Approximation Methods for Posteriors and Marginal Likelihoods

- Laplace approximation
- Bayesian Information Criterion (BIC)
- Variational approximations
- Expectation Propagation (EP)
- Markov chain Monte Carlo methods (MCMC)
- Exact Sampling
- ...

Answers and expectations

For a function $f(x)$ and distribution $P(x)$, the expectation of f with respect to P is

$$E_{P(x)}\left[f(x)\right] = \sum_x f(x)P(x)$$

The expectation is the average of f, when x is drawn from the probability distribution P

The Monte Carlo principle

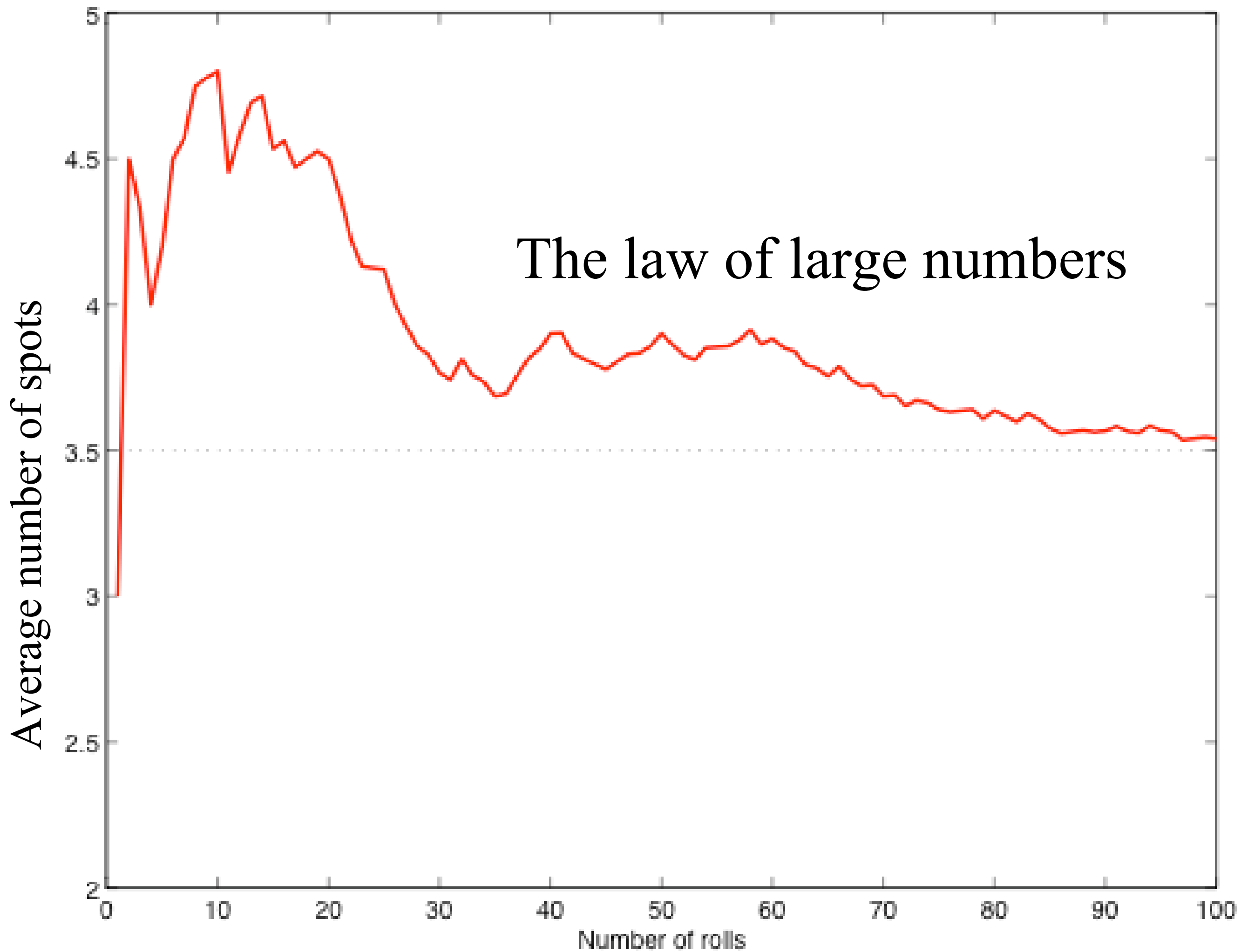
The expectation of f with respect to P can be approximated by

$$E_{P(x)}\left[f(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)$$

where the x_i are sampled from $P(x)$

Example 1: the average # of spots on a die roll

The Monte Carlo principle



Number of rolls

More formally…

$$\mu = E_{P(x)}\left[f(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) = \mu_{MC}$$

μ_{MC} is consistent, $(\mu_{MC} - \mu) \to 0$ a.s. as $n \to \infty$
μ_{MC} is unbiased, with $E[\mu_{MC}] = \mu$
μ_{MC} is asymptotically normal, with

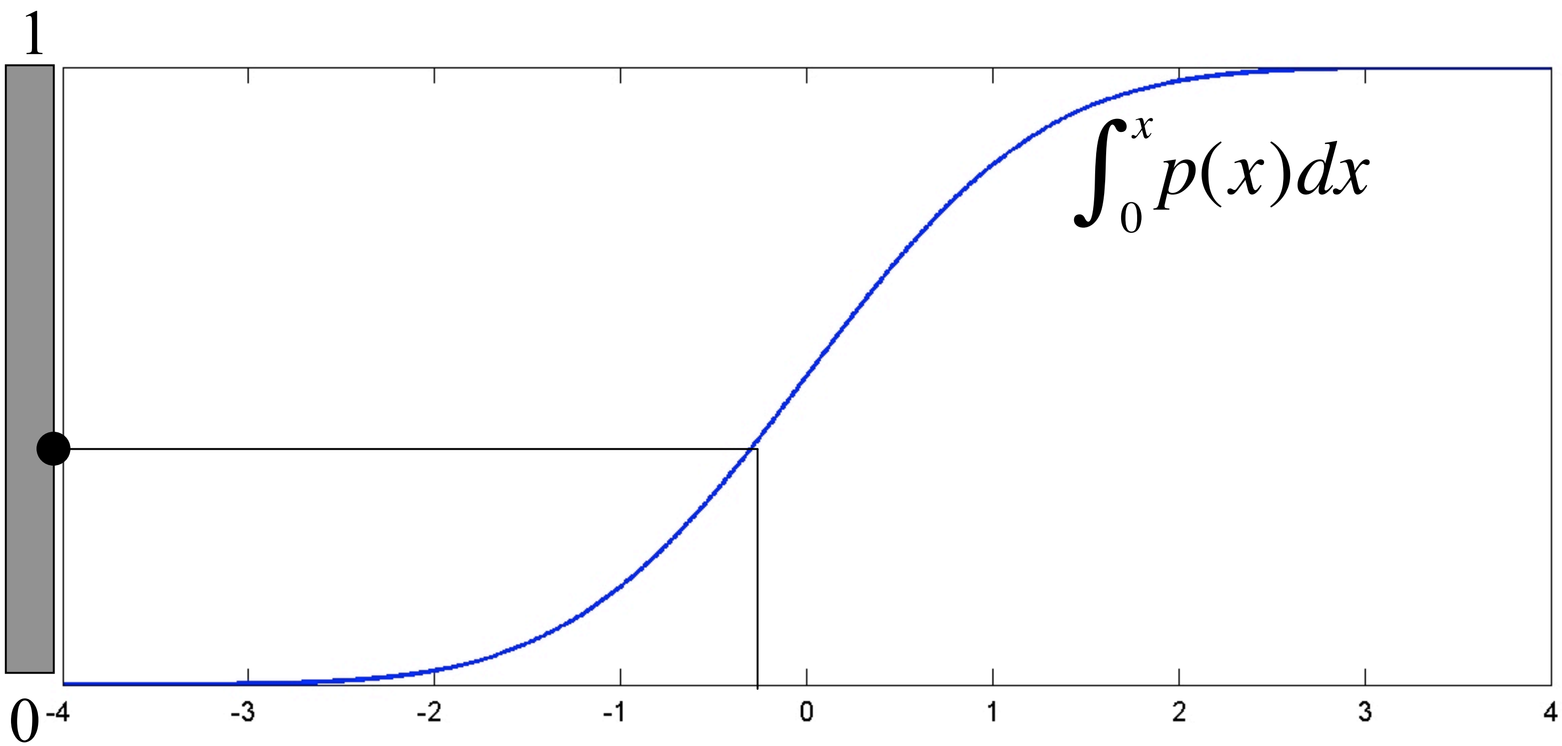
$$\sqrt{m}(\mu_{MC} - \mu) \to N(0, \sigma^2_{MC}) \quad \text{in distribution}$$

$$\sigma^2_{MC} = E_{P(x)}\left[(f(x) - E_{P(x)}[f(x)])^2\right]$$

When simple Monte Carlo fails

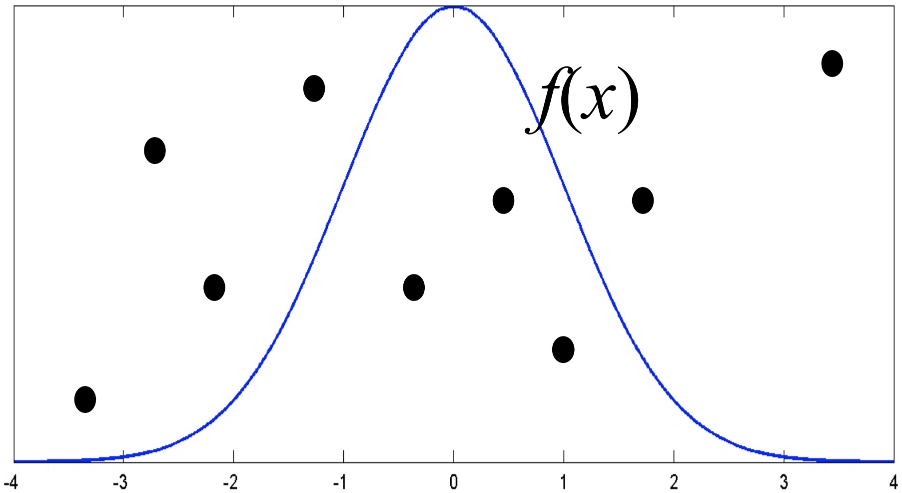
- Efficient algorithms for sampling only exist for a relatively small number of distributions

Inverse cumulative distribution



(requires CDF be invertible)

Rejection sampling



Rejection sampling

Want to sample from: $f(\theta) = g(\theta) / \int g(\theta) d\theta$

Rejection sampling uses an easy to sample from density $s(\theta)$

Requirement: $g(\theta) / s(\theta)$ is (upper) bounded by *A.*

Rejection sampling algorithm

For each sample

Do until one θ is accepted

1. sample a point θ from the known distribution $s(\theta)$;
2. sample *y* from the uniform distribution on [0, 1];
3. if $A\, y \leq g(\theta) / s(\theta)$ then break and accept θ;

When simple Monte Carlo fails

- Efficient algorithms for sampling only exist for a relatively small number of distributions
- Sampling from distributions over large discrete state spaces is computationally expensive
  - mixture model with *n* observations and *k* components, HMM with *n* observations and *k* states, k^n possibilities
- Sometimes we want to sample from distributions for which we only know the probability of each state up to a multiplicative constant

Why Bayesian inference is hard

$$P(h \mid d) = \frac{P(d \mid h)P(h)}{\boxed{\sum_{h' \in H} P(d \mid h')P(h')}}$$

Evaluating the posterior probability of a hypothesis requires summing over all hypotheses

(statistical physics: computing partition function)

Modern Monte Carlo methods

- Sampling schemes for distributions with large state spaces known up to a multiplicative constant
- Two example approaches:
  - importance sampling
  - Markov chain Monte Carlo

Importance sampling

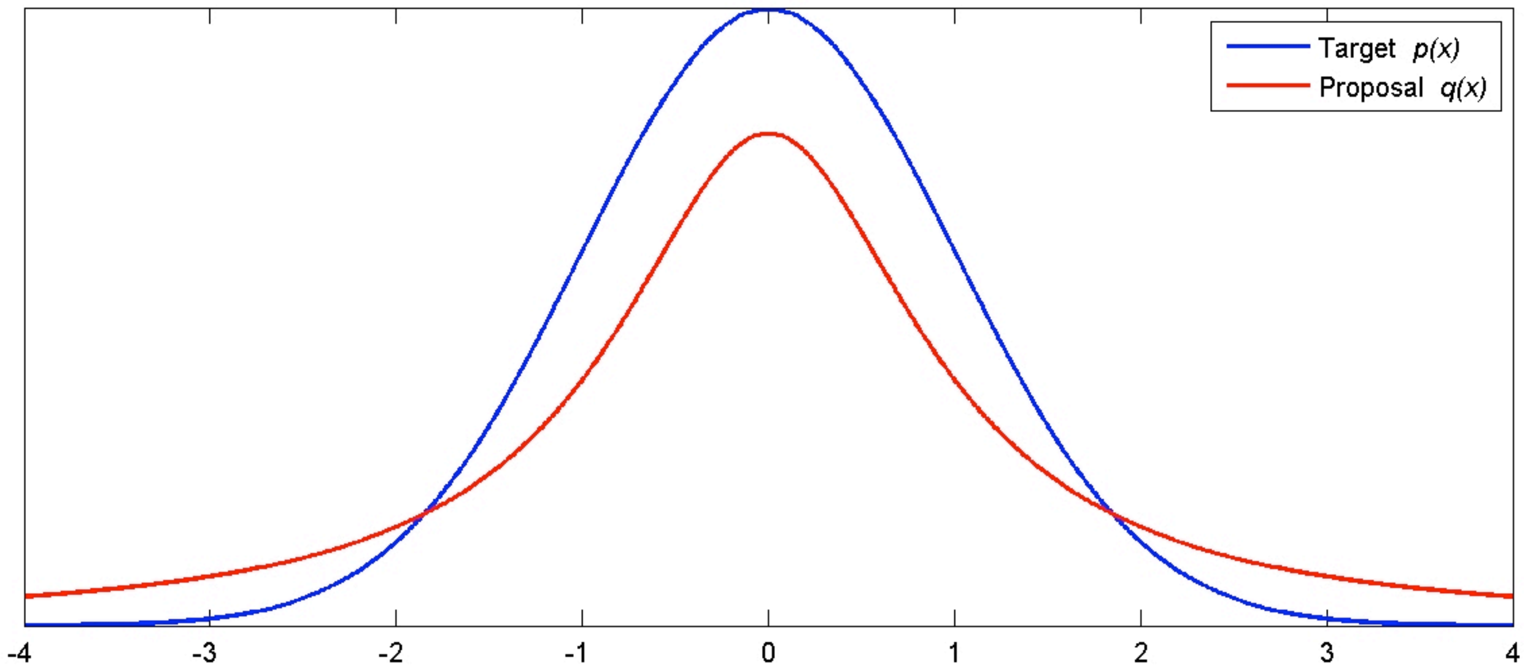
Basic idea: generate from the wrong distribution, assign weights to samples to correct for this

$$E_{p(x)}\left[f(x)\right] = \int f(x)\,p(x)\,dx$$

$$= \int f(x)\frac{p(x)}{q(x)}q(x)\,dx$$

$$\approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{p(x_i)}{q(x_i)} \quad \text{for } x_i \sim q(x)$$

Importance sampling



works when sampling from proposal is easy, target is hard

An alternative scheme…

$$E_{p(x)}\left[f(x)\right] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i)\frac{p(x_i)}{q(x_i)} \quad \text{for } x_i \sim q(x)$$

$$E_{p(x)}\left[f(x)\right] \approx \frac{\sum_{i=1}^{n} f(x_i)\frac{p(x_i)}{q(x_i)}}{\sum_{i=1}^{n} \frac{p(x_i)}{q(x_i)}} \quad \text{for } x_i \sim q(x)$$

works when $p(x)$ is known up to a multiplicative constant

More formally…

μ_{IS} is consistent, $(\mu_{IS} - \mu) \to 0$ a.s. as $n \to \infty$
μ_{IS} is asymptotically normal, with

$$\sigma_{IS}^2 = E_{p(x)}\left[(f(x) - E_{p(x)}[f(x)])^2 \frac{p(x)}{q(x)}\right]$$

μ_{IS} is biased, with

$$\mu_{IS} - \mu = \frac{1}{n}\left(E_{p(x)}[f(x)]E_{p(x)}\left[\frac{p(x)}{q(x)}\right] - E_{p(x)}\left[f(x)\frac{p(x)}{q(x)}\right]\right)$$

Optimal importance sampling

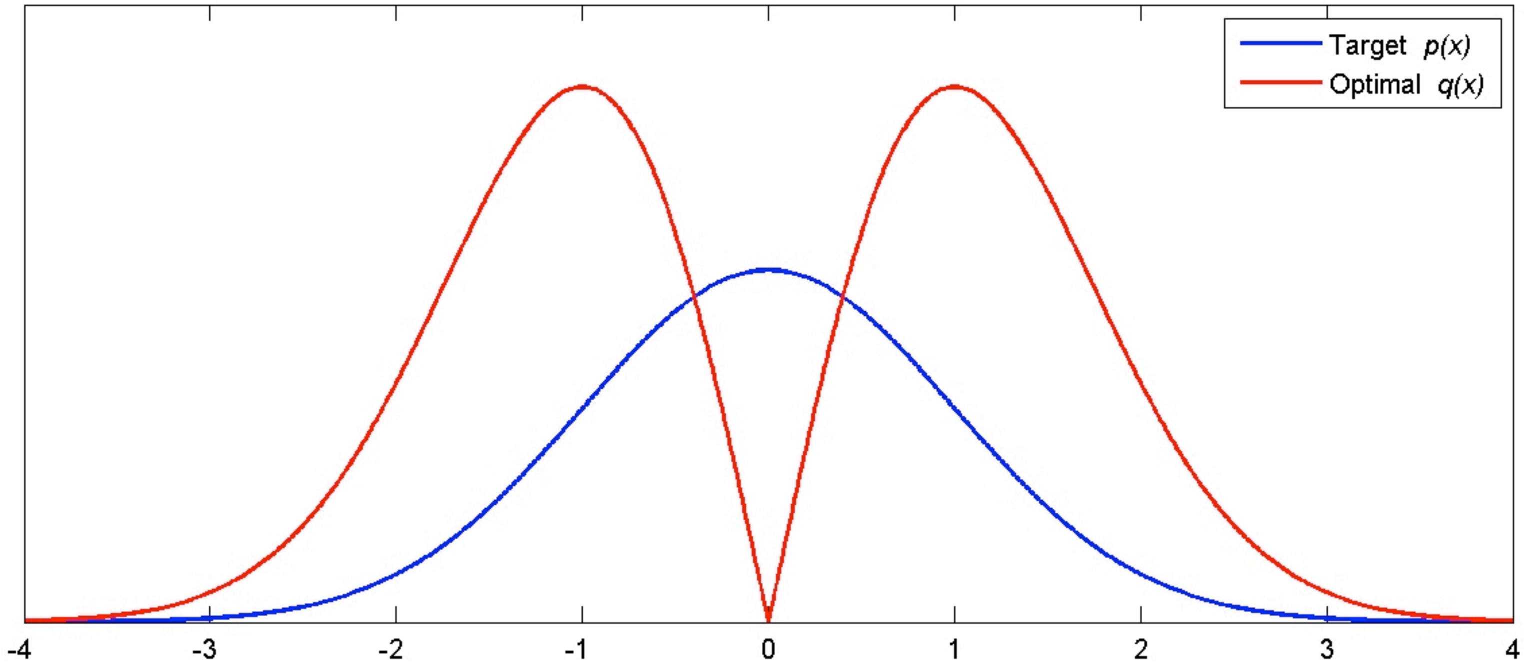
- Asymptotic variance is

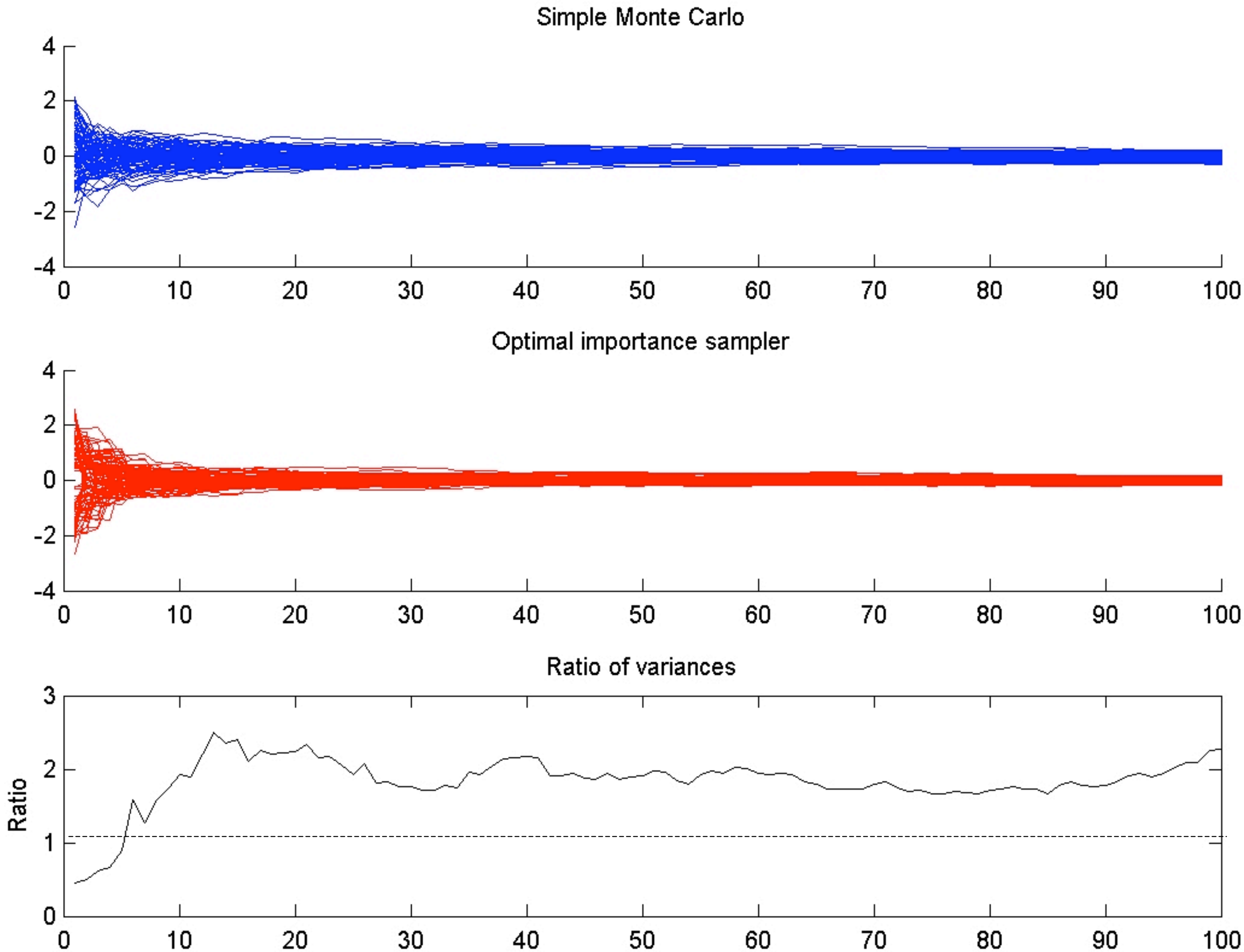
$$\sigma_{IS}^2 = E_{p(x)}\left[(f(x) - E_{p(x)}[f(x)])^2 \frac{p(x)}{q(x)}\right]$$

- This is minimized by

$$q(x) \propto \left| f(x) - E_{p(x)}[f(x)] \right| p(x)$$

Optimal importance sampling





Simple Monte Carlo
4
2
0
-2
-4
0
10
20
30
40
50
60
70
80
90
100
Optimal importance sampler
4
2
0
-2
-4
0
10
20
30
40
50
60
70
80
90
100
Ratio of variances
3
2
1
0
Ratio
0
10
20
30
40
50
60
70
80
90
100

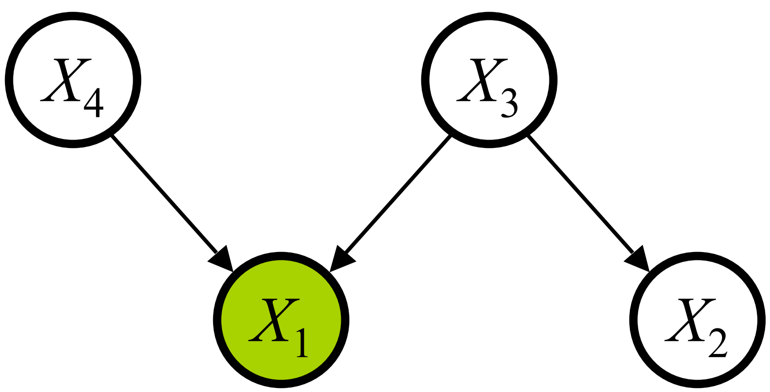
Likelihood weighting

- A particularly simple form of importance sampling for posterior distributions
- Use the prior as the proposal distribution
- Weights:

$$\frac{p(\theta \mid D)}{p(\theta)} = \frac{p(D \mid \theta)\, p(\theta)}{p(D)\, p(\theta)} = \frac{p(D \mid \theta)}{p(D)} \propto p(D \mid \theta)$$

Likelihood weighting

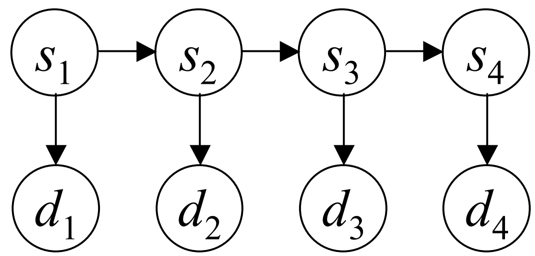
- Generate samples of all variables except observed variables
- Assign weights proportional to probability of observed data given values in sample



Importance sampling

- A general scheme for sampling from complex distributions that have simpler relatives
- Simple methods for sampling from posterior distributions in some cases (easy to sample from prior, prior and posterior are close)
- Can be more efficient than simple Monte Carlo
  - particularly for, e.g., tail probabilities
- Also provides a solution to the question of how we can update beliefs as data come in…

Particle filtering



We want to generate samples from $P(s_4|d_1, \ldots, d_4)$

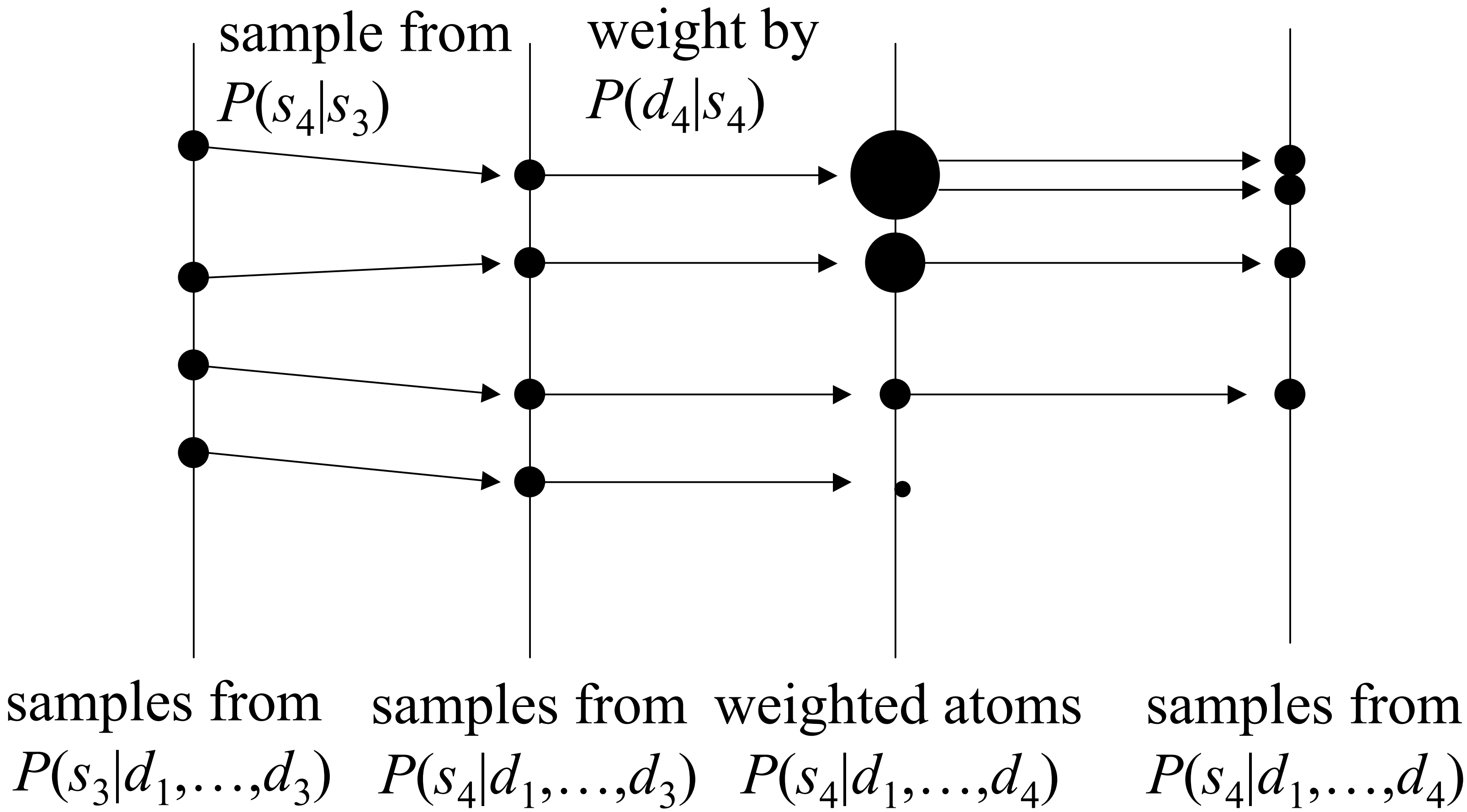
$$P(s_4 \mid d_1,\ldots,d_4) \propto P(d_4 \mid s_4)P(s_4 \mid d_1,\ldots,d_3)$$

$$= P(d_4 \mid s_4)\sum_{s_3} P(s_4 \mid s_3)P(s_3 \mid d_1,\ldots,d_3)$$

We can use likelihood weighting if we can sample from $P(s_4|s_3)$ and $P(s_3|d_1, \ldots, d_3)$

Particle filtering

$$P(s_4 \mid d_1,\ldots,d_4) \propto P(d_4 \mid s_4)\sum_{s_3} P(s_4 \mid s_3)P(s_3 \mid d_1,\ldots,d_3)$$



Tweaks and variations

- If we can enumerate values of s_4, can sample from

$$P(s_4 \mid d_1,\ldots,d_4) \propto P(d_4 \mid s_4) \sum_{i=1}^{n} P(s_4 \mid s_3^{(i)})$$

- No need to resample at every step, since we can accumulate weights over multiple observations
  - resampling reduces diversity in samples
  - only necessary when variance of weights is large
- Stratification and clever resampling schemes reduce variance (Fearnhead, 2001)

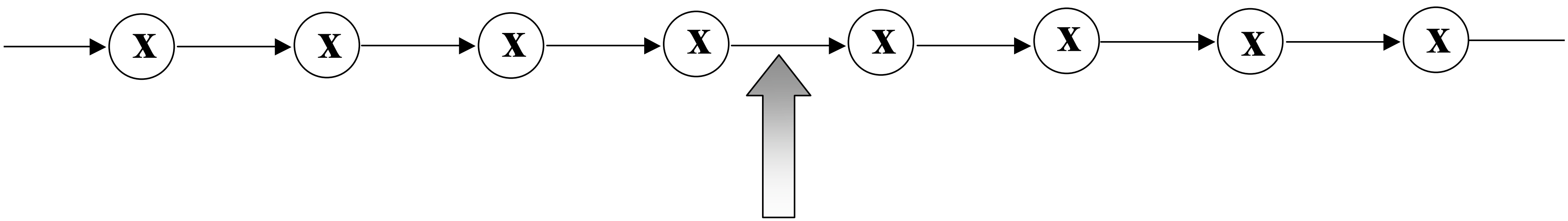
The promise of particle filters

- People need to be able to update probability distributions over large hypothesis spaces as more data become available
- Particle filters provide a way to do this with limited computing resources…
  - maintain a fixed finite number of samples
- Not just for dynamic models
  - can work with a fixed set of hypotheses, although this requires some further tricks for maintaining diversity

Markov chain Monte Carlo

- Basic idea: construct a *Markov chain* that will converge to the target distribution, and draw samples from that chain
- Just uses something proportional to the target distribution (good for Bayesian inference!)
- Can work in state spaces of arbitrary (including unbounded) size (good for nonparametric Bayes)

Markov chains



Transition matrix

$\mathbf{T} = P(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$

Variables $\mathbf{x}^{(t+1)}$ independent of all previous variables given immediate predecessor $\mathbf{x}^{(t)}$

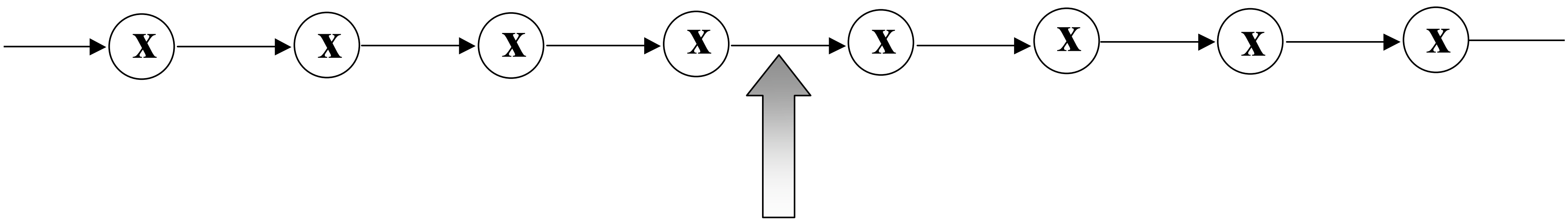
An example: card shuffling

- Each state $\mathbf{x}^{(t)}$ is a permutation of a deck of cards (there are 52! permutations)
- Transition matrix $\mathbf{T}$ indicates how likely one permutation will become another
- The transition probabilities are determined by the shuffling procedure
  - riffle shuffle
  - overhand
  - one card

Convergence of Markov chains

- Why do we shuffle cards?
- Convergence to a uniform distribution takes only 7 riffle shuffles…
- Other Markov chains will also converge to a *stationary distribution*, if certain simple conditions are satisfied (called “ergodicity”)
  - e.g. every state can be reached in some number of steps from every other state

Markov chain Monte Carlo



Transition matrix

$$\mathbf{T} = P(\boldsymbol{x}^{(t+1)} | \boldsymbol{x}^{(t)})$$

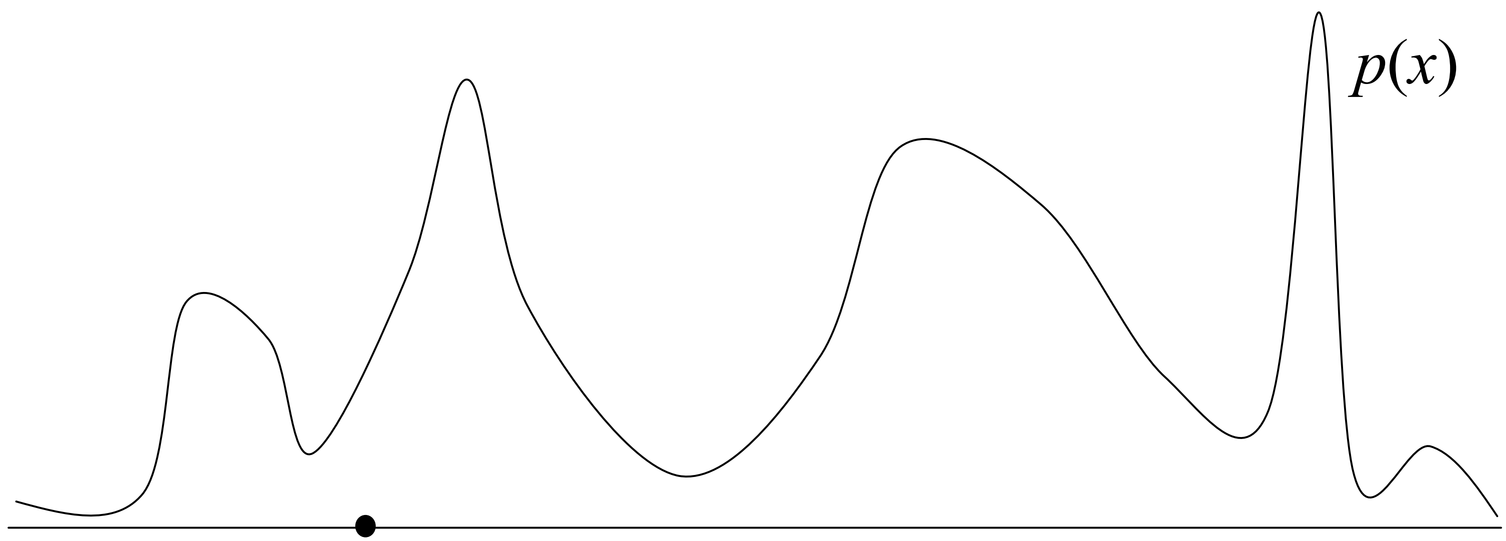
- States of chain are variables of interest
- Transition matrix chosen to give target distribution as stationary distribution

Metropolis-Hastings algorithm

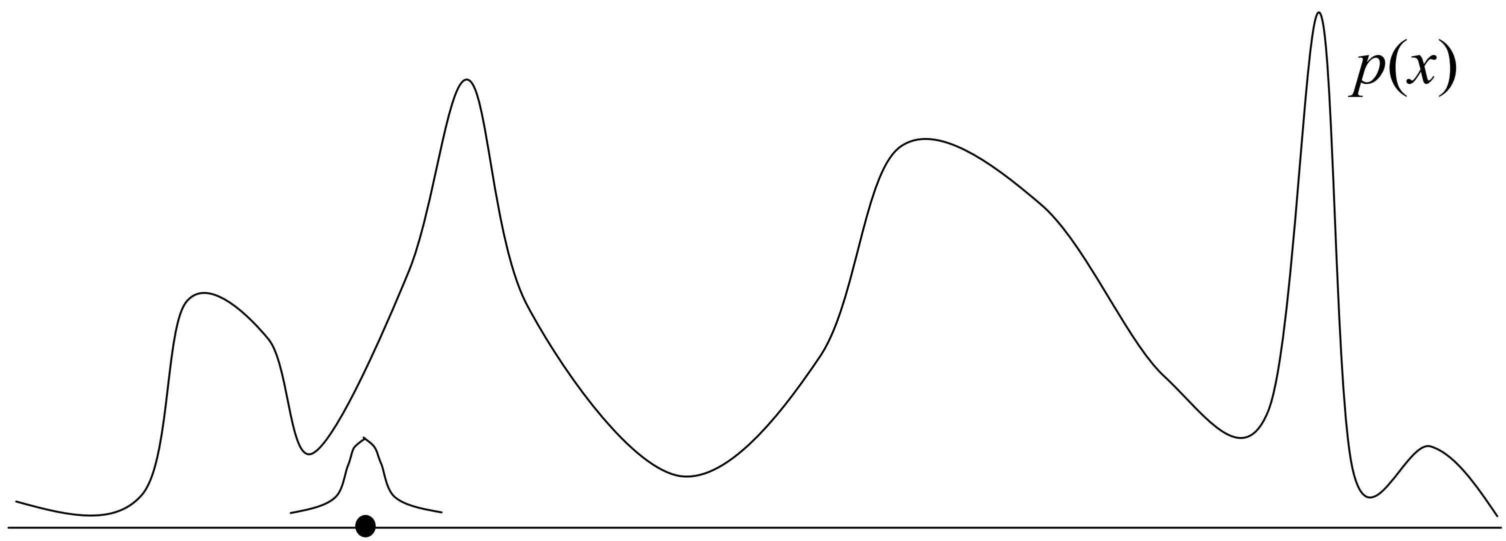
- Transitions have two parts:
  - proposal distribution: $Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$
  - acceptance: take proposals with probability

$$A(\mathbf{x}^{(t)},\mathbf{x}^{(t+1)}) = \min\left(1, \frac{P(\mathbf{x}^{(t+1)})\ Q(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})}{P(\mathbf{x}^{(t)})\ Q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})} \right)$$

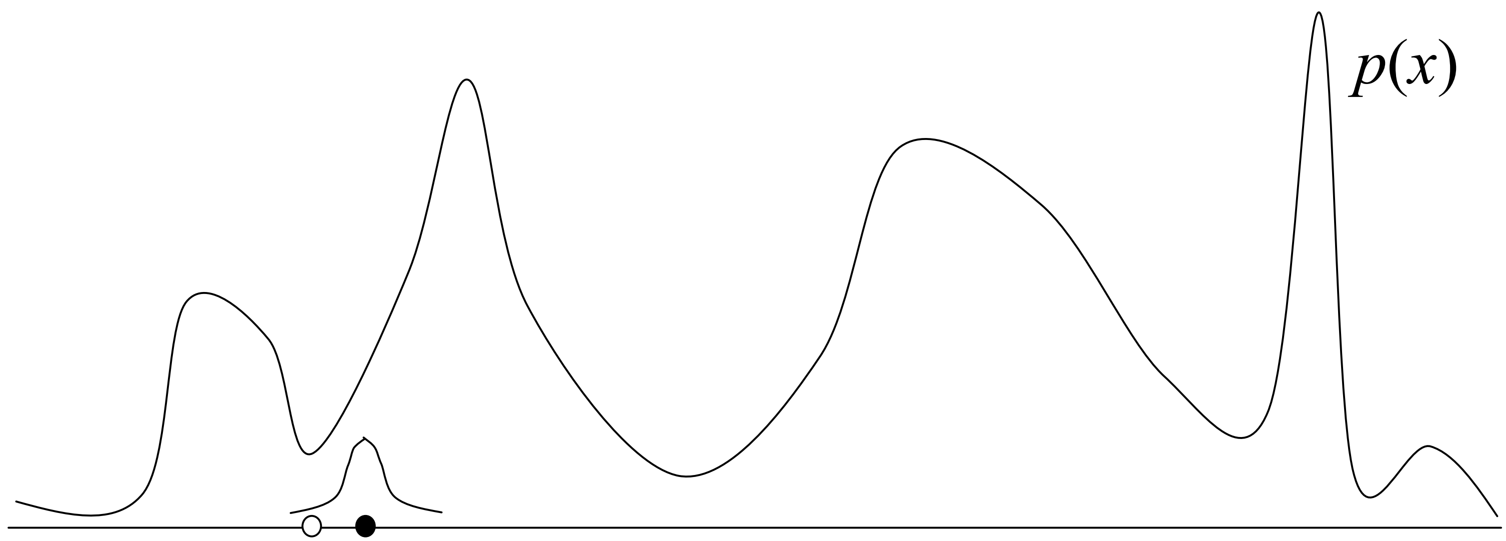
Metropolis-Hastings algorithm



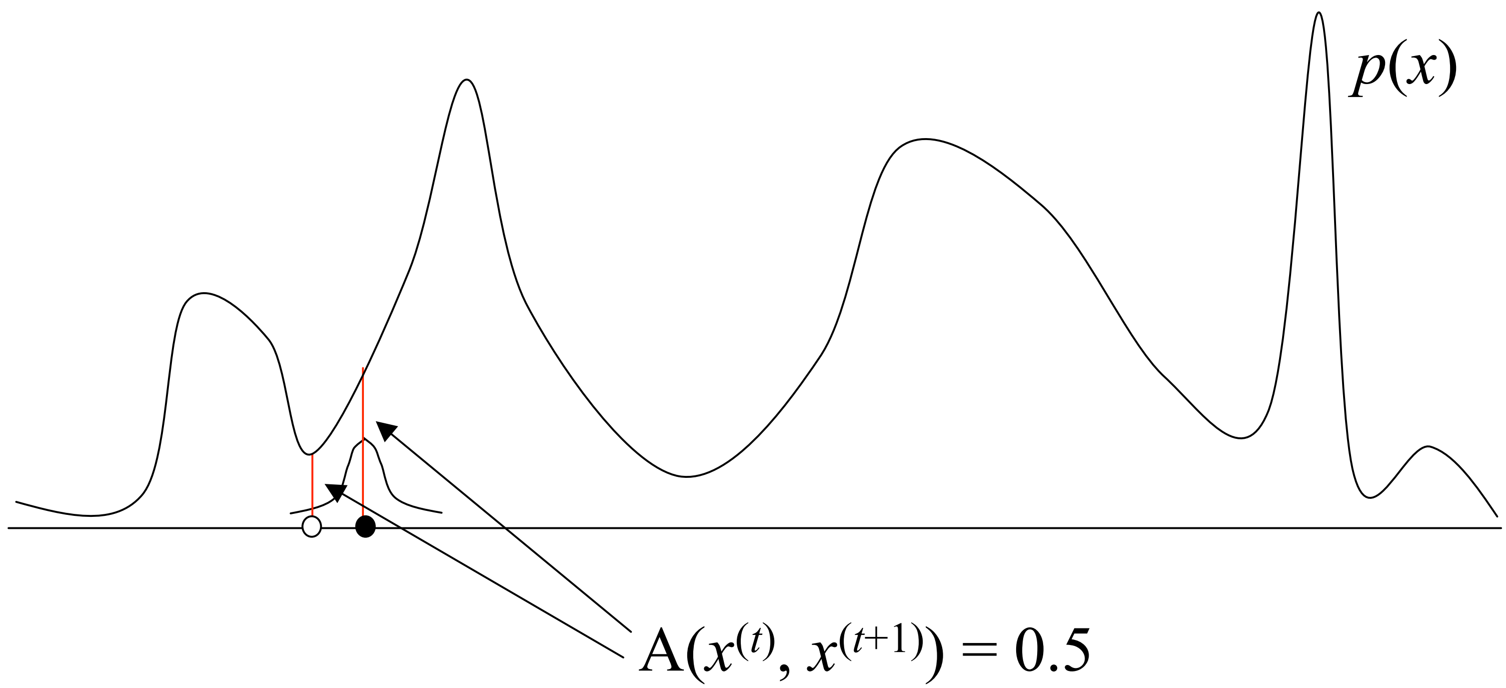
Metropolis-Hastings algorithm



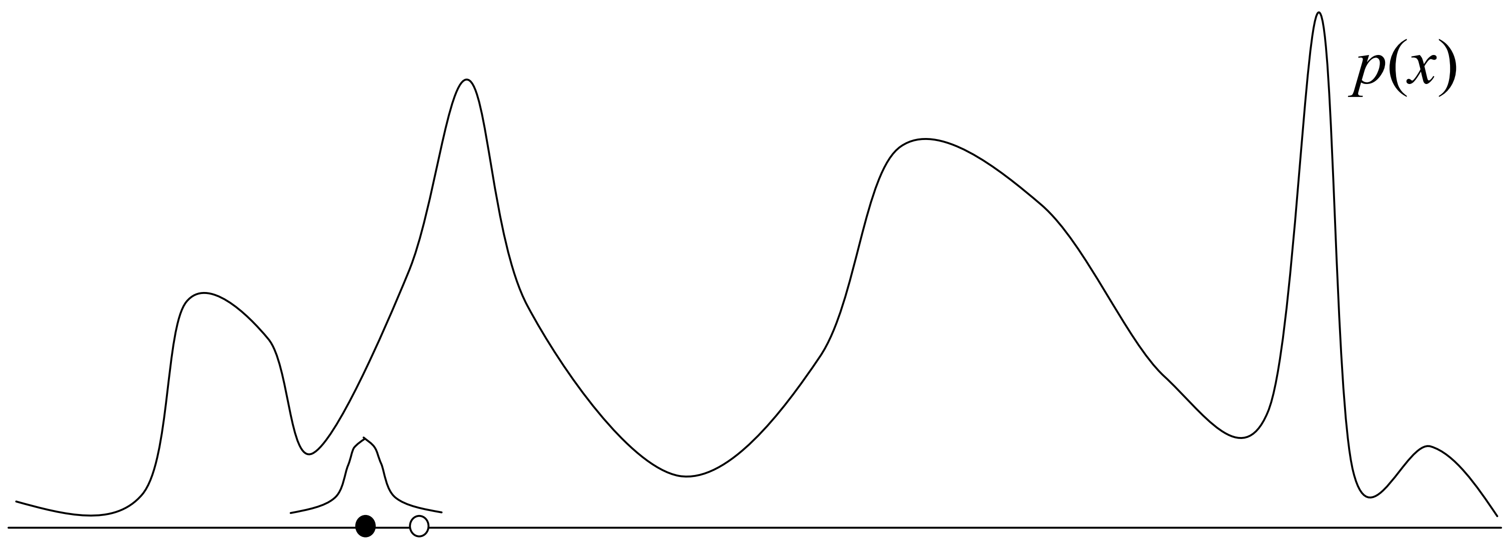
Metropolis-Hastings algorithm



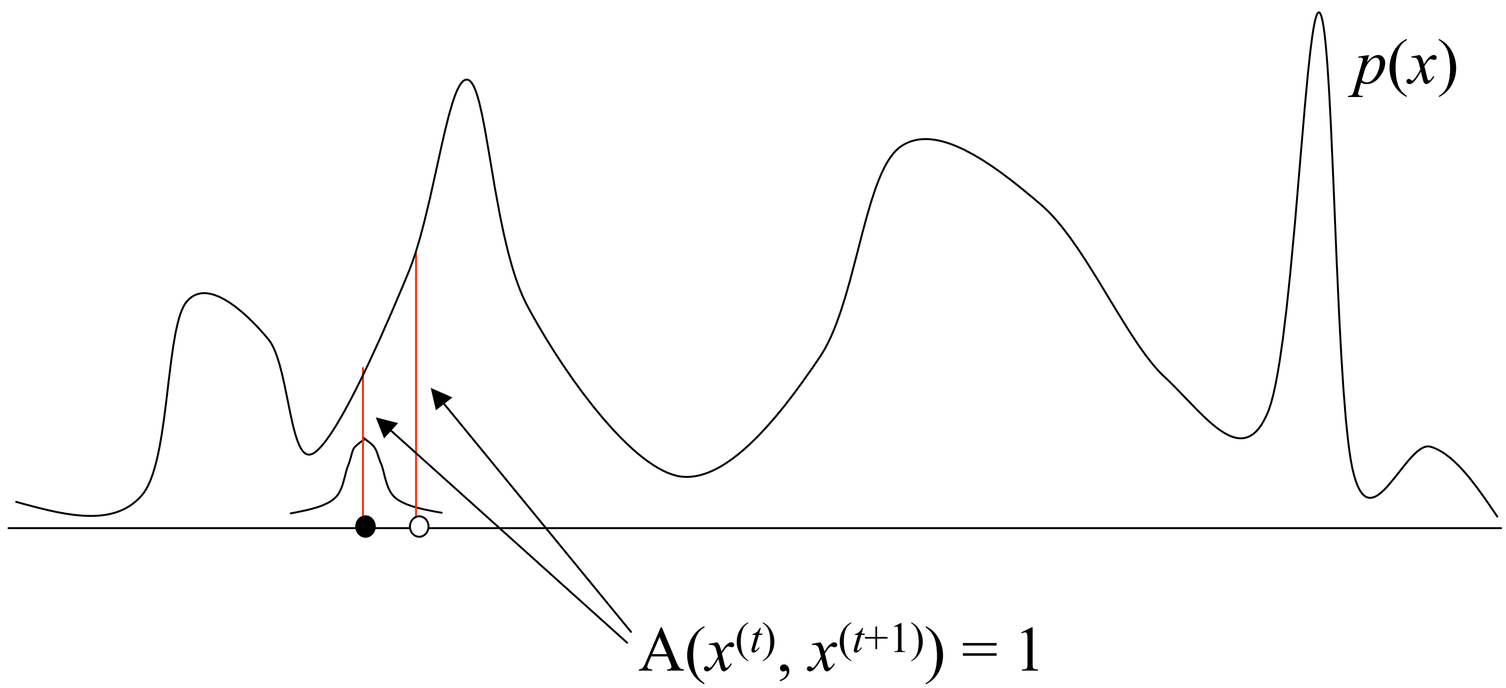
Metropolis-Hastings algorithm



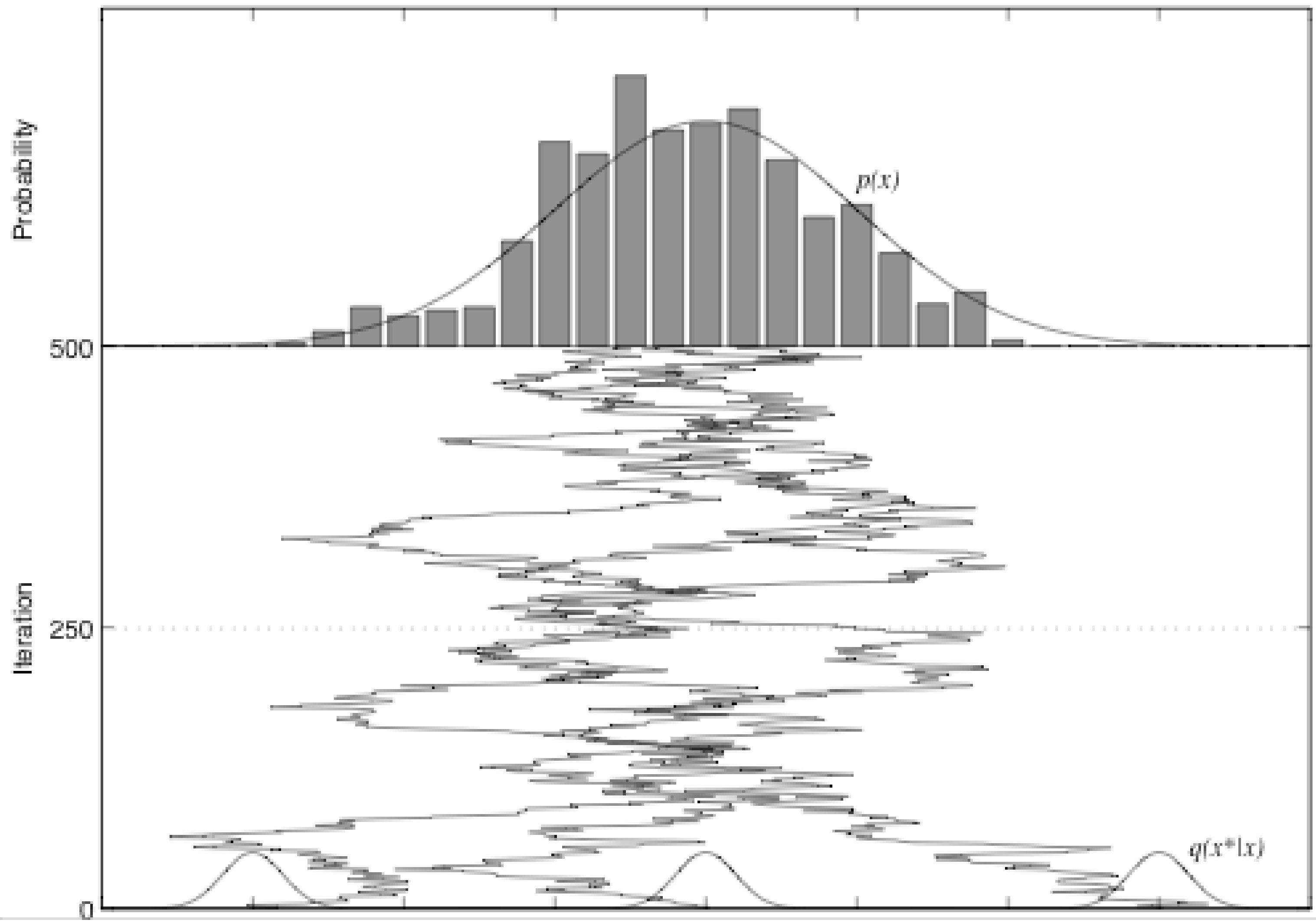
Metropolis-Hastings algorithm



Metropolis-Hastings algorithm



Metropolis-Hastings in a slide



Metropolis-Hastings algorithm

- For right stationary distribution, we want

$$\int \pi(\mathbf{x})T(\mathbf{x}, \mathbf{y})d\mathbf{x} = \pi(\mathbf{y})$$

- Sufficient condition is detailed balance:

$$\pi(\mathbf{x})T(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y})T(\mathbf{y}, \mathbf{x})$$

Metropolis-Hastings algorithm

$$\begin{aligned} T(\mathbf{x},\mathbf{y}) &= Q(\mathbf{y}|\mathbf{x})A(\mathbf{x},\mathbf{y}) \\ &= Q(\mathbf{y}|\mathbf{x}) \min\left\{1, \frac{\pi(\mathbf{y})Q(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})Q(\mathbf{y}|\mathbf{x})}\right\} \end{aligned}$$

$$\begin{aligned} \pi(\mathbf{x})T(\mathbf{x},\mathbf{y}) &= \pi(\mathbf{x})Q(\mathbf{y}|\mathbf{x}) \min\left\{1, \frac{\pi(\mathbf{y})Q(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})Q(\mathbf{y}|\mathbf{x})}\right\} \\ &= \min\left\{\pi(\mathbf{x})Q(\mathbf{y}|\mathbf{x}), \pi(\mathbf{y})Q(\mathbf{x}|\mathbf{y})\right\} \end{aligned}$$

This is symmetric in (**x**,**y**) and thus satisfies detailed balance