# Lecture 14
# Monte Carlo Sampling

**Lei Li** and Yuxiang Wang
UCSB

# Why MC Sampling?

- Goal: Generate samples from a distribution $p(x)$
- Important in physics, economics, statistics, and CS.

# Application of MC

- Bayesian inference:
  - Compute Expectation

$$E[f(x)] = \int f(x)p(x)dx$$

  - Used in EM alg.
- Bayesian optimization
  - find optimal: $\arg\max_x f(x)$

# Monte Carlo Principle

- Goal: to estimate $E[f(x)] = \int f(x)p(x)dx$
- sample x1, … xN (iid) from a distribution p(x),
- compute $\hat{s} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$
- By strong law of large numbers

$$\hat{s} \xrightarrow[N\to\infty]{a.s.} E[f(x)]$$

# Theorem

- The estimate is unbiased

$$E[\hat{s}] = E[f(x)]$$

- $$Var[\hat{s}] = \frac{Var(f)}{N} = O(\frac{1}{N})$$

# Challenges

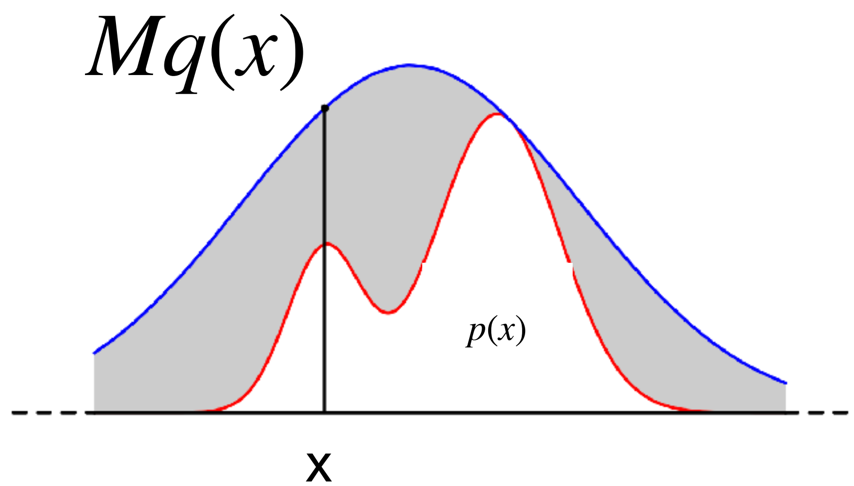
- p(x) may not be possible to efficiently sample from
  - e.g. Cauchy distribution
  - a posterior distribution p(z|x) without closed form
  - p(x) may be un-normalized
- The samples might not be i.i.d.
  - as in MCMC

# MC Sampling methods

- Rejection sampling
- Importance sampling
- Markov chain Monte Carlo (MCMC)
  - Metropolis-Hastings sampling
  - Gibbs sampling
  - Hamiltonian Monte Carlo (HMC)
  - Langevin Monte Carlo
- Sequential Monte Carlo
  - Particle filter

# Rejection Sampling

- Instead of directly sample from p(x), sample from q(x)
- Repeat:
  1. sample $x_i \sim q(x)$
  2. sample $u \sim U[0,1]$
  3. if $u < \dfrac{p(x_i)}{Mq(x_i)}$, then accept $x_i$, otherwise reject. (M is constant)



# Limitation of Rejection Sampling

- Need to compute the upper bound of ratio $p(x)/q(x)$, not always possible
- Acceptance rate is small if M is large
- Acceptance rate exponentially small when dimensionality is large.

# Importance Sampling

- Sampling from proposal distribution q(x)
- Compute importance weight $w(x) = \dfrac{p(x)}{q(x)}$
- Estimate $\hat{s}_{IS} = \dfrac{1}{N}\sum_{i=1}^{N} w(x_i) f(x_i)$

# Importance Sampling

- p(x) can be un-normalized, need reweighing ==> Sampling Importance Re-sampling
- proposal q(x) must be non-zero when p(x)>0
- Theorem: $E[\hat{s}_{IS}] = E[f(x)]$ (unbiased)

$$\hat{s}_{IS} \xrightarrow[N \to \infty]{a.s.} E[f(x)] = \int f(x)p(x)dx$$

# How to choose proposal for Importance Sampling?

- Find one that minimizes the variance of the estimator

$$Var_{q(x)}[\hat{s}] = E_{q(x)}[f(x)^2 w((x)^2] - E[f(x)]^2$$

- Theorem:
  - the variance is minimal when choosing the following optimal importance distribution

$$q^*(x) = \frac{|f(x)|p(x)}{\int |f(x)|p(x)dx}$$

  - not always possible to directly sample from.

# Sampling Importance Re-sampling

- Sampling $x_1 \ldots x_N$ from proposal distribution q(x)
- Compute importance weight $w(x) = \dfrac{p(x)}{q(x)}$
- Compute normalized weight $\hat{w}_i = \dfrac{w_i}{\sum_j w_j}$
- Sampling $\tilde{x}_1 \ldots \tilde{x}_N$ with replacement from $\{x_1 \ldots x_N\}$ with probability $\{\hat{w}_i\}$

# Markov chain Monte Carlo

# Markov chain

- Markov chain $p(x_{n+1} \mid x_1 \ldots x_n) = p(x_{n+1} \mid x_n)$

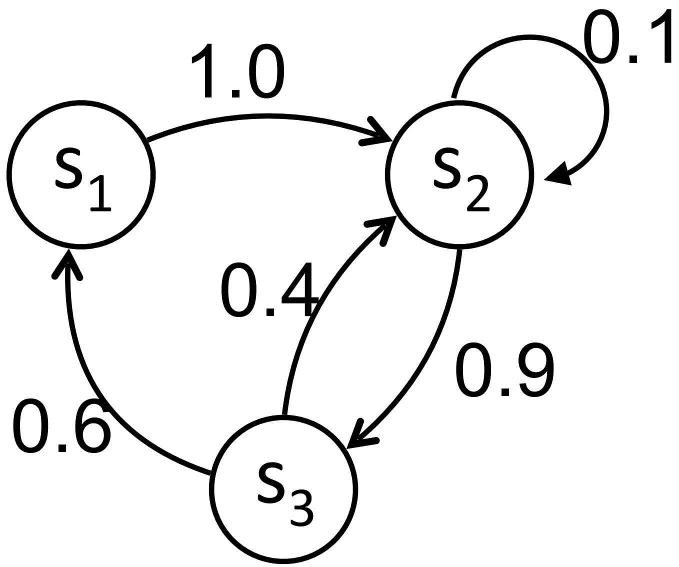
$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$$

- Transition probability:
  $T(x_n, x_{n+1}) = p(x_{n+1} \mid x_n)$
- A distribution $h(x)$ is **stationary** if
  $$h(x) = \int T(x', x) h(x') dx'$$

# Example

State transition graph

(probabilistic finite state machine)

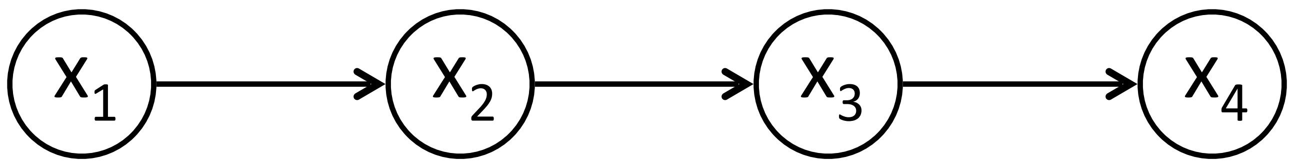



Transition prob.

$$T = \begin{bmatrix} 0 & 1.0 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$

# Detailed Balance and Reversible chain

- Markov chain $p(x_{n+1} \mid x_1 \ldots x_n) = p(x_{n+1} \mid x_n)$




- Sufficient condition (but not necessary) for stationary is *detailed balance* property

$$h(x)T(x, x') = h(x')T(x', x)$$

- A Markov chain satisfying detailed balance property is **reversible**

# Example of Stationary distribution

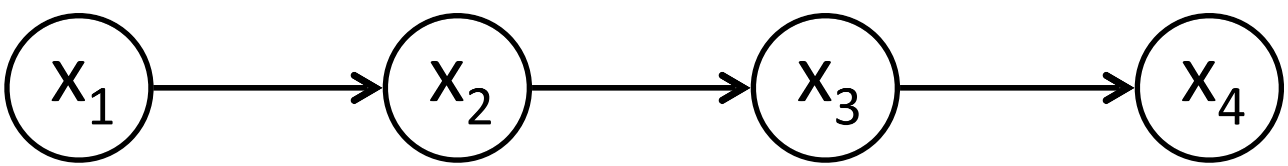
- (0.22, 0.41, 0.37)

$$T = \begin{bmatrix} 0 & 1.0 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$

$$(0.22, 0.41, 0.37) \begin{bmatrix} 0 & 1.0 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix} = (0.22, 0.41, 0.37)$$

# Markov chain Monte Carlo

- Main idea:
  - construct a Markov chain, so that its stationary distribution is our target distribution
  - starting from some initial samples, keep updating the samples from Markov chain according to transition prob.
  - as we sample sufficiently large steps, the samples will converge to stationary distribution (under certain condition, e.g. ergodic)



# Metropolis-Hastings Algorithm

1. start from an initial sample x0,
2. Iterate:

   (1) sample $x_{new}$ from a proposal $q(x \mid x_t)$

   (2) compute acceptance ratio

$$A(x_{new}, x_n) = \min\left(1, \frac{p(x_{new})q(x_n \mid x_{new})}{p(x_n)q(x_{new} \mid x_n)}\right)$$

   (3) sample $u \sim U(0,1)$

   (4) if $u < A(x_{new}, x_n)$, (accept) $x_{n+1} \leftarrow x_{new}$

   (5) otherwise $x_{n+1} = x_n$ (reject)

# Correctness of MH algorithm

- Theorem:
  - The transition kernel of the Markov chain defined by MH algorithm satisfy detailed balance condition.
  - Therefore p(x) is the stationary distribution of this Markov chain
- What is the transition kernel?

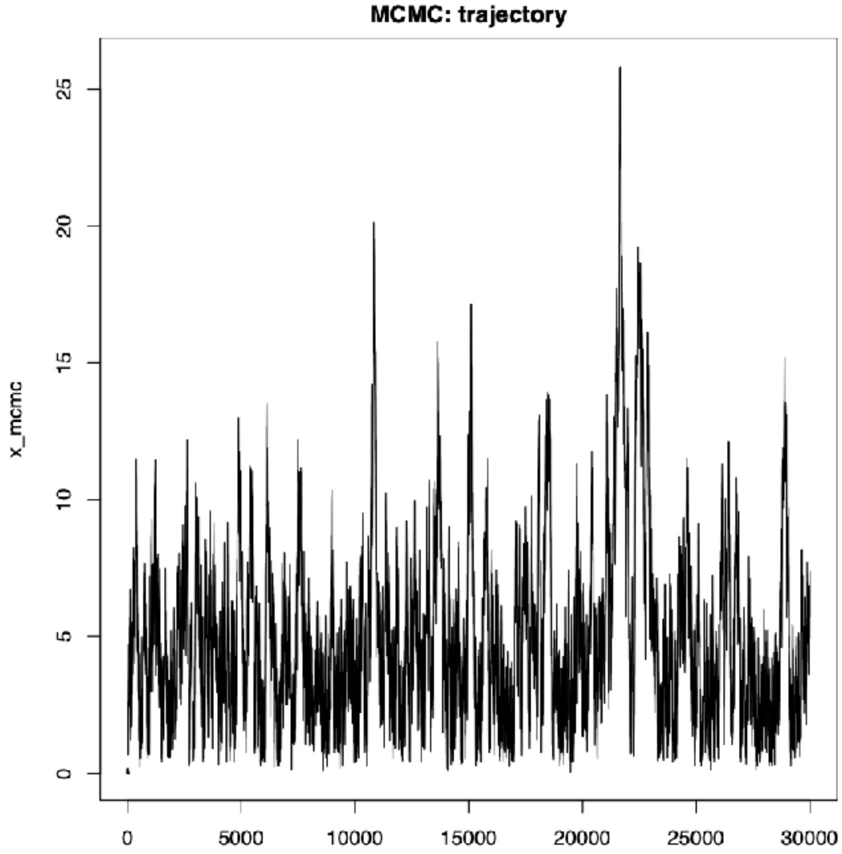
$$T(x, y) = q(y \mid x)A(y, x) + \delta(x = y)(1 - r(x))$$

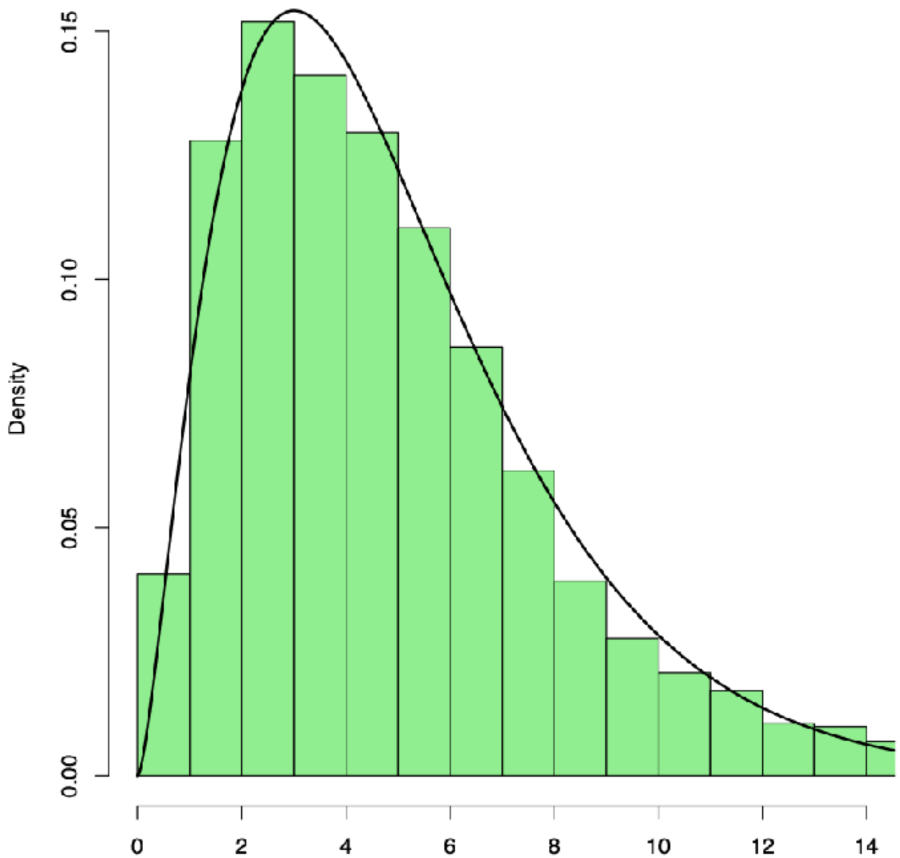
$$r(x) = \int q(y \mid x)A(y, x)dy$$

# Example: sampling Chi-squared distribution

$$\frac{1}{2^{k/2}\Gamma(k/2)}\, x^{k/2-1} e^{-x/2}$$

proposal: N(x, 0.5^2)



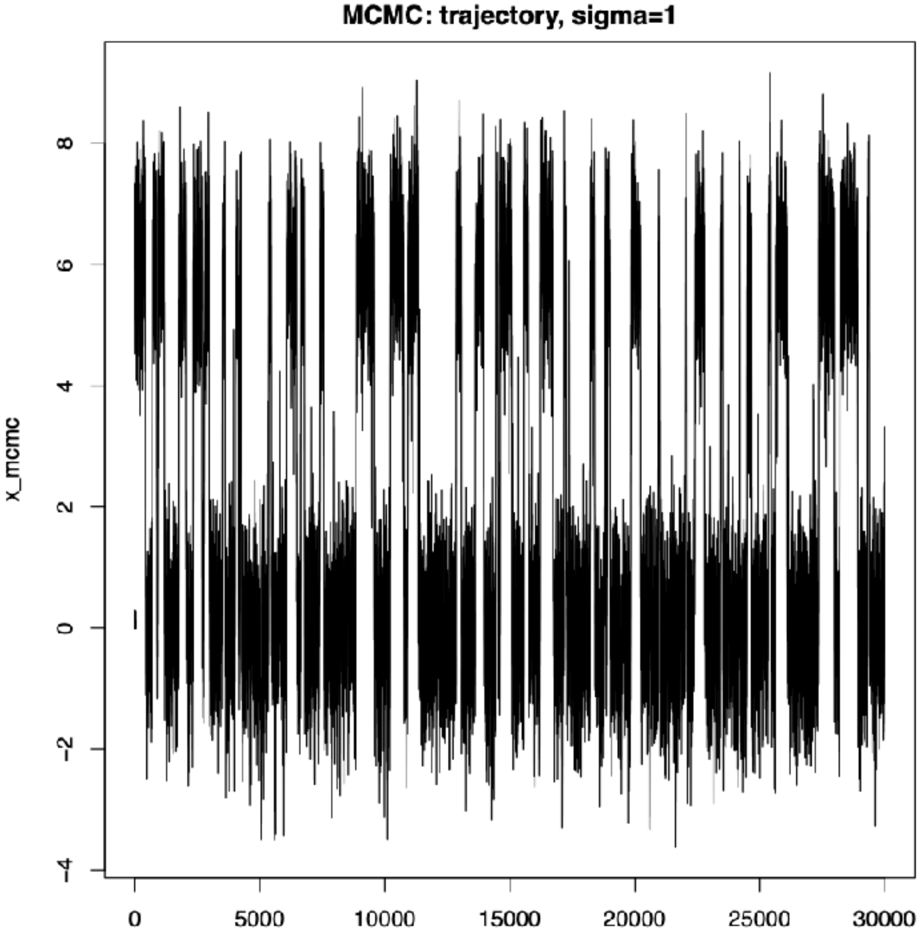



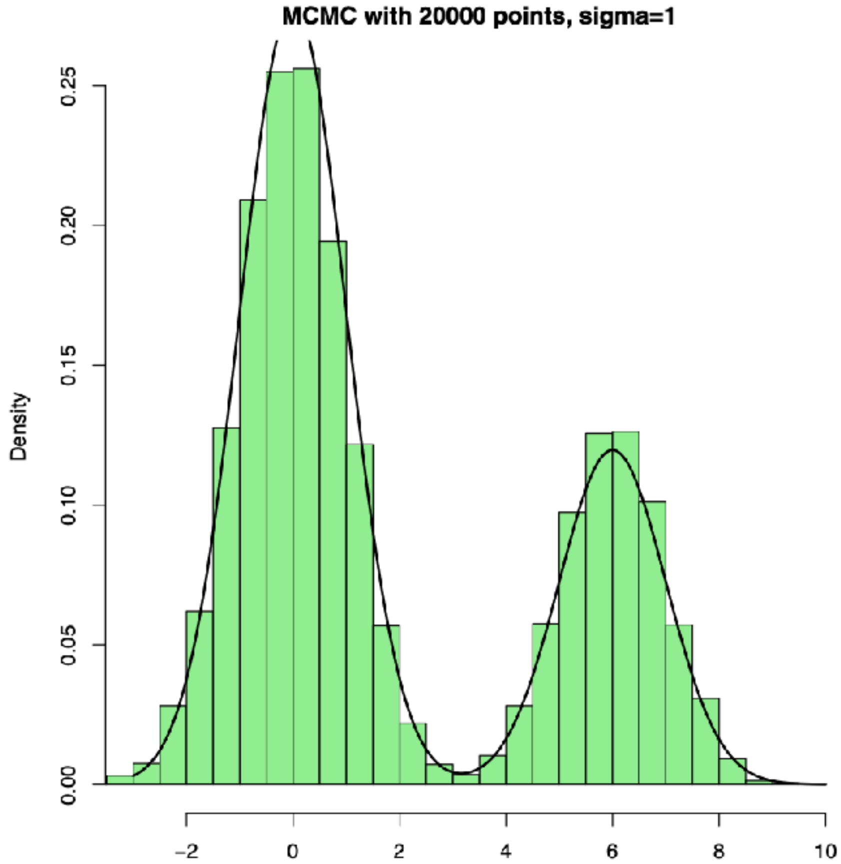

trace plot, to examine the behavior

# Example 2: Mixture of Gaussian

$$p(x) = 0.7\phi(x; 0, 1) + 0.3\phi(x; 5, 1)$$

proposal: N(x, 1.0)

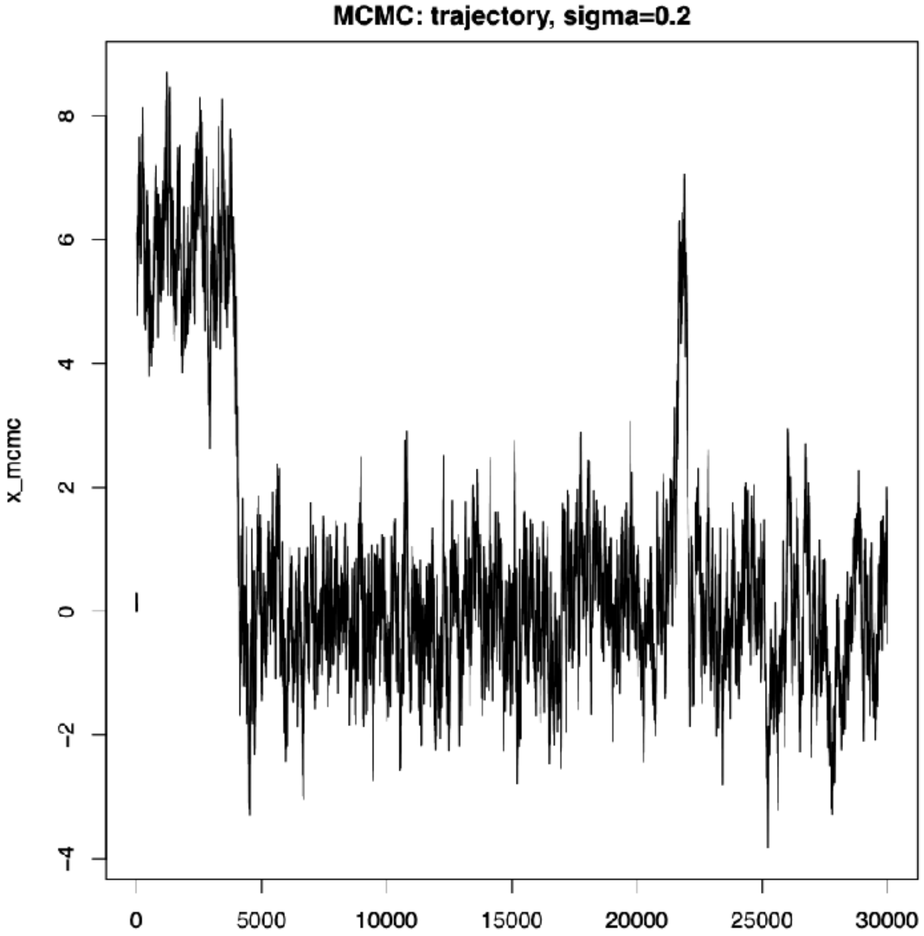


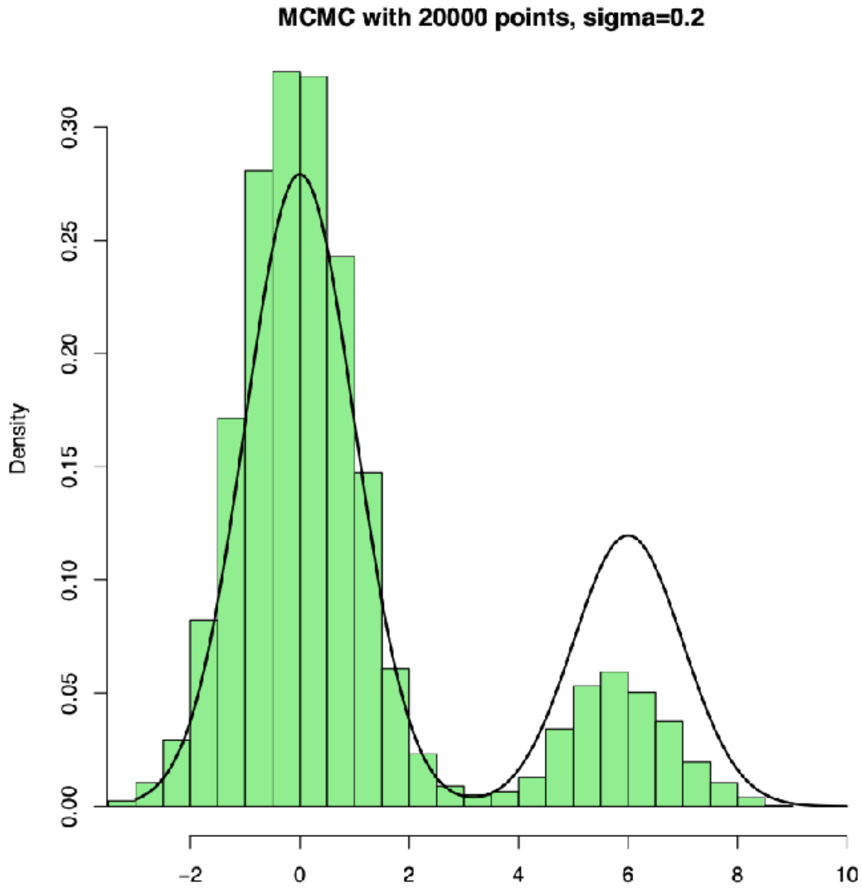



# Example 2: Mixture of Gaussian

$$p(x) = 0.7\phi(x; 0, 1) + 0.3\phi(x; 5, 1)$$

proposal: N(x, 0.2^2)



# Gibbs sampling

- Special case of MCMC
- Sampling two variables x1, x2,
- if we choose proposal distribution to be

$$q(x_1^{new} \mid x_1^{old}, x_2^{old}) = p(x_1 \mid x_2^{old})$$
$$q(x_2^{new} \mid x_1^{old}, x_2^{old}) = p(x_2 \mid x_1^{old})$$

- Much easier to implement if the conditional probability can be calculated.
- In practice, use collapsed Gibbs sampling

# More advanced MCMC

- Reversible jump MCMC
  - if we have varying numbers of variables to sample (i.e. the dimensionality changes)
  - see [Peter Green, 1995]
- Hamiltonian Monte Carlo
  - use gradient information to perform deterministic sampling over one sampling pass

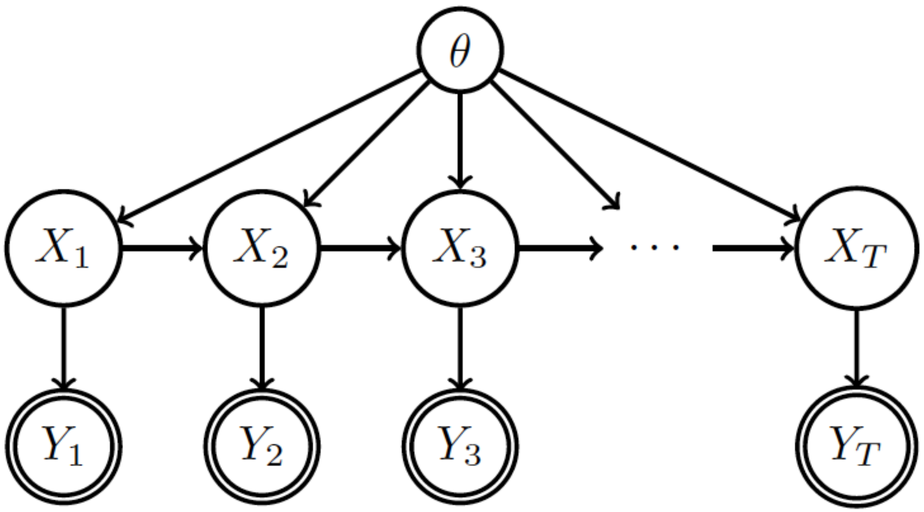
# Monte Carlo EM

- Iterate until convergence
  1. E step: use X, current $\theta$, and a proposal distribution q, to sample from $p(z_{1..N} \mid x_{1..N}; \theta)$
  2. M step, maximization over samples

$$\theta \leftarrow \underset{\theta}{\operatorname{argmax}} E_{p(z_{1..N} \mid x_{1..N}; \theta_{old})} \log p(x_n,\ z_n \mid \theta)$$

# Sequential Monte Carlo

# General State Space Model




$$x_t = f_\theta(x_{t-1}) + v_t$$

$$y_t = g_\theta(x_t) + w_t$$

noise

Goal:

To have an online Bayesian algorithm that can track $p(\theta|y_1..y_T)$

Challenge:

Simultaneous estimation of static parameters and dynamic variables for nonlinear dynamics and nonGaussian noises

# An (extremely) Simplified Example

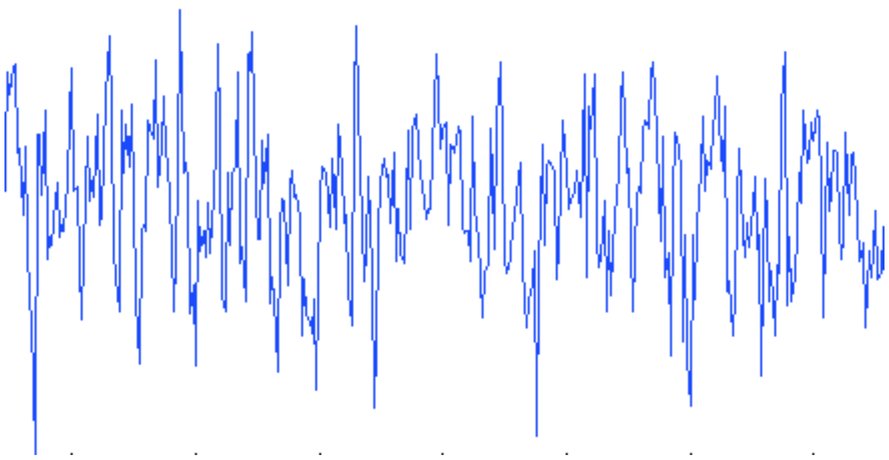
$$x_t = \sin(\theta x_{t-1}) + v_t,\ v_t \sim N(0, \sigma^2)$$

$$y_t = x_t + w_t,\ w_t \sim N(0, \sigma^2_{\text{obs}})$$

Observation: $y_1 \ldots y_T$

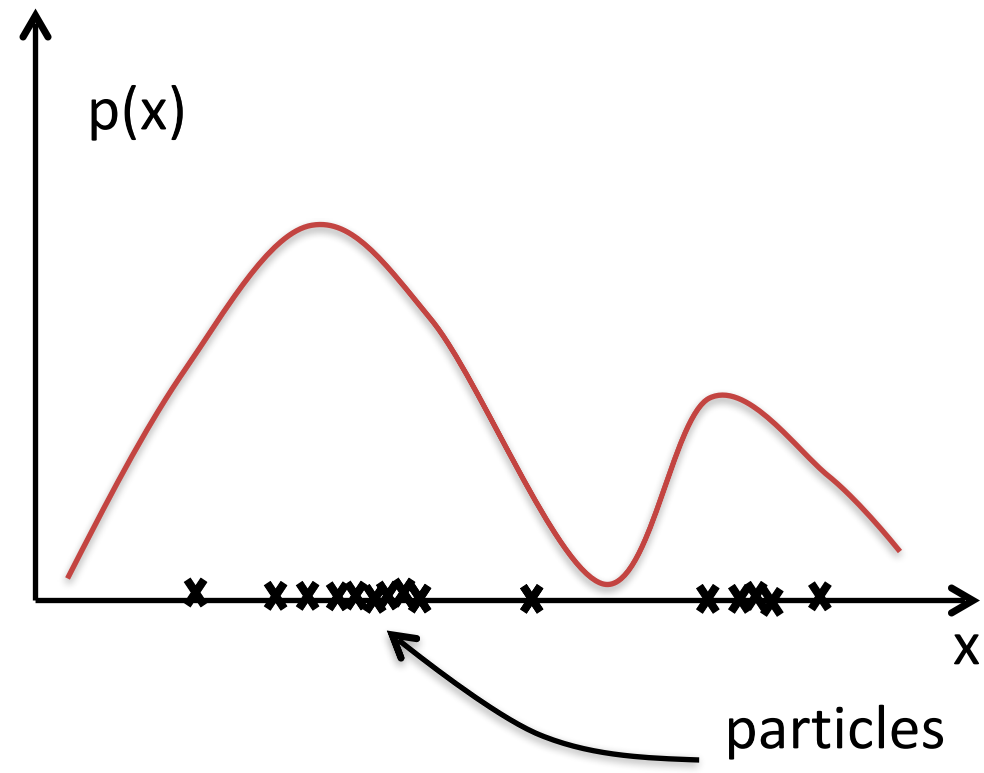
To estimate θ

$p(\theta | y_1 .. y_T)$



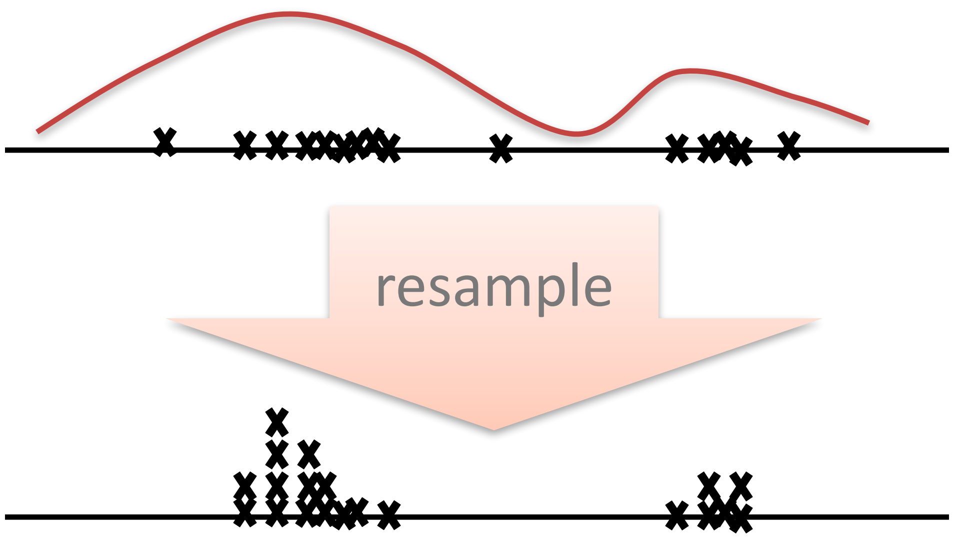
# Particle filter
# (Sequential importance sampling with re-sampling)

- At time tick t=1,
  - Sample $x_1 \sim p(x_1)$, get N particles $x_1^i$.



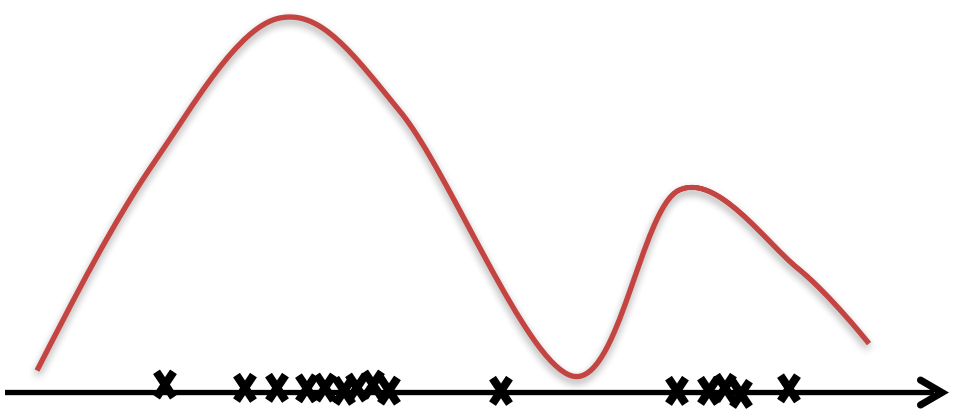
# Particle filter

- At time tick $t=1$,
  - Sample $x_1 \sim p(x_1)$, get N particles $x_1^i$.
  - Weight each particle with $w^i = p(y_1 \mid x_1^i)$
  - Resample $x_1^i$ w.r.t. weight $w^i$



# Particle filter

- At time tick t=1,
  - Sample $x_1 \sim p(x_1)$, get N particles $x_1^i$.
  - Weight each particle with $w^i = p(y_1 \mid x_1^i)$
  - Resample $x_1^i$ w.r.t. weight $w^i$
- At time tick t > 1,
  - Sample $x_t \sim f_\theta(x_t \mid x_{t-1})$, get N particles



# Particle filter

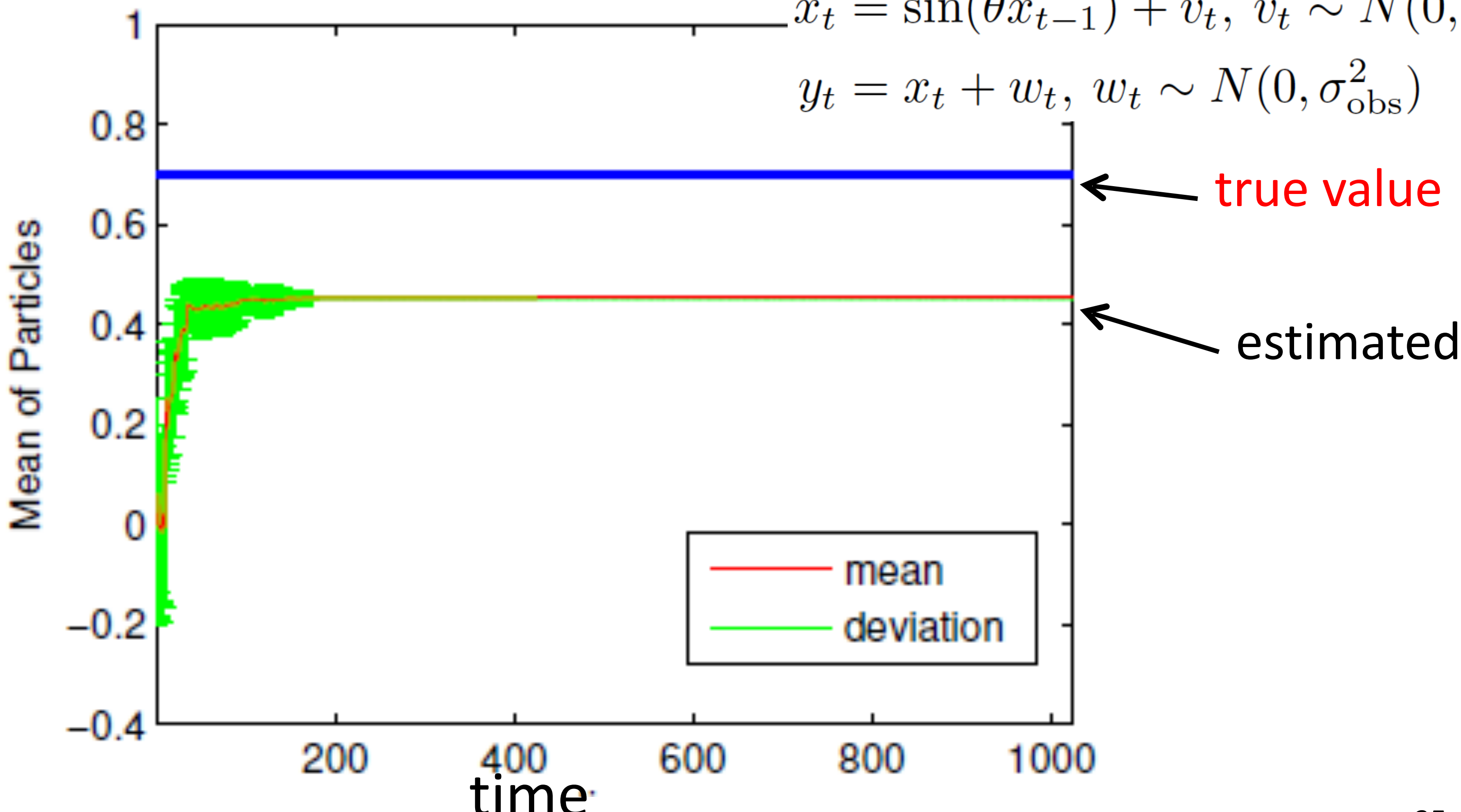
- At time tick t=1,
  - Sample $x_1 \sim p(x_1)$, get N particles $x_1^i$.
  - Weight each particle with $w^i = p(y_1 \mid x_1^i)$
  - Resample $x_1^i$ w.r.t. weight $w^i$
- At time tick t > 1,
  - Sample $x_t \sim f_\theta(x_t \mid x_{t-1})$, get N particles
  - Weight each particle with $w^i = p(y_t \mid x_t^i)$
  - Resample w.r.t. weight $w^i$

# Particle Filter does not work for static variable (parameter)

Mean of particles (θ)

$$x_t = \sin(\theta x_{t-1}) + v_t,\ v_t \sim N(0, \sigma^2)$$

$$y_t = x_t + w_t,\ w_t \sim N(0, \sigma^2_{\text{obs}})$$



# Summary

- Monte Carlo sampling is very useful if the density is hard to compute
- MCMC: construct a Markov chain with the target distribution being its stationary distribution
- Metropolis-Hastings algorithm

# Next Up

- Convex Optimization