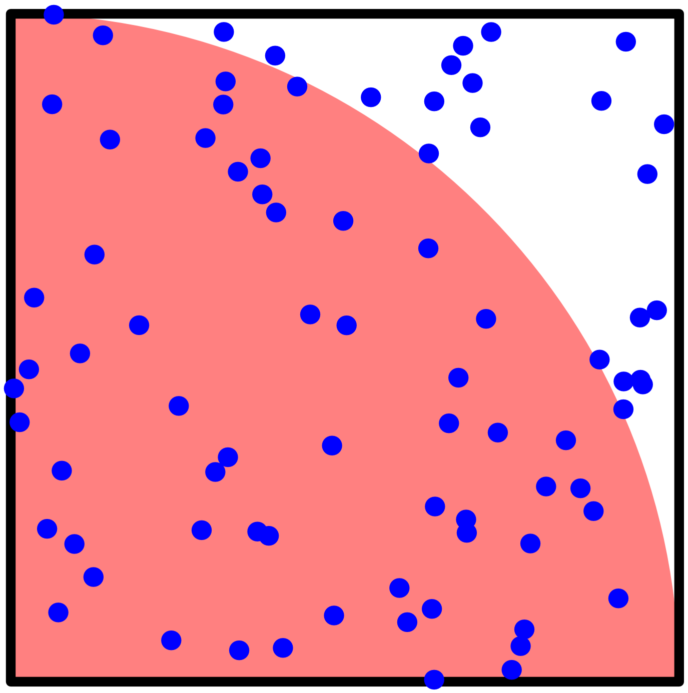
"Error bars" shrink like $\sqrt{S}$

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

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
```

Alternatives to Monte Carlo

There are other methods of numerical integration!

Example: (nice) 1D integrals are easy:

```
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 `tolerance=0`, but Octave works.)

In higher dimensions sometimes determinstic approximations work:
Variational Bayes, Laplace, . . . (covered later)

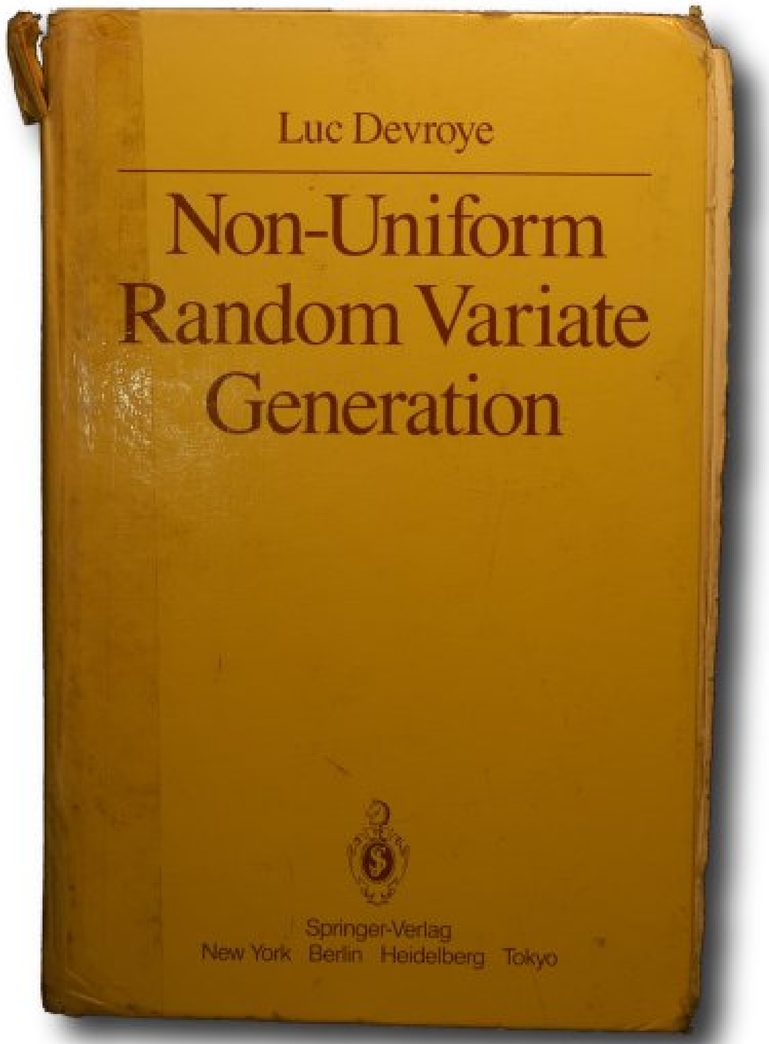
Reminder

Want to sample to approximate expectations:

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

How do we get the samples?

Sampling simple distributions



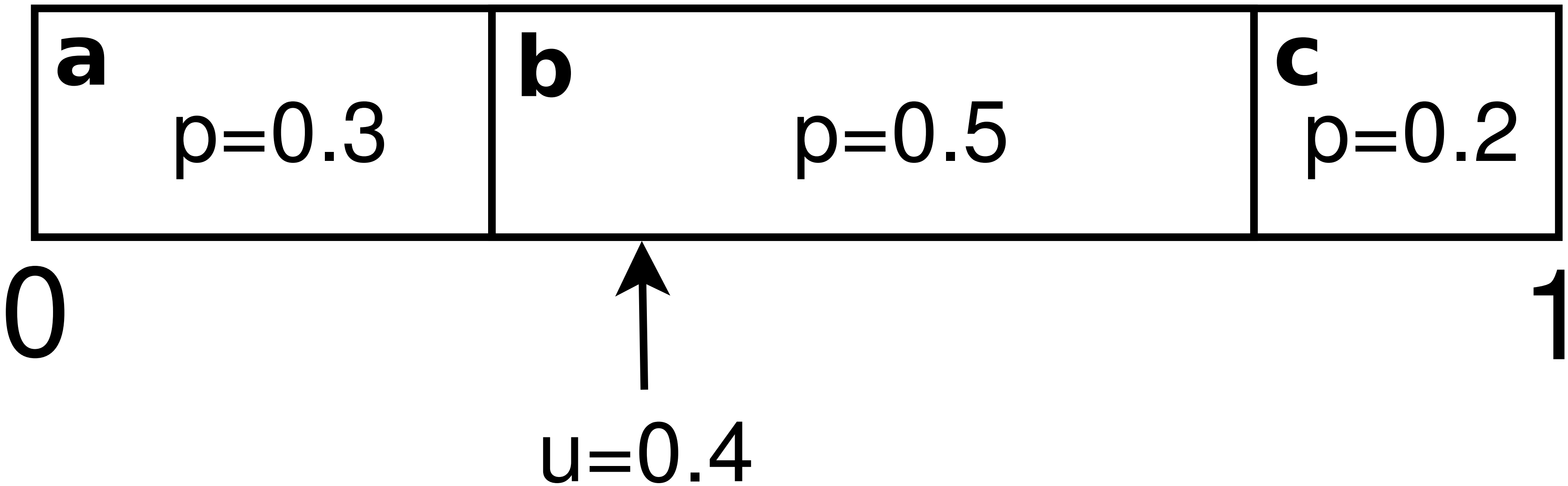
Use library routines for univariate distributions

(and some other special cases)

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

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

Sampling discrete values



$$u \sim \mathrm{Uniform}[0, 1]$$

$$u = 0.4 \quad \Rightarrow \quad x = \mathsf{b}$$

There are more efficient ways for large numbers of values and samples. See Devroye book.

Sampling from densities

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

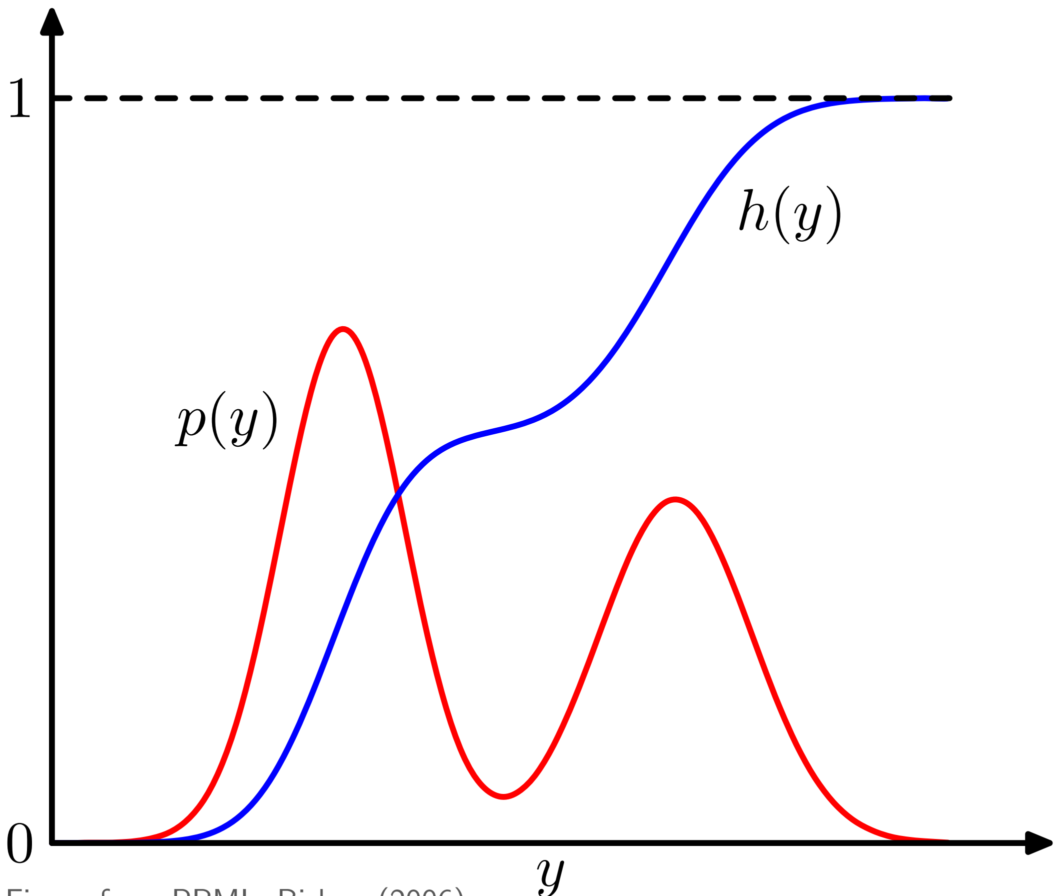


Figure from PRML, Bishop (2006)

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

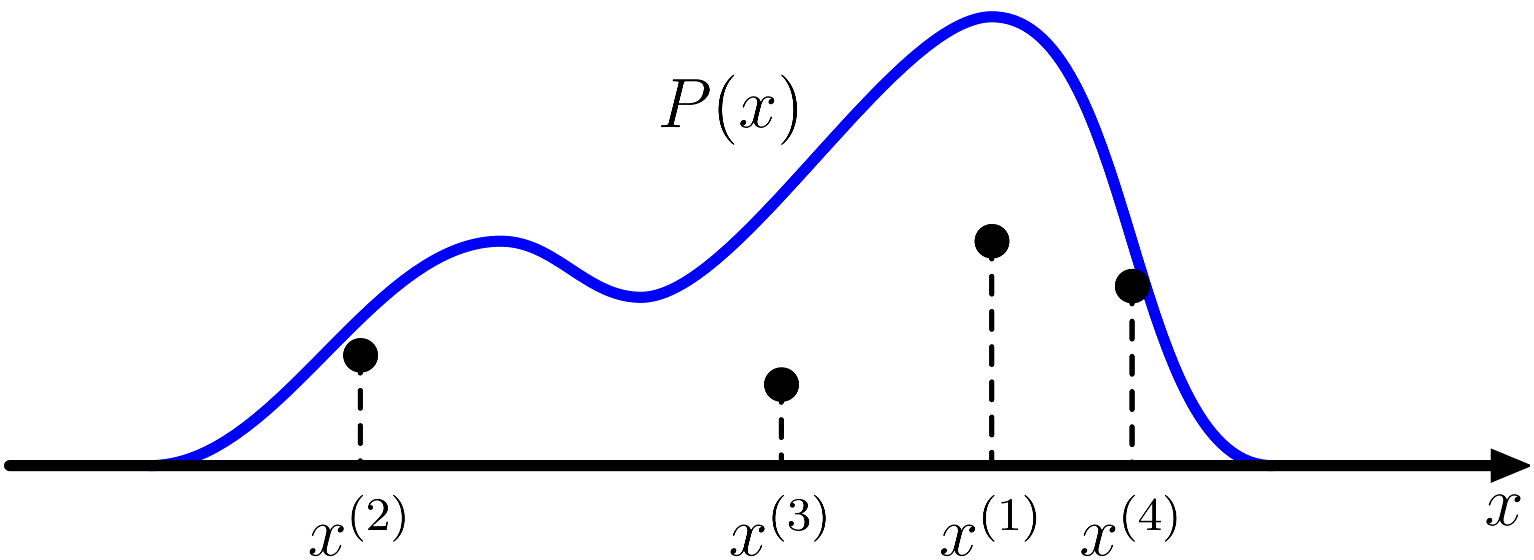
$u \sim$ Uniform[0,1]

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

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

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 $q(x)$:

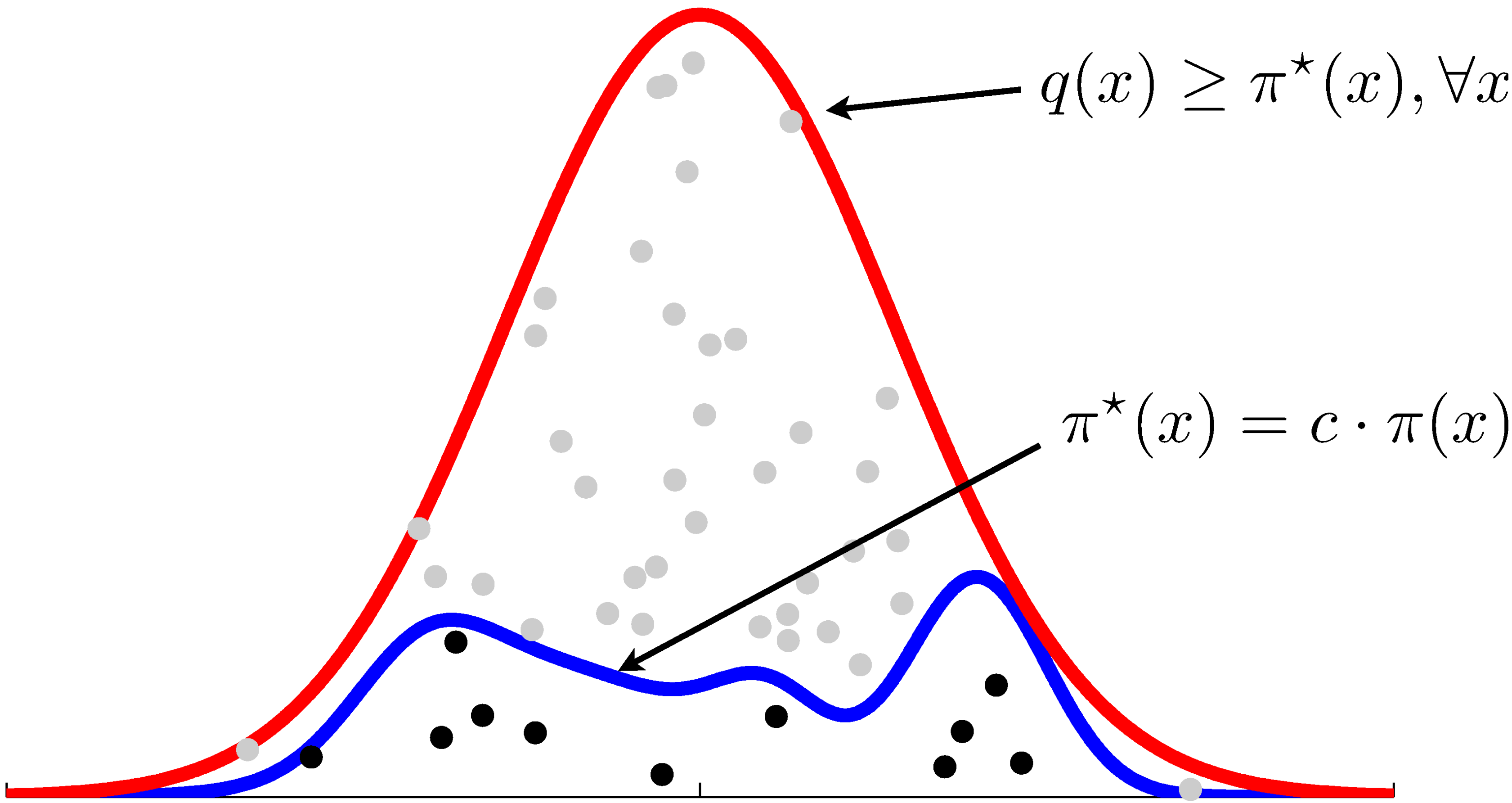


Figure credit: Ryan P. Adams

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)

If only know $P(x) = P^*(x)/\mathcal{Z}_P$ up to constant:

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

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

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

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

Summary so far

- **Monte Carlo**
  approximate expectations with a sample average
- **Rejection sampling**
  draw samples from complex distributions
- **Importance sampling**
  apply Monte Carlo to ‘any’ sum/integral

Next: High dimensional problems: MCMC

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:

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

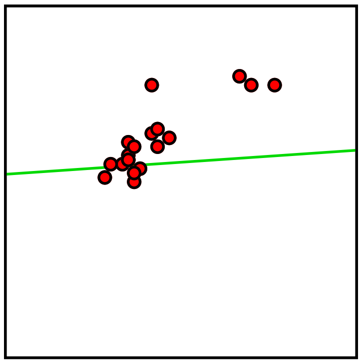
Infinite / undefined variance if $\sigma \leq 1/\sqrt{2}$

Reminder

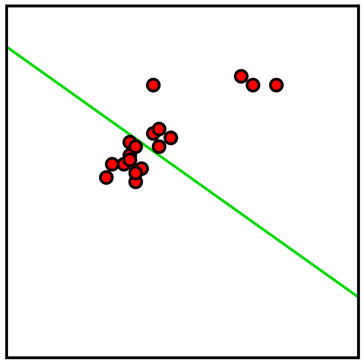
Need to sample large, non-standard distributions:

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

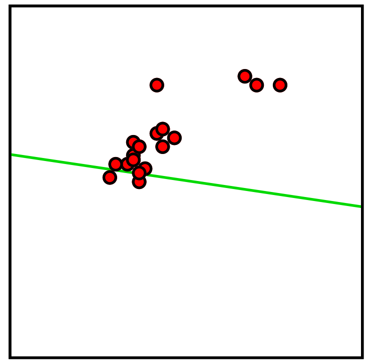
Importance sampling weights



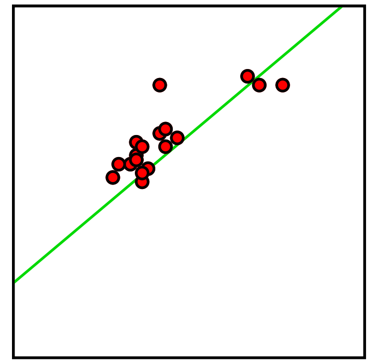
$w =$ 0.00548



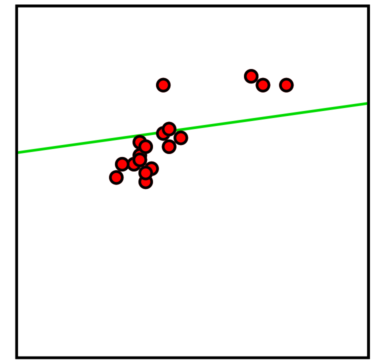
$w =$ 1.59e-08



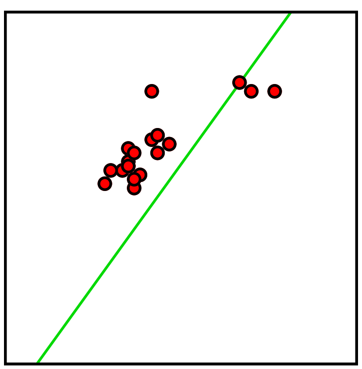
$w =$ 9.65e-06



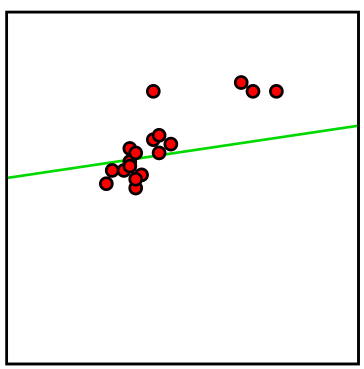
$w =$ 0.371



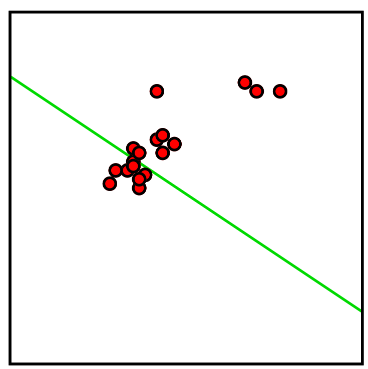
$w =$ 0.103



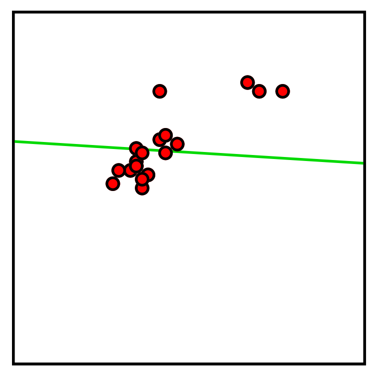
$w =$ 1.01e-08



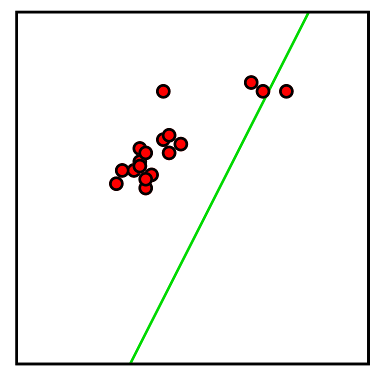
$w =$ 0.111



$w =$ 1.92e-09

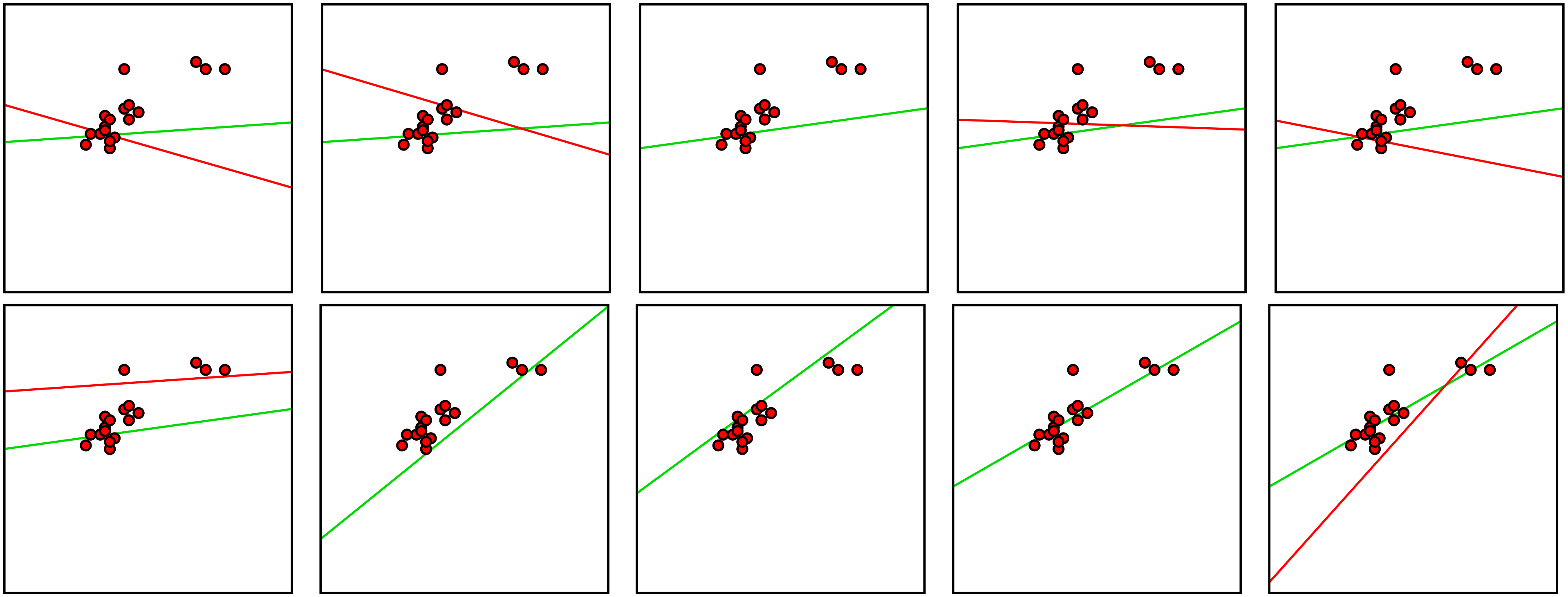


$w =$ 0.0126

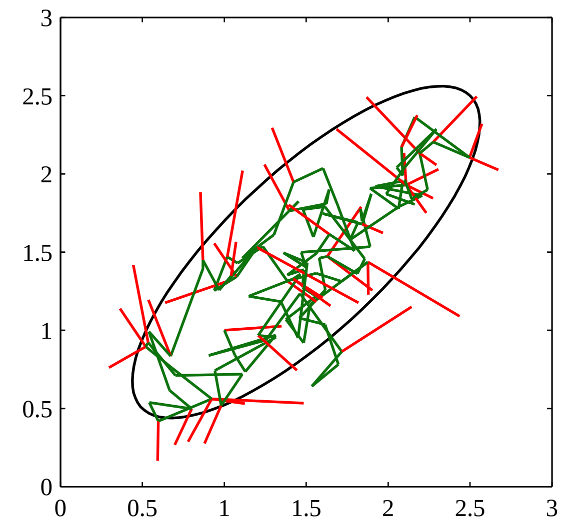


$w =$ 1.1e-51

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 $Q(\theta';\theta) = Q(\theta;\theta')$

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

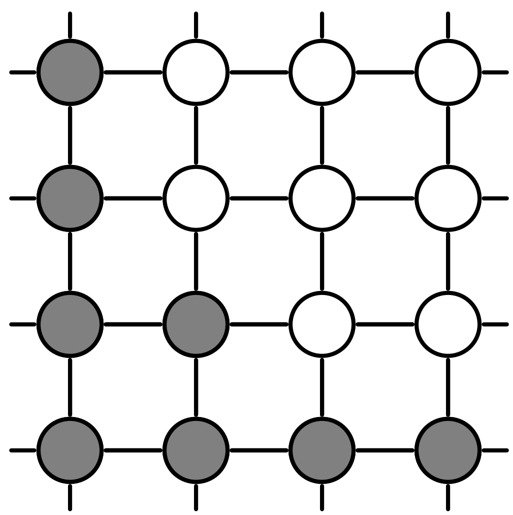
(Received March 6, 1953)

THE purpose of this paper is to describe a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. Classical statistics is assumed,

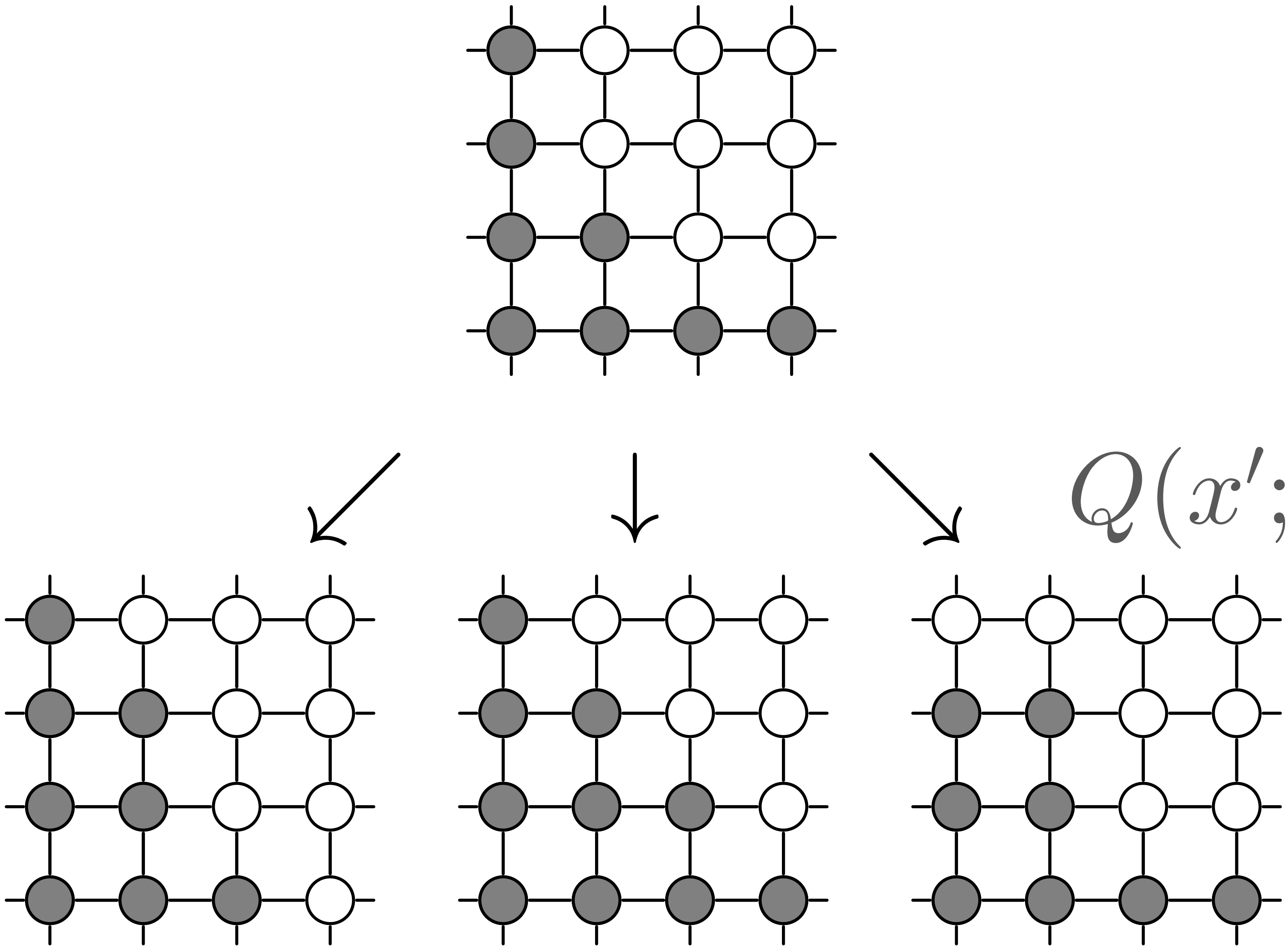
>20,000 citations

Target distribution

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

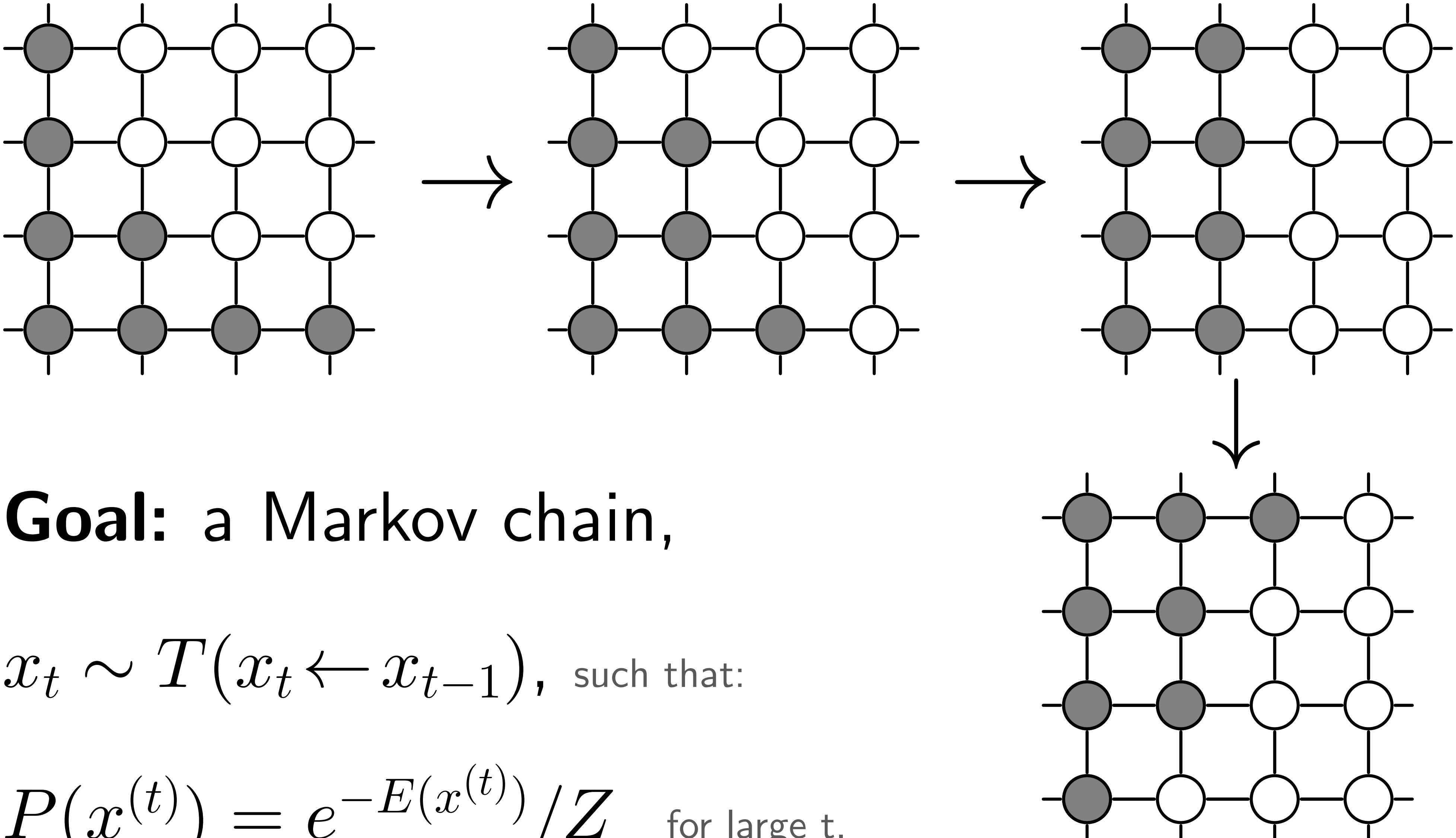
e.g. $\mathbf{x} =$ 

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,

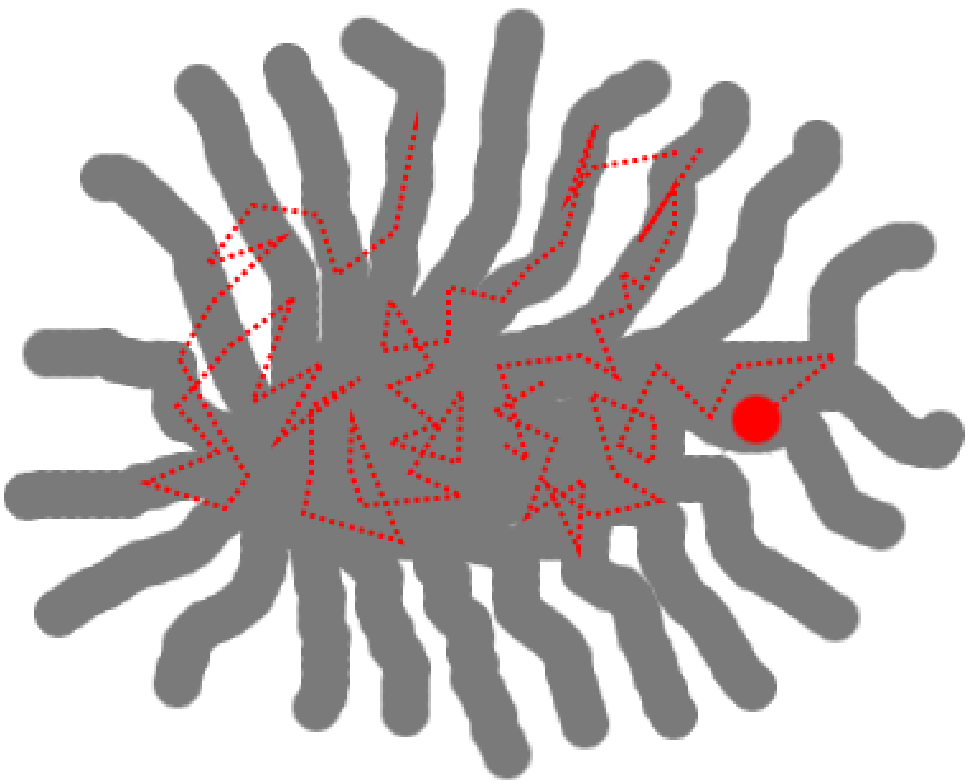
$x^{(t)}$ is also a sample from P.

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

Ergodicity

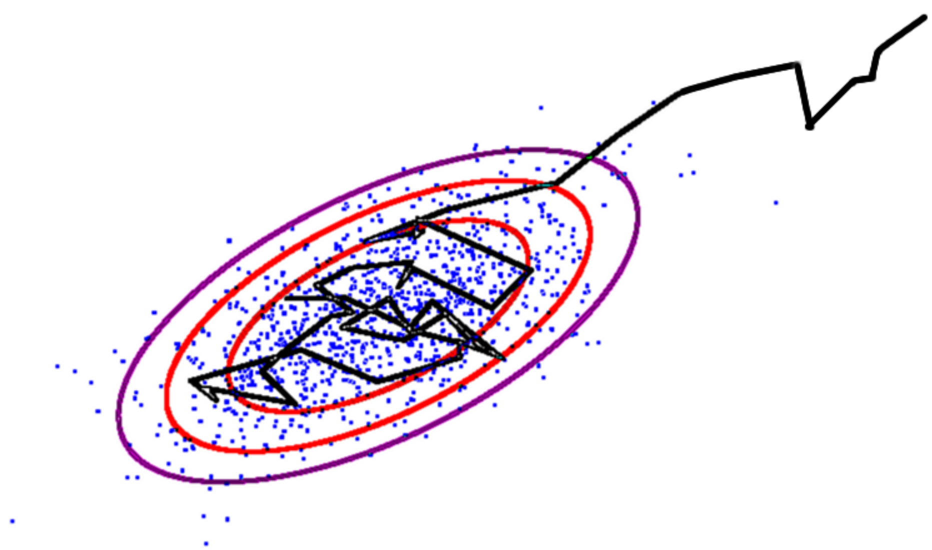
Unique invariant distribution

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

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

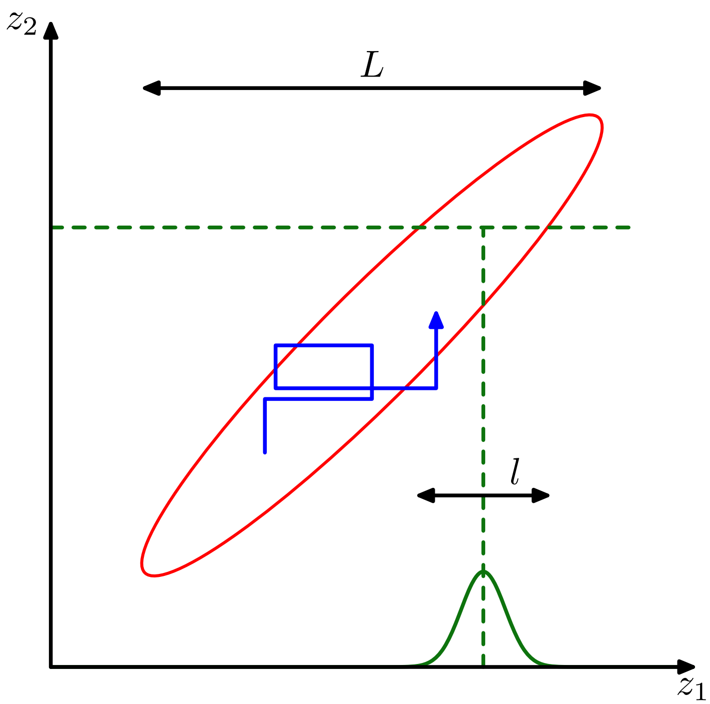
T must leave target invariant

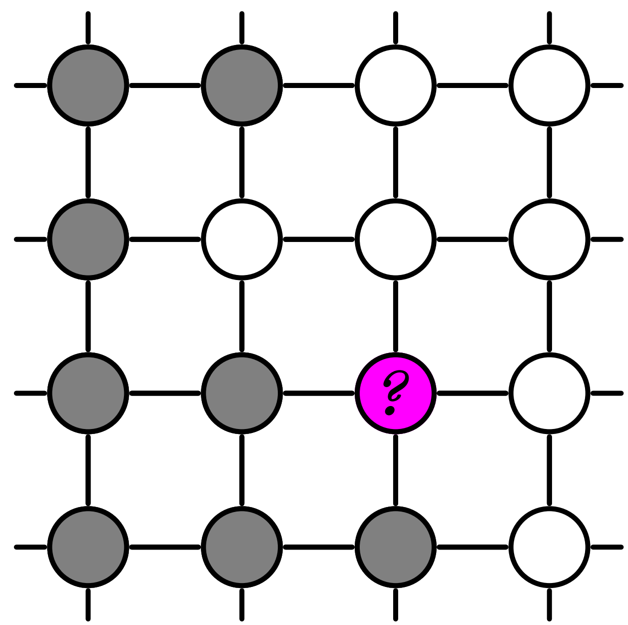
T must be able to get everywhere in K steps

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*

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

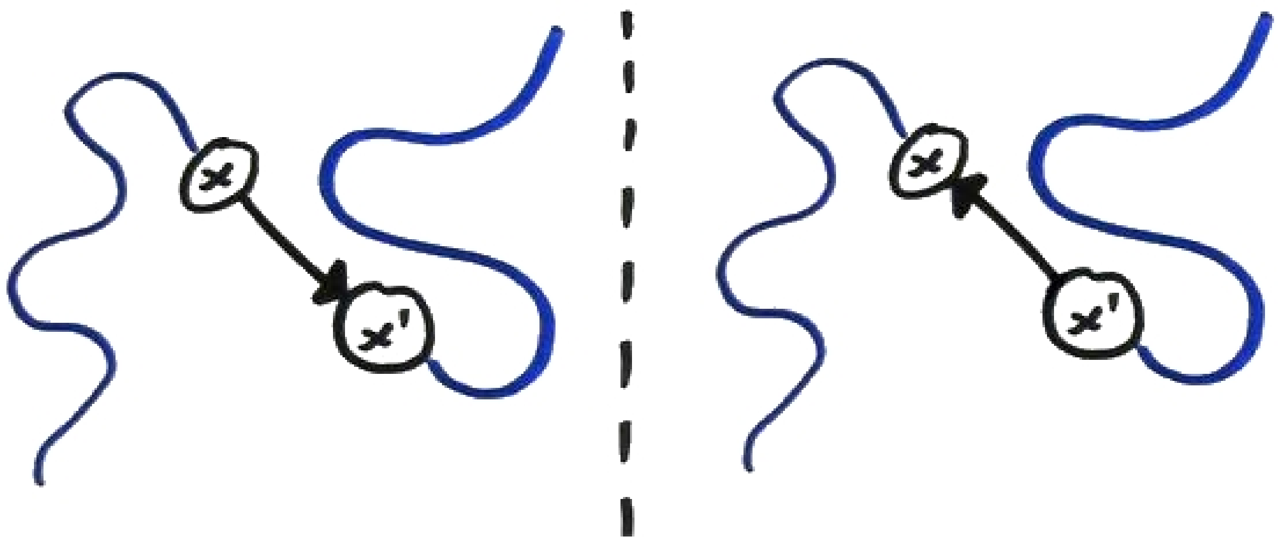
A necessary condition: there exists R such that:

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

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

Balance condition

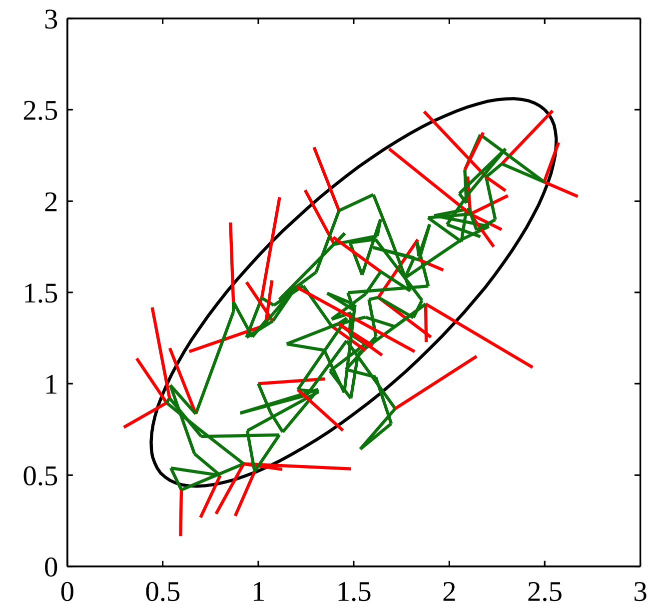
$$T(x' \leftarrow x)\, P(x) \;=\; R(x \leftarrow x')\, P(x')$$



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



PRML, Bishop (2006)

Arbitrary proposals $\sim Q$:

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

Satisfies detailed balance by rejecting moves:

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

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

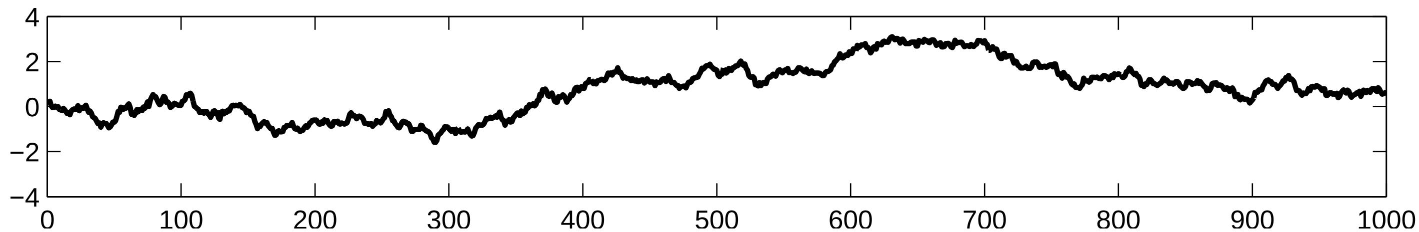
```
function samples = dumb_metropolis(init, log_ptilde, iters, sigma)

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

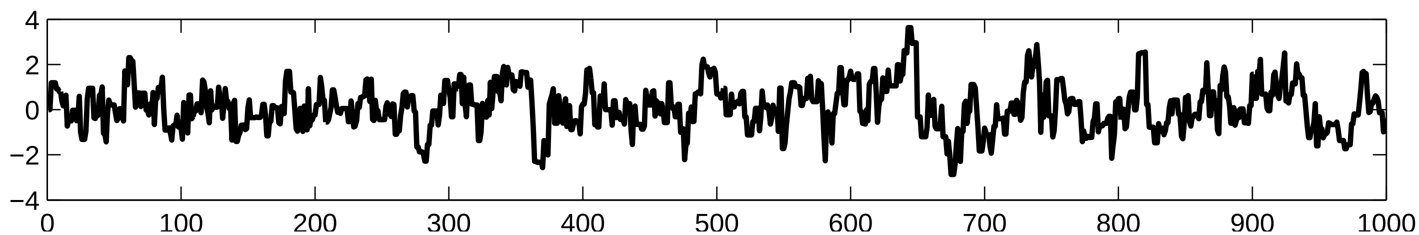
Explore $\mathcal{N}(0,1)$ with different step sizes σ

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

`sigma(0.1)`
99.8% accepts

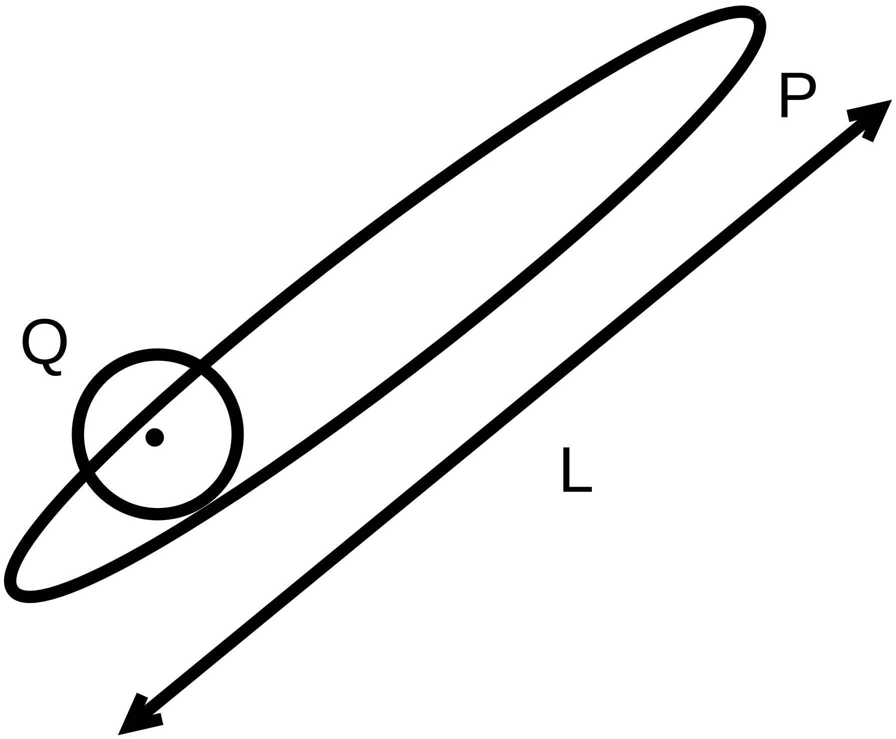


`sigma(1)`
68.4% accepts



`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:

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

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).