NUS National University of Singapore | School of Computing

CS5340
Uncertainty Modeling in AI

Lecture 9:
Monte Carlo Inference

(or “how to do approximate inference with samples”)

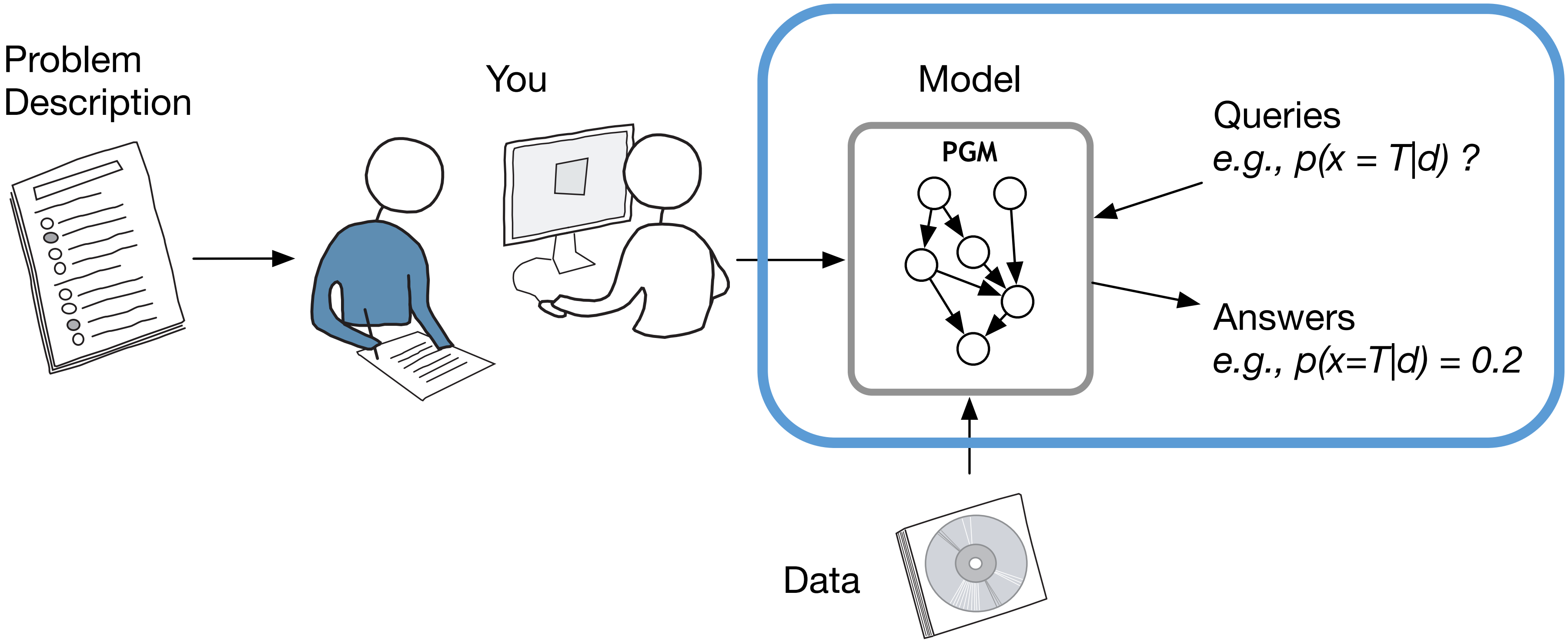
Asst. Prof. Harold Soh

AY 2022/23

Semester 2

CS5340 in a nutshell

CS5340 is about how to "**represent**" and "**reason**" with uncertainty in a computer.



Course Schedule

Week	Date	Lecture Topic	Tutorial Topic
1	12 Jan	Introduction to Uncertainty Modeling + Probability Basics	Introduction
2	19 Jan	Simple Probabilistic Models	Probability Basics
3	26 Jan	Bayesian networks (Directed graphical models)	More Basic Probability
4	2 Feb	Markov random Fields (Undirected graphical models)	DGM modelling and d-separation
5	9 Feb	Variable elimination and belief propagation	MRF + Sum/Max Product
6	16 Feb	Factor graph and the junction tree algorithm	**Quiz 1**
-	-	RECESS WEEK	
7	2 Mar	Mixture Models and Expectation Maximization (EM)	Linear Gaussian Models
8	9 Mar	Hidden Markov Models (HMM)	Probabilistic PCA
9	**16** Mar	Monte-Carlo Inference (Sampling)	Linear Gaussian Dynamical System
10	23 Mar	Variational Inference	MCMC + Sequential VAE
11	30 Mar	Inference and Decision-Making (Special Topic)	**Quiz 2**
12	6 Apr	Gaussian Processes (Special Topic)	Wellness Day
13	13 Apr	**Project Presentations**	Closing

Learning Outcomes

- Students should be able to:

1. Explain the Monte Carlo principle.
2. Apply the Importance Sampling technique for computing expectations.
3. Apply Rejection, Metropolis-Hasting, Metropolis and Gibbs sampling methods to perform approximate inference.
4. Use Markov chain properties, i.e. homogenous, stationary distribution, irreducibility, aperiodicity, egordicity and detail balance, to show validity of MH algorithm.

Acknowledgements

- A lot of slides and content of this lecture are adopted from:

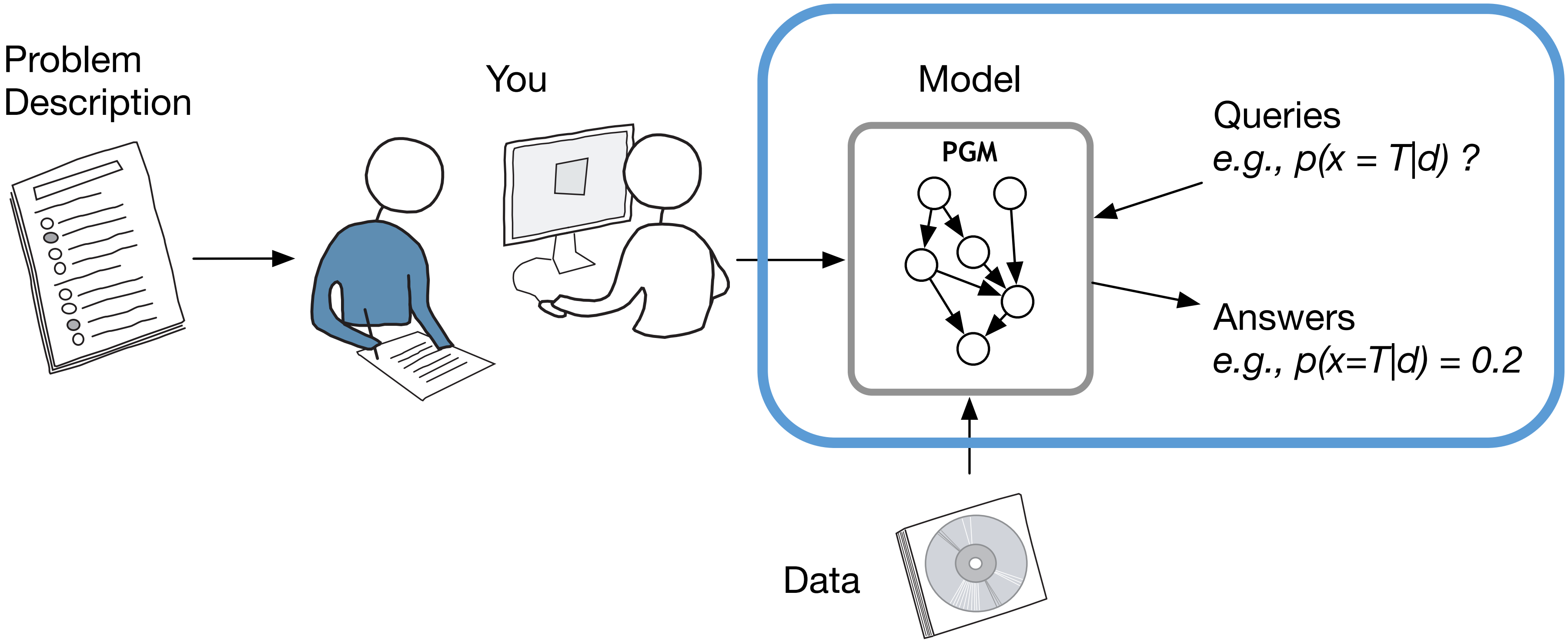
1. "An introduction to MCMC for Machine Learning", Christophe Andrieu et. al. (In Extra Readings on Piazza)
2. "Pattern Recognition and Machine Learning", Christopher Bishop, Chapter 11.
3. http://www.cs.cmu.edu/~epxing/Class/10708/lectures/lecture16-MC.pdf http://www.cs.cmu.edu/~epxing/Class/10708/lectures/lecture17-MCMC.pdf, Eric Xing, CMU.
4. "Machine Learning – A Probabilistic Perspective", Kevin Murphy, Chapter 23.
5. "Probabilistic Graphical Models", Daphne Koller and Nir Friedman, chapter 12.
6. Lee Gim Hee's slides

Monte-Carlo Sampling: Introduction & Motivation

Notion and Motivation

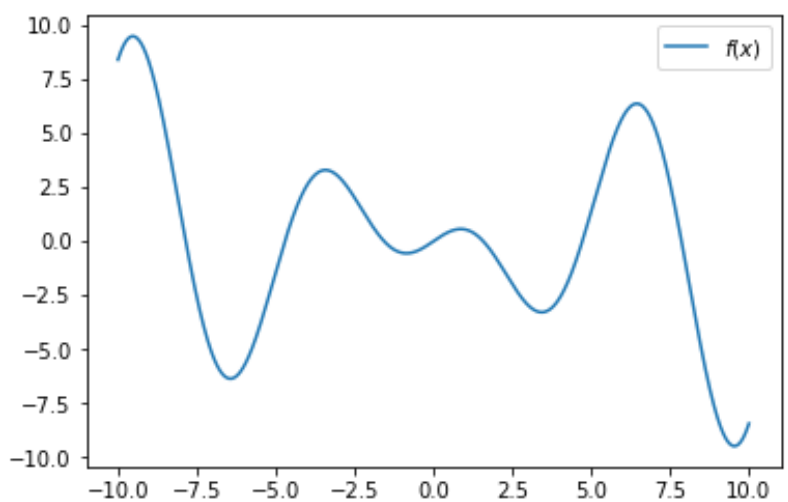
CS5340 in a nutshell

CS5340 is about how to “**represent**” and “**reason**” with uncertainty in a computer.



Key Ideas

- What if you cannot perform exact inference?
- Perform *approximate* inference via sampling
- Want "good" samples



$$\mathbb{E}_{x \sim p(x)}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$
$$x_i \sim p(x)$$

History of Monte Carlo Sampling

- Invented by Stan Ulam in 1946 when he was playing solitaire
- Compute the chances of a successful game outcome.
- First attempted exhaustive combinatorial calculations.
- Decided to lay out several games at random and then counting the number of successful plays.
- Recognized computers made this practical!

Stanislaw Ulam
1909-1984

Metropolis, Nicholas, and Stanislaw Ulam. "The monte carlo method." *Journal of the American statistical association* 44.247 (1949): 335-341.

Image source: https://en.wikipedia.org/wiki/Stanislaw_Ulam

Idea Behind Monte Carlo Sampling

Ulam's idea of selecting a statistical sample to approximate a hard combinatorial problem by a much simpler problem is at the heart of modern Monte Carlo simulation.

Use randomness to solve a possibly deterministic problem.

Monte-Carlo

Pioneers of Monte Carlo Sampling

Stanislaw Ulam
1909-1984

John von Neumann
1903-1957

Nicholas Metropolis
1915-1999

Marshall Rosenbluth
1927-2003

Arianna Rosenbluth
1927-

Edward Teller
1908-2003

Augusta H. Teller
1909-2000

Image from: https://str.llnl.gov/str/May07/pdfs/05_07.5.pdf

Metropolis algorithm is selected as one of the top 10 algorithms that had the greatest influence on science and engineering in the 20th century.

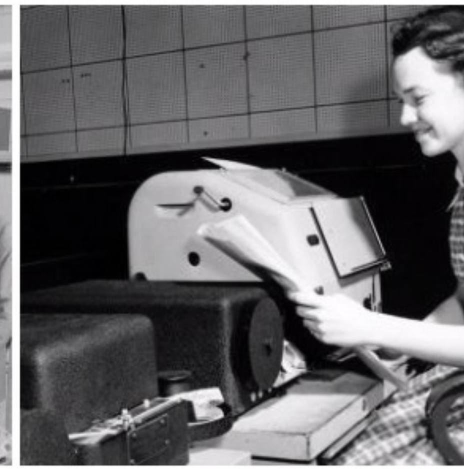
[Beichl & Sullivan 2000]

The MANIAC

The MANIAC I at Los Alamos in 1952. Photo courtesy of LANL.

Early IBM calculating machines at Los Alamos

Paul Stein and Nick Metropolis playing modified chess with the MANIAC I



A MANIAC coding operator

Image Credit: https://www.atomicheritage.org/history/computing-and-manhattan-project

Why Do We Need Sampling?

Bayesian inference and learning:

Given some unknown variables $X \in \mathcal{X}$ and data $Y \in \mathcal{Y}$, the following *typically* intractable integration problems are central to Bayesian statistics.

1. **Normalization**. To obtain the posterior $p(x \mid y)$ given the prior $p(x)$ and likelihood $p(y \mid x)$, the normalizing factor in Bayes' theorem needs to be computed

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{\int_{\mathcal{X}} p(y \mid x')p(x')\,dx'}$$

Can be intractable to compute

Why Do We Need Sampling?

2. **Marginalization**: Given the joint posterior of $(X, Z) \in \mathcal{X} \times \mathcal{Z}$, we may often be interested in the marginal posterior.

$$p(x \mid y) = \int_{\mathcal{Z}} p(x, z \mid y)\, dz$$

Can be intractable to compute

3. **Expectation**: The objective of the analysis is often to obtain summary statistics of the form

$$\mathbb{E}_{p(x|y)}(f(x)) = \int_{\mathcal{X}} f(x) p(x \mid y)\, dx$$

Can be intractable to compute

for some function of interest $f : \mathcal{X} \to \mathbb{R}^{n_f}$ integrable with respect to $p(x \mid y)$.

From Lecture 7: The General EM Algorithm

1. Choose an initial setting for the parameters θ^{old}.

2. Expectation step: Evaluate $p(\mathrm{Z}|\mathrm{X}, \theta^{old})$.

3. Maximization step: Evaluate θ^{new} given by:

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}})$$

where

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

4. Check for convergence of either the log likelihood or the parameter values, if not converged:

$$\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$$

Recall the EM Algorithm

- What if the expectation could **not** be performed analytically?

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) = \underbrace{\int p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{\text{old}}) \ln p(\mathbf{Z}, \mathbf{X}|\boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z}}_{\text{Cannot be computed analytically!}}$$

Sampling and the EM Algorithm

- Approximate integral by a finite sum over samples $\{Z^l\}$, drawn from current estimate $p(Z \mid X, \theta^{old})$, then:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{old}}) \simeq \frac{1}{L} \sum_{l=1}^{L} \ln p(\mathbf{Z}^{(l)}, \mathbf{X} | \boldsymbol{\theta})$$

- The Q function is then optimized in the usual way in the M step.

- This procedure is called the Monte Carlo EM algorithm.

Overview

- Sampling Basics:
  - Monte-Carlo Principle
- Basic Sampling Techniques
  - Rejection Sampling
  - Importance Sampling
- Markov Chain Monte-Carlo (MCMC)
  - Metropolis-Hastings Algorithm
  - Theory
  - Gibbs Sampling

The Basics

Monte Carlo Approach and Key Properties

Key Ideas

- The Monte-Carlo Estimate
- Is the Monte-Carlo Estimate a "good" estimate?
- Properties:
  - Unbiased (Bias = 0)
  - Consistent (Converges to the true value as $N \to \infty$)
  - Converges at rate $1/\sqrt{N}$ (independent of dimensionality)

Empirical Point-Mass Function

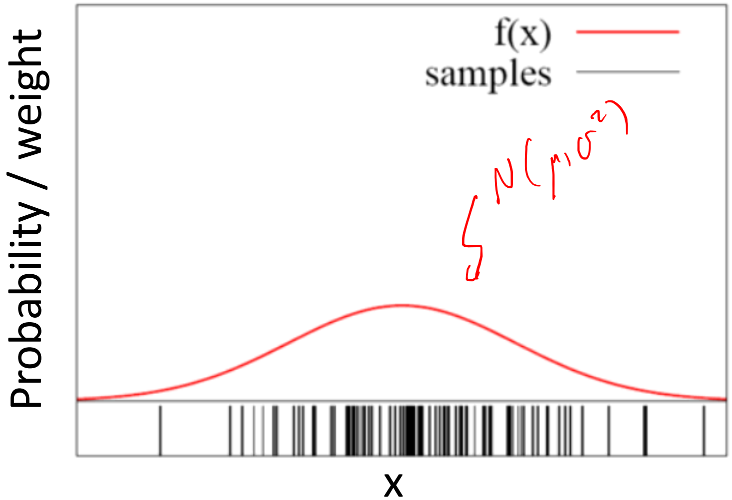
- Draw an i.i.d. set of samples $\{x^{(i)}\}_{i=1}^{N}$ from a target density $p(x)$ defined on a space $\mathcal{X}$.

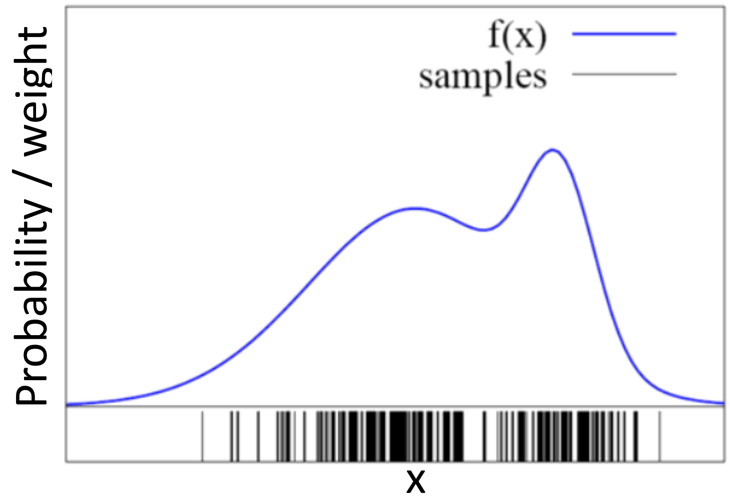
- Approximate the target density $p(x)$ with the following empirical point-mass function:

$$p_N(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^{(i)}}(x)$$

Delta-Dirac mass located at $x(i)$

Non-Parametric Representation





The more samples are in an interval, the higher the probability of that interval.

No restriction on the *type* of distribution (e.g. can be multi-modal, non-Gaussian, etc)

The Monte Carlo Approach

- Approximate the expectation/integrals (or very large sums) $I(f)$ with tractable sums $I_N(f)$

$$I_N(f) = \frac{1}{N}\sum_{i}^{N} f(x^{(i)}) \approx I(f) = \int_{\mathcal{X}} f(x)p(x)dx$$

- $I_N(f)$ is an estimator for $I(f)$
- How "good" of an estimator is $I_N(f)$?

$I_N(f)$ is Unbiased

Bias $= \mathbb{E}[I_N(f)] - I(f) = 0$ (Unbiased)

In other words, $\mathbb{E}[I_N(f)] = I(f)$

Proof:

$$\mathbb{E}[I_N(f)] = \mathbb{E}\left[\frac{1}{N}\sum_{i}^{N} f\left(x^{(i)}\right)\right]$$

$$= \frac{1}{N}\sum_{i}^{N} \mathbb{E}[f\left(x^{(i)}\right)]$$

$$= \frac{1}{N}\sum_{i}^{N} I(f) = I(f)$$

Consistency and Convergence

- By weak law of large numbers, $I_N(f)$ converges in probability to $I(f)$ ("consistent")

$$I_N(f) \xrightarrow{p} I(f) \text{ as } N \to \infty$$

"Converges in probability": $\forall \epsilon > 0, \lim_{N\to\infty} p(|I_N(f) - I(f)| > \epsilon) = 0$

- By strong law of large numbers, $I_N(f)$ converges almost surely to $I(f)$.

$$I_N(f) \xrightarrow{a.s.} I(f) \text{ as } N \to \infty$$

"Converges almost surely": $p\left(\lim_{N\to\infty} I_N(f) = I(f)\right) = 1$

If you want proofs: https://www.randomservices.org/random/sample/LLN.html

Variance of $I_N(f)$

Define the variance of $f(x)$ as:

$$\sigma_f^2 = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2 = \mathbb{E}[f(x)^2] - I(f)^2 < \infty$$

Then,

$$\mathbb{V}[I_N(f)] = \frac{\sigma_f^2}{N}$$

Proof:

$$\mathbb{V}[I_N(f)] = \mathbb{V}\left[\frac{1}{N}\sum_i^N f\left(x^{(i)}\right)\right] = \frac{1}{N^2}\sum_i^N \mathbb{V}[f\left(x^{(i)}\right)]$$

$$= \frac{1}{N^2}\sum_i^N \sigma_f^2 = \frac{N}{N^2}\sigma_f^2 = \frac{\sigma_f^2}{N}$$

Mean Squared Error (MSE)

Mean Square Error (MSE): since $\mathbb{E}[I_N(f)] = I(f)$,

$$\mathrm{MSE}[I_N(f)] = \mathbb{V}[I_N(f)]$$

Why?

$$\begin{aligned}\mathrm{MSE}[I_N(f)] &= \mathrm{Bias}^2 + \mathrm{Variance}\\ &= 0 + \mathbb{V}[I_N(f)]\end{aligned}$$

Note: $\mathrm{MSE}[I_N(f)] = \frac{\sigma_f^2}{N} \to 0$ as $N \to \infty$

See: http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf for bias-variance decomposition of the MSE.

Convergence rate of the Error

Since $\text{MSE}[I_N(f)] = \mathbb{V}[I_N(f)] = \frac{\sigma_f^2}{N}$,

$$\text{STDEV}[I_N(f)] \propto \frac{1}{\sqrt{N}}$$

"Converges at rate $1/\sqrt{N}$"

Note: the above is independent of the dimensionality

By Central Limit Theorem,

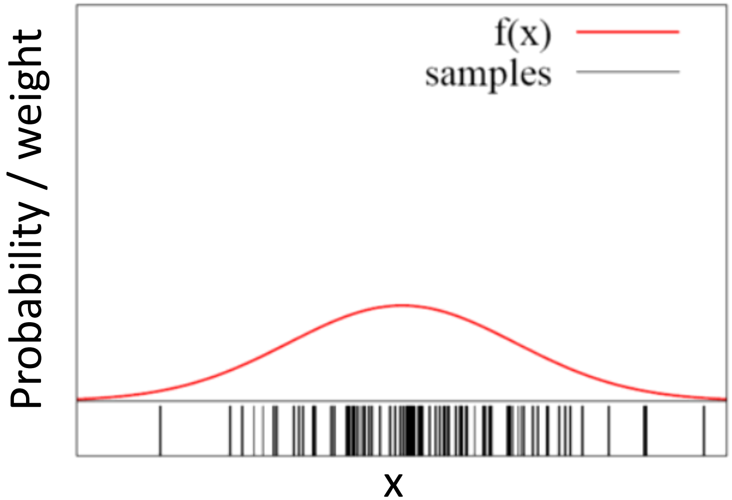
$$\sqrt{N}\left(I_N(f) - I(f)\right) \xrightarrow{d} \mathcal{N}(0, \sigma_f^2) \text{ as } N \to \infty$$

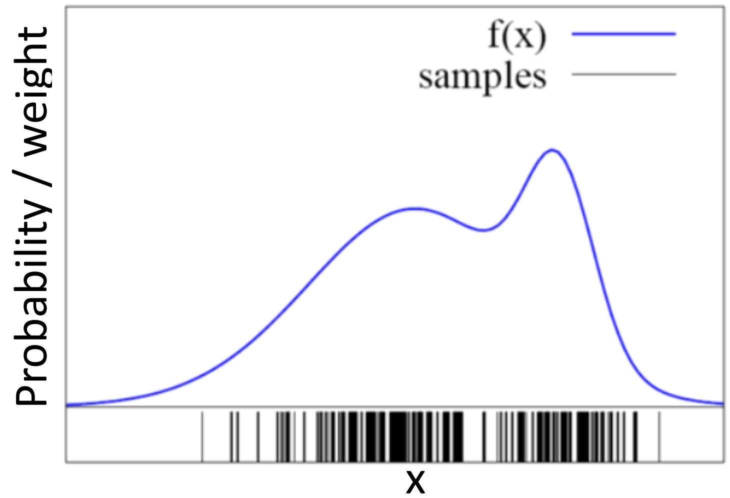
"Converges in distribution": $\lim_{N\to\infty} F_N(x) = F(x)$

Key Ideas

- The Monte-Carlo Estimate
- Properties:
  - Unbiased (Bias = 0)
  - Consistent (Converges to the true value as $N \to \infty$)
  - Converges at rate $1/\sqrt{N}$ (independent of dimensionality)

Empirical Point Mass Function





The more samples are in an interval, the higher the probability of that interval.

But:

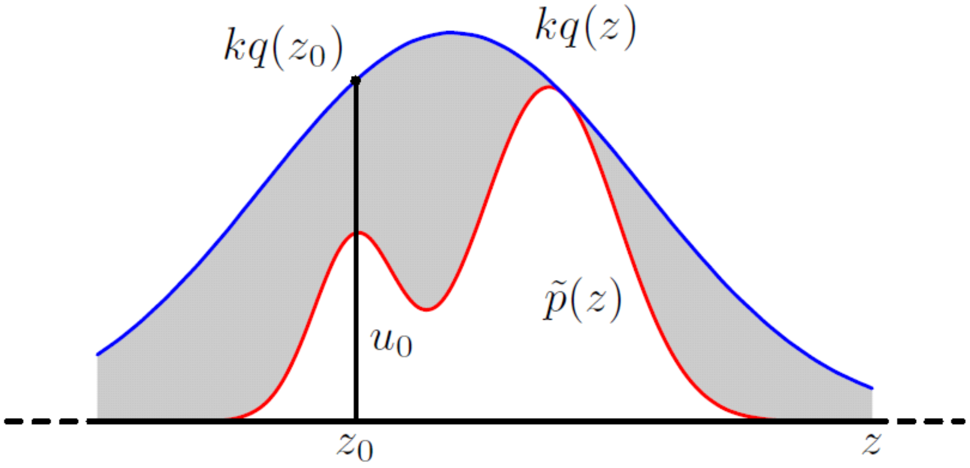
How to draw samples from a function/distribution?

Rejection Sampling

Basic Idea and Algorithm

Key Idea

- **Problem:** What if the distribution is not easy to sample from?
- **Idea:** Sample from a simpler (“proposal”) distribution and randomly accept samples that meet some criteria (“acceptance region”).



Rejection Sampling

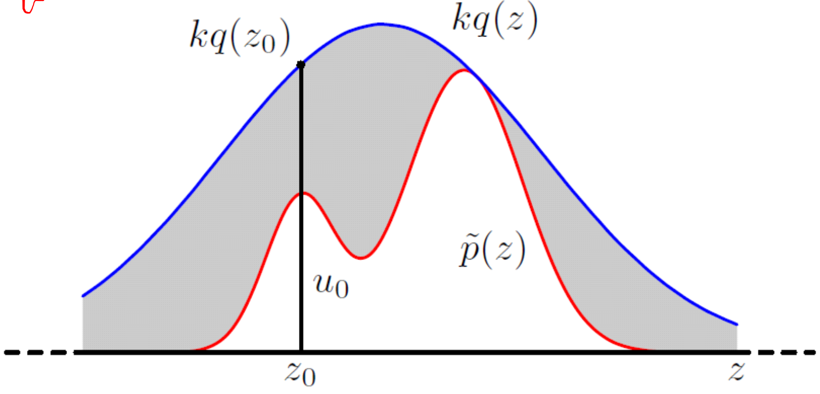
- Want: sample from a distribution $p(z)$, where direct sampling is difficult.

- Unable to easily evaluate $p(z)$ due to an unknown normalizing constant Z_p, so that:

$$p(z) = \frac{1}{Z_p} \widetilde{p}(z)$$

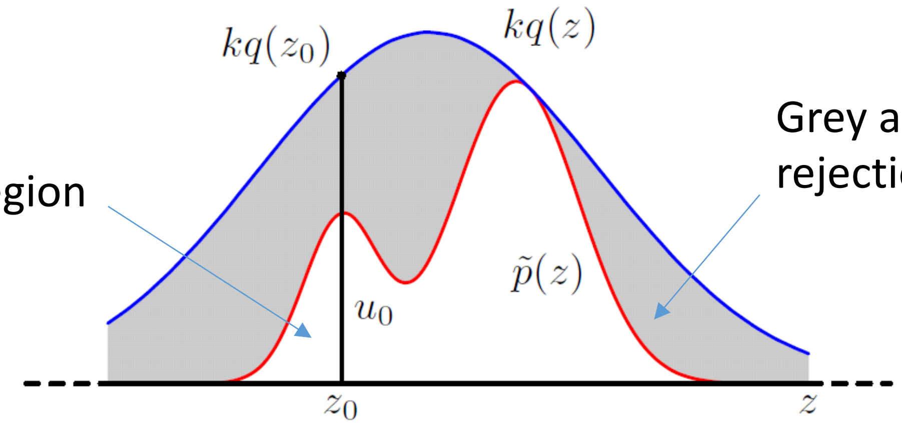
hard to compute! easy to compute!

- But: $\tilde{p}(z)$ can be evaluated.



Rejection Sampling

White area:
acceptance region



Grey area:
rejection region

Algorithm : Rejection Sampling

Set $i = 1$
Repeat until $i = N$ // draw N samples

Proposal distribution $q(z)$ is an easier-to-sample distribution e.g. Gaussian!

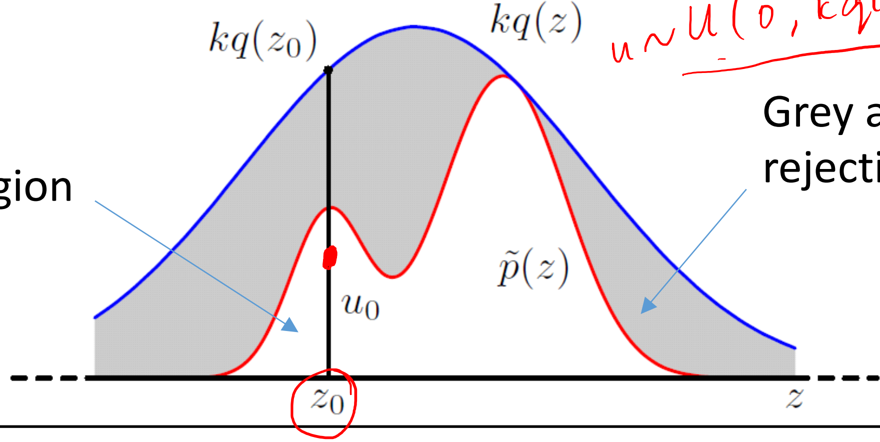
1. Sample $z^{(i)} \sim q(z)$ and $u \sim U_{(0,1)}$ // sample from proposal distribution $q(z)$ // sample from uniform distribution $U_{(0,1)}$
2. If $u < \frac{\tilde{p}(z^{(i)})}{kq(z^{(i)})}$, then accept $z^{(i)}$ and increment the counter i by 1.
3. Otherwise, reject.

Image source: "Machine Learning and Pattern Recognition", Christopher Bishop

Rejection Sampling

$u \sim U(0, kq(z_0))$ $\quad u < \tilde{p}(z_0)$

White area: acceptance region



Grey area: rejection region

$u \sim U(0,1)$

$u < \dfrac{\tilde{p}(z^{(i)})}{kq(z^{(i)})}$

Algorithm : Rejection Sampling

Set $i = 1$

Repeat until $i = N$ // draw N samples

Accept proposal $z^{(i)}$ when u falls in the acceptance region.

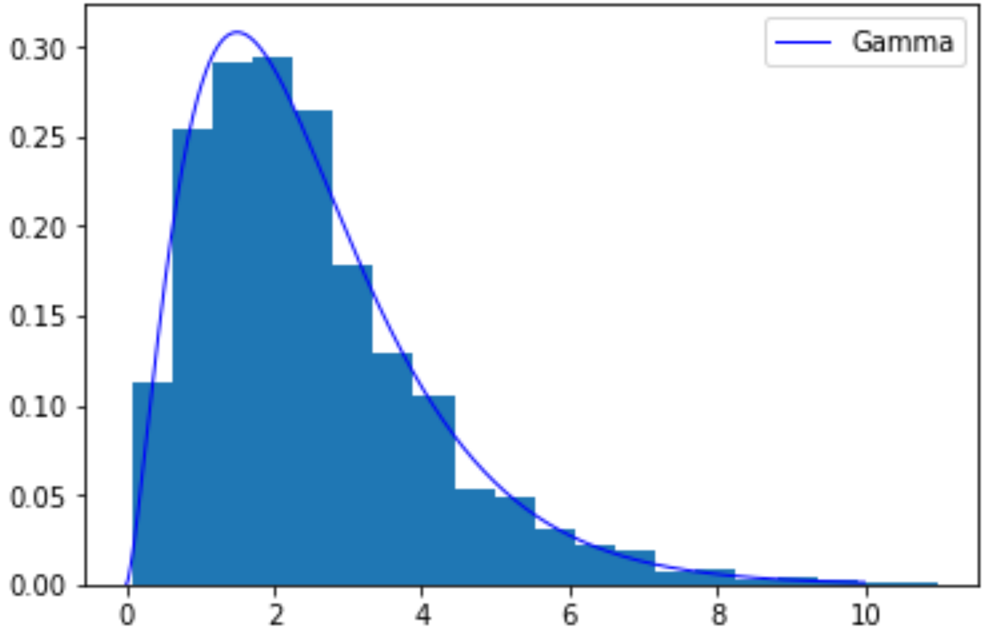
1. Sample $z^{(i)} \sim q(z)$ and $u \sim U_{(0,1)}$ // sample from proposal distribution $q(z)$ // sample from uniform distribution $U_{(0,1)}$

2. If $u < \frac{\tilde{p}(z^{(i)})}{kq(z^{(i)})}$, then accept $z^{(i)}$ and increment the counter i by 1.
3. Otherwise, reject. // accept proposal $z^{(i)}$ if $u < \frac{\tilde{p}(z^{(i)})}{kq(z^{(i)})}$,

// constant k is chosen such that $\tilde{p}(z^{(i)}) \leq kq(z)$for all values of z

Image source: “Machine Learning and Pattern Recognition”, Christopher Bishop

Tutorial Sheet



Rejection Sampling: Limitations

- It is not always possible to bound $\frac{\tilde{p}(z)}{q(z)}$ with a reasonable constant k over the whole space $\mathcal{Z}$.

- If k is too large, the acceptance probability:

$$\Pr(z \text{ accepted}) = \Pr\left(u < \frac{\tilde{p}(z)}{kq(z)}\right) = \frac{1}{k},$$

  will be too small.

- Impractical in high dimensional space scenarios.

NUS National University of Singapore | School of Computing

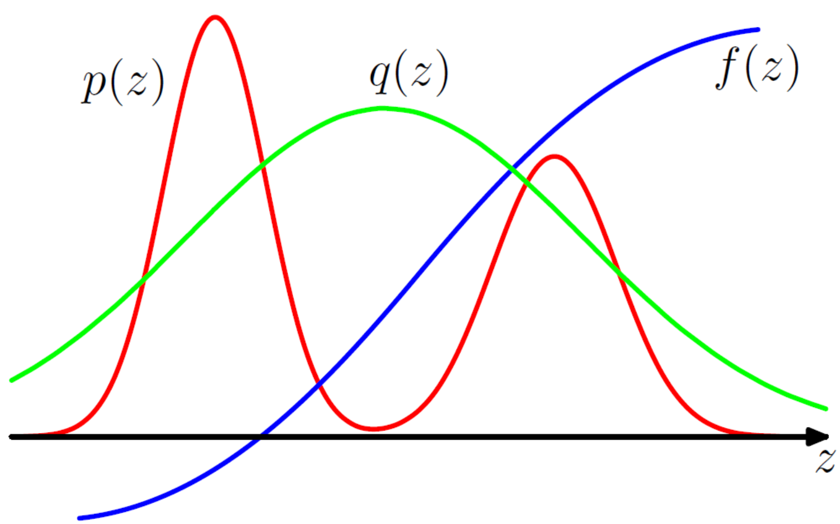
Importance Sampling

“Weighting” Samples

Key Ideas

- How can we use all the samples instead of throwing them away when computing expectations?
- Importance Sampling "trick"
  - Weigh samples.

$$\mathbb{E}[f] \approx \frac{1}{N}\sum_{i}^{N} \underbrace{\frac{p\left(z^{(i)}\right)}{q\left(z^{(i)}\right)}}_{\substack{\text{importance weight} \\ r_i}} f\left(z^{(i)}\right)$$



Importance Sampling

- Given a target distribution $p(x)$ which is difficult to draw samples directly.
- Importance sampling provides a framework for approximating expectations of a function $f(x)$ w.r.t. $p(x)$.
- Samples $\{z^{(i)}\}$ are drawn from a simpler distribution $q(z)$, i.e. proposal distribution.

Importance Sampling

- Express expectation in the form of a finite sum over samples $\{z^{(l)}\}$ weighted by the ratios $p(z^{(l)})/q(z^{(l)})$:

$$\mathbb{E}_p[f] = \int f(z)p(z)dz$$

$$= \int \underbrace{\boxed{f(z)\frac{p(z)}{q(z)}}}_{= g(z)} q(z)dz \qquad \mathbb{E}_q[g(z)]$$

$$\mathbb{E}[f] \approx \frac{1}{N}\sum_{i}^{N} \frac{p(z^{(i)})}{q(z^{(i)})} f(z^{(i)})$$

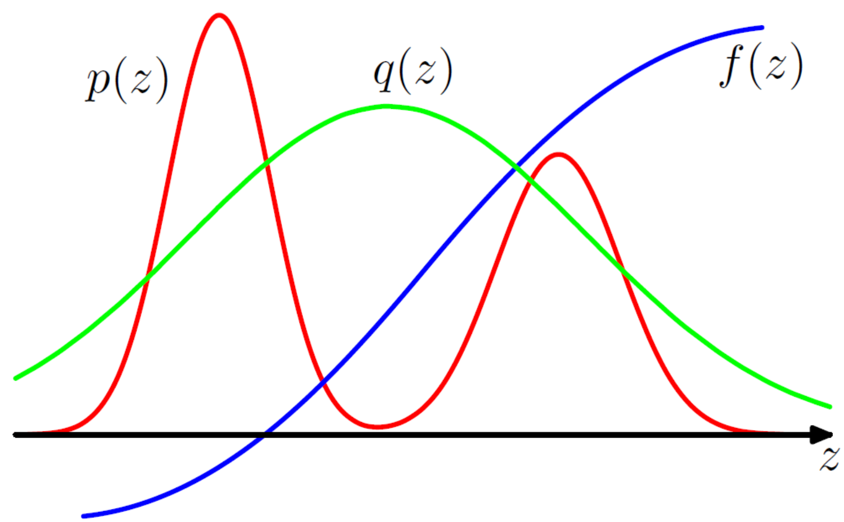


Image source: “Machine Learning and Pattern Recognition”, Christopher Bishop

Importance Sampling

$$\mathbb{E}[f] \approx \frac{1}{N}\sum_{i}^{N} \underbrace{\frac{p\left(z^{(i)}\right)}{q\left(z^{(i)}\right)}}_{r_i} f\left(z^{(i)}\right)$$

- $r_i = p(z^{(i)})/q(z^{(i)})$ are known as importance weights.

- they correct the bias introduced by sampling from the wrong distribution.

- unlike rejection sampling, all of the samples generated are retained.

Properties

The importance sampling estimator:

$$\hat{I}_N(f) = \frac{1}{N}\sum_{i=1}^{N} f(z^{(i)})r_i$$

Is unbiased and converges almost surely to $I(f)$

The variance is:

$$\mathbb{V}\left[\hat{I}_N(f)\right] = \mathbb{E}[f^2(z)r^2(z)] - I(f)^2$$

$r_i = \frac{p(z^i)}{q(z^i)}$

Importance Sampling: Unnormalized Distributions

- Often the case that $\tilde{p}(z)$ can be evaluated easily, but not $p(z) = \tilde{p}(z)/Z_p$, where Z_p is unknown.

- Let us define the proposal distribution in similar form, i.e. $q(z) = \tilde{q}(z)/Z_q$.

$\mathbb{E}[f] \approx \frac{1}{N}\sum_i^N \frac{p(z^{(i)})}{q(z^{(i)})} f(z^{(i)}) = \frac{1}{N}\sum \frac{\tilde{p}}{Z_p}\frac{Z_q}{\tilde{q}} f$

- We then have:

$$\mathbb{E}[f] \approx \frac{Z_q}{Z_p}\frac{1}{N}\sum_i^N \frac{\tilde{p}\left(z^{(i)}\right)}{\tilde{q}\left(z^{(i)}\right)} f\left(z^{(i)}\right) = \frac{Z_q}{Z_p}\frac{1}{N}\sum_i^N \widetilde{r}_i f\left(z^{(i)}\right)$$

- Where $\widetilde{r}_i = \frac{\tilde{p}\left(z^{(i)}\right)}{\tilde{q}\left(z^{(i)}\right)}$

Importance Sampling: Unnormalized Distributions

- Use the same sample set to evaluate the ratio Z_p/Z_q

$$\frac{Z_p}{Z_q} = \frac{1}{Z_q}\int \tilde{p}(z)dz = \int \frac{\tilde{p}(z)}{\tilde{q}(z)} q(z)dz \approx \frac{1}{N}\sum_i^N \tilde{r}_i$$

$$\frac{1}{Z_q}\int \tilde{p}(z)\,dz$$
$$= \frac{1}{Z_q}\int \frac{\tilde{p}(z)}{q(z)} q(z)\,dz$$
$$= \frac{1}{\cancel{Z_q}}\int \frac{\tilde{p}(z)}{\frac{\tilde{q}(z)}{\cancel{Z_q}}} q(z)\,dz$$

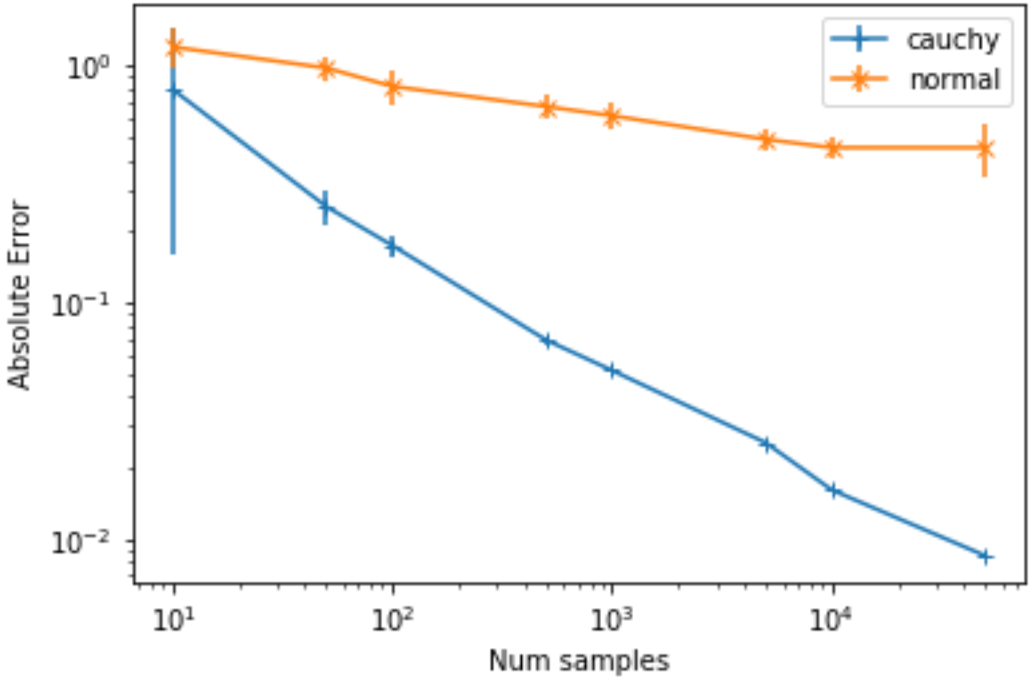
- So,

$$\mathbb{E}[f] \approx \frac{Z_q}{Z_p}\frac{1}{N}\sum_i^N \frac{\tilde{p}\left(z^{(i)}\right)}{\tilde{q}\left(z^{(i)}\right)} f\left(z^{(i)}\right) \approx \left(\frac{1}{\cancel{N}}\sum_j^N \tilde{r}_j\right)^{-1} \frac{1}{\cancel{N}}\sum_i^N \tilde{r}_i f\left(z^{(i)}\right)$$

$$\mathbb{E}[f] \approx \sum_i^N \frac{\tilde{r}_i}{\sum_j \tilde{r}_j} f\left(z^{(i)}\right) = \sum_i^N w_i f\left(z^{(i)}\right)$$

where $w_i = \frac{\tilde{r}_i}{\sum_j \tilde{r}_j} = \frac{\tilde{p}\left(z^{(i)}\right)/\tilde{q}\left(z^{(i)}\right)}{\sum_m \tilde{p}\left(z^{(m)}\right)/\tilde{q}\left(z^{(m)}\right)}$ Importance weight which is easy to compute!

Tutorial Sheet



Choosing a good q

- How to choose a good q?

Choosing a good q

- $q(x)$ should be proportional to $|f(x)|p(x)$
- $q(x) > 0$ whenever $p(x) \neq 0$
- Should be easy to sample from
- Easy to compute density $q(x)$

- Success depends on how “good” $q(z)$ is.
- **Further reading:** Adaptive importance sampling

Markov Chain Monte Carlo (MCMC) Sampling

Intuition and Algorithm

Key Ideas: MCMC

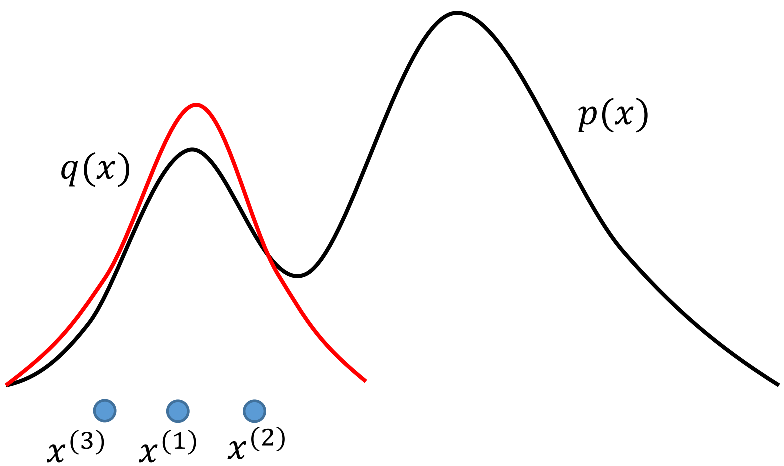
$x_1 \rightarrow x_2 \rightarrow x_1 \rightarrow \cdots$

- Generate samples $x^{(i)}$ while exploring the state space $\mathcal{X}$ using a Markov chain.
- The chain is constructed to spend more time in the most important regions.
- In particular, it is constructed so that the samples $x^{(i)}$ mimic samples drawn from the target distribution $p(x)$.

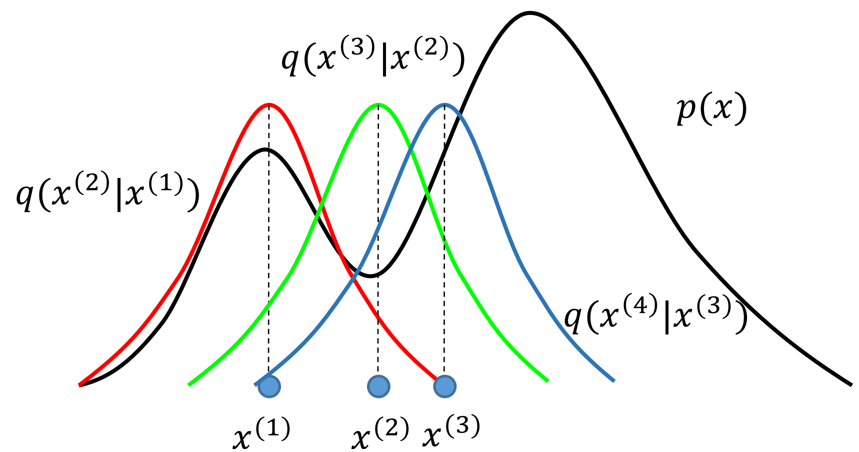
Markov Chain Monte Carlo (MCMC)

- MCMC algorithms feature adaptive proposals:
  - Instead of $q(x')$, we use $q(x'|x)$ where x' is the new state being sampled, and x is the current sample.
  - As x changes, $q(x'|x)$ can also change (as a function of x').

Importance sampling with bad proposal $q(x)$



MCMC with adaptive proposal $q(x'|x)$



MCMC: Metropolis-Hasting Algorithm

Algorithm : Metropolis-Hasting

1. Initialize $x^{(0)}$
2. For $i = 0$ to $N - 1$
3. Sample $u \sim \mathcal{U}_{[0,1]}$ // draw acceptance threshold
4. Sample $x' \sim q(x' | x^{(i)})$ // draw from proposal
5. If $u < \mathcal{A}\left(x', x^{(i)}\right) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}$ // acceptance probability
6. $x^{(i+1)} = x'$ // new sample is accepted
7. else
8. $x^{(i+1)} = x^{(i)}$ // new sample is rejected
   // we create a duplicate of the previous sample

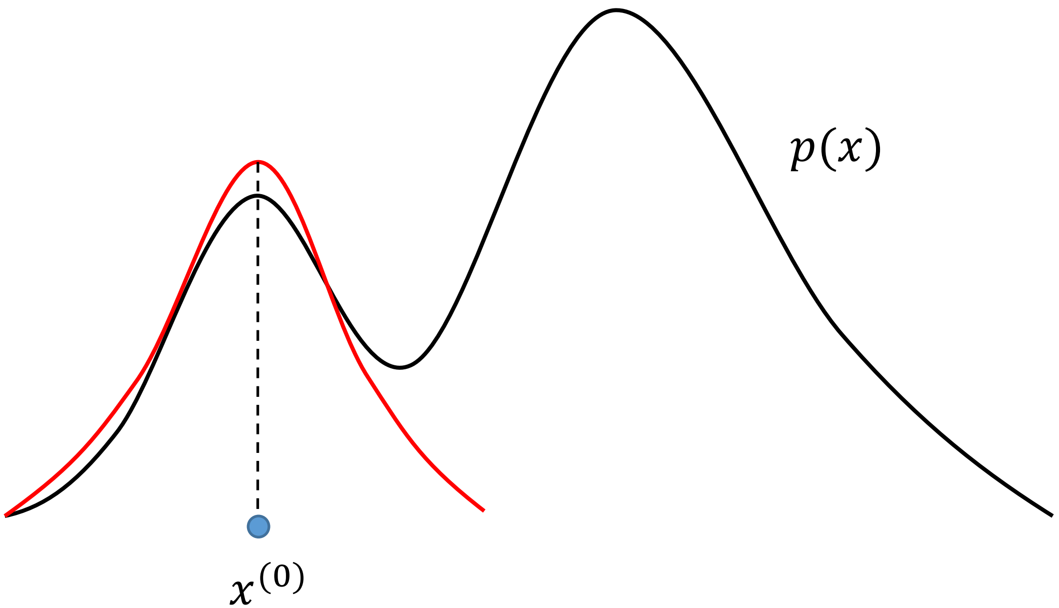
MCMC: Metropolis-Hasting Algorithm

Example:

- Our goal is to sample from a bimodal distribution $p(x)$.
- Let $q(x'|x)$ be a Gaussian centered on x.

Initialize $x^{(0)}$

$$\boxed{\mathcal{A}\left(x', x^{(i)}\right) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}}$$



MCMC: Metropolis-Hasting Algorithm

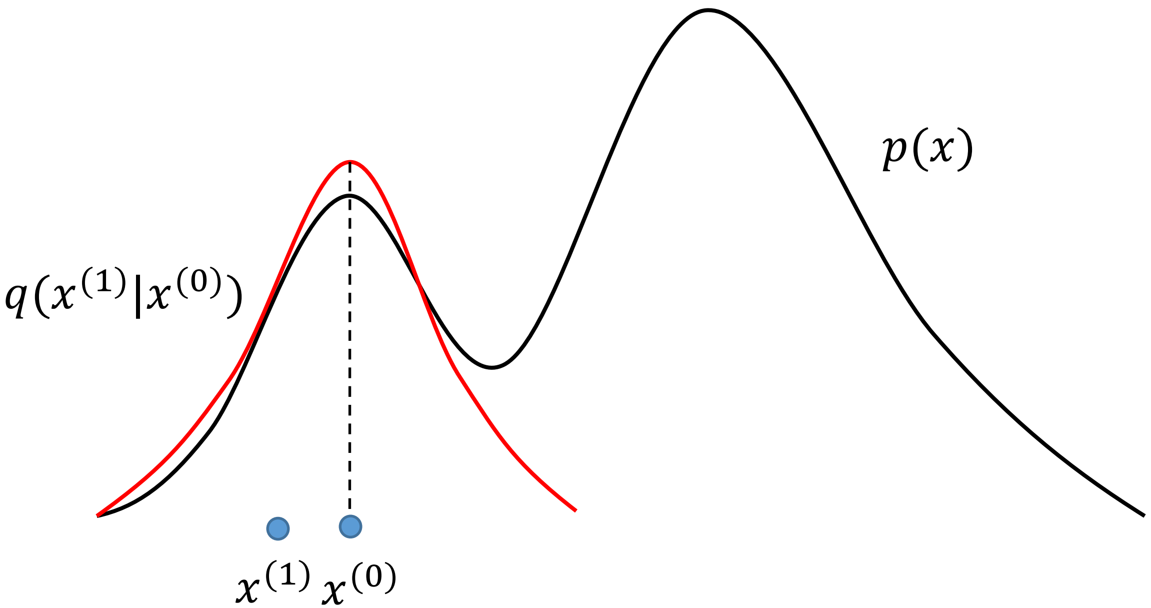
Example:

- Our goal is to sample from a bimodal distribution $p(x)$.
- Let $q(x'|x)$ be a Gaussian centered on x.

$$\boxed{\mathcal{A}\left(x', x^{(i)}\right) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}}$$

Initialize $x^{(0)}$

Draw, accept $x^{(1)}$



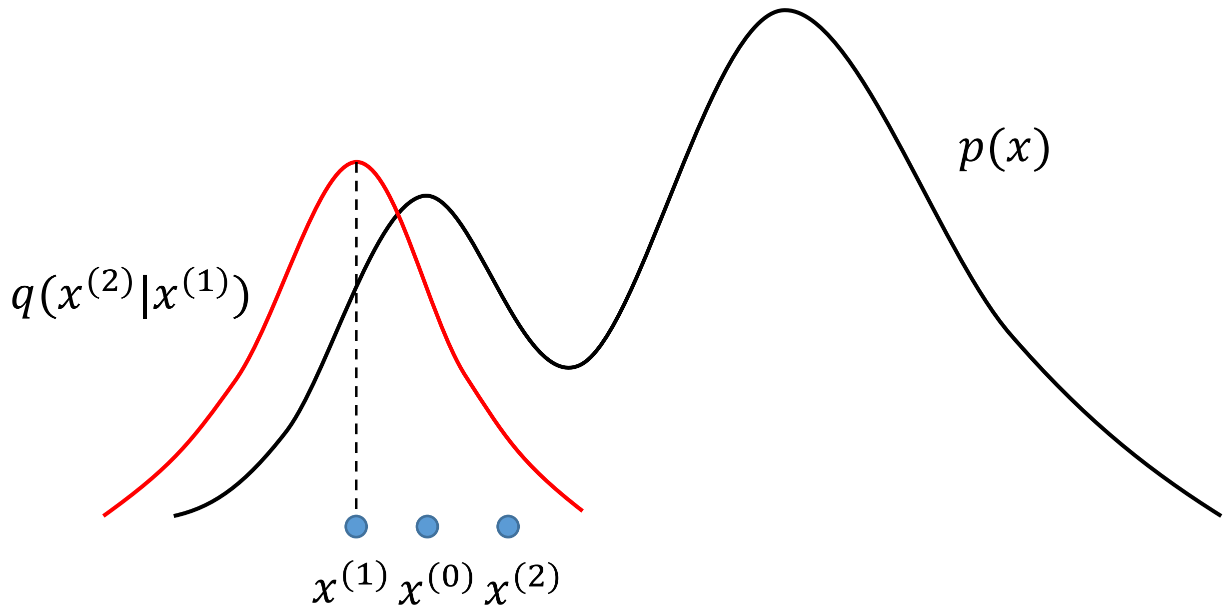
MCMC: Metropolis-Hasting Algorithm

Example:

- Our goal is to sample from a bimodal distribution $p(x)$.
- Let $q(x'|x)$ be a Gaussian centered on x.

$$\mathcal{A}(x', x^{(i)}) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}$$

Initialize $x^{(0)}$
Draw, accept $x^{(1)}$
Draw, accept $x^{(2)}$



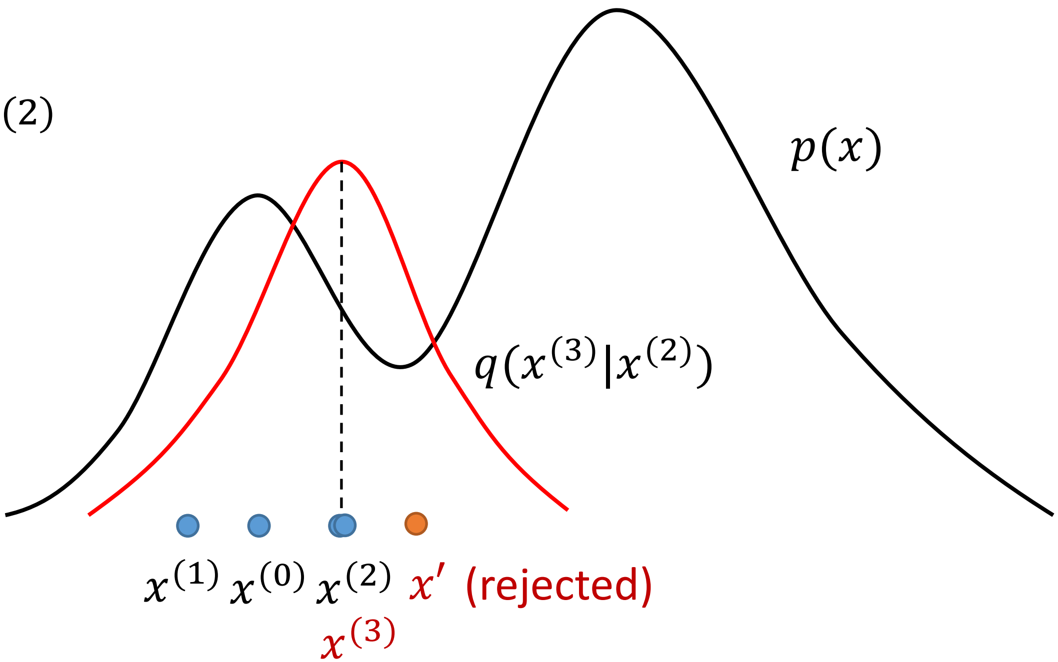
MCMC: Metropolis-Hasting Algorithm

Example:

- Our goal is to sample from a bimodal distribution $p(x)$.
- Let $q(x'|x)$ be a Gaussian centered on x.

$$\boxed{\mathcal{A}(x', x^{(i)}) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}}$$

Initialize $x^{(0)}$
Draw, accept $x^{(1)}$
Draw, accept $x^{(2)}$
Draw but reject; set $x^{(3)} = x^{(2)}$



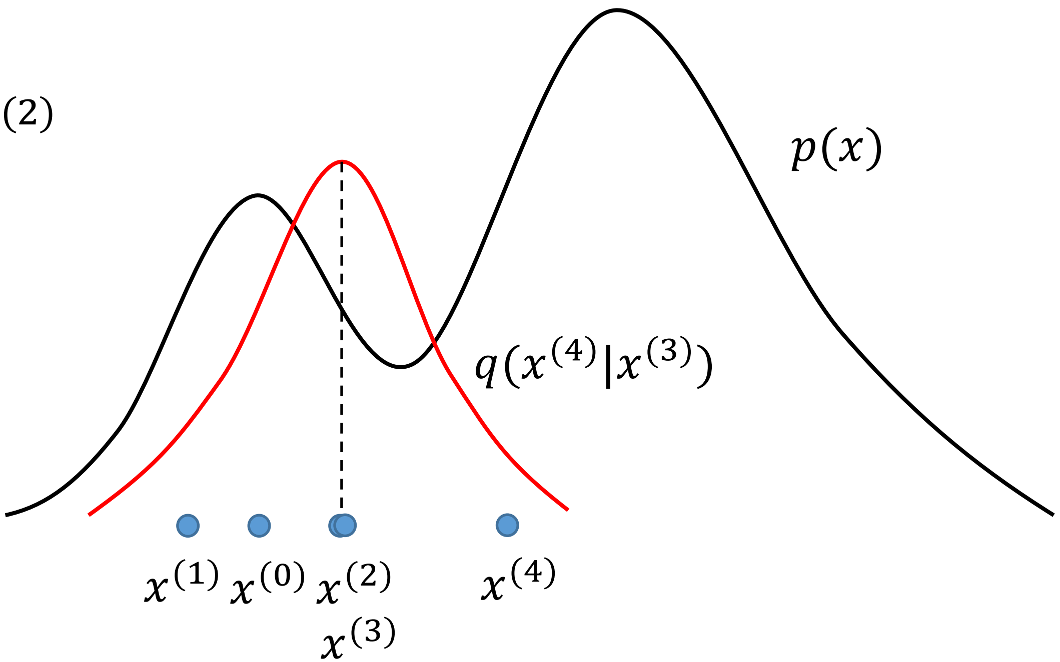
MCMC: Metropolis-Hasting Algorithm

Example:

- Our goal is to sample from a bimodal distribution $p(x)$.
- Let $q(x'|x)$ be a Gaussian centered on x.

$$\boxed{\mathcal{A}\left(x', x^{(i)}\right) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}}$$

Initialize $x^{(0)}$
Draw, accept $x^{(1)}$
Draw, accept $x^{(2)}$
Draw but reject; set $x^{(3)} = x^{(2)}$
Draw, accept $x^{(4)}$



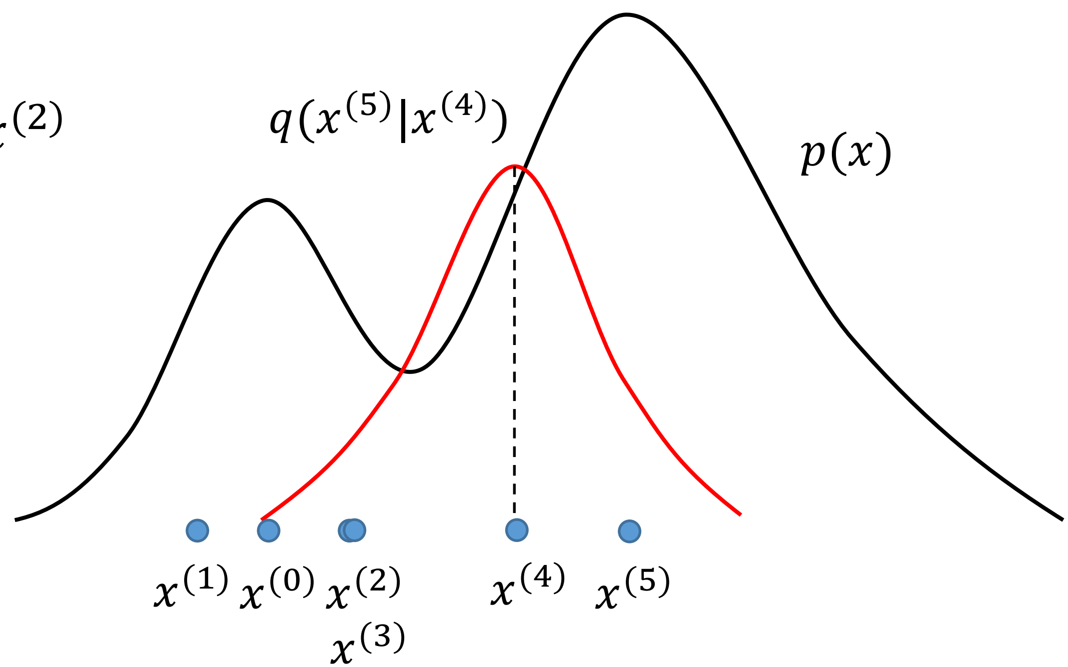
MCMC: Metropolis-Hasting Algorithm

Example:

- Our goal is to sample from a bimodal distribution $p(x)$.
- Let $q(x'|x)$ be a Gaussian centered on x.

$$\mathcal{A}(x', x^{(i)}) = \min\left\{1, \frac{\tilde{p}(x')q(x^{(i)}|x')}{\tilde{p}(x^{(i)})q(x'|x^{(i)})}\right\}$$

Initialize $x^{(0)}$
Draw, accept $x^{(1)}$
Draw, accept $x^{(2)}$
Draw but reject; set $x^{(3)} = x^{(2)}$
Draw, accept $x^{(4)}$
Draw, accept $x^{(5)}$



The adaptive proposal $q(x'|x^{(i)})$ allows us to sample both modes of $p(x)$!

Burn-In Period

- The initial samples may follow a very different distribution, especially if the starting point is in a region of low density.
- As a result, a burn-in period is typically necessary, where an initial number of samples (e.g. the first 1,000 or so) are thrown away.

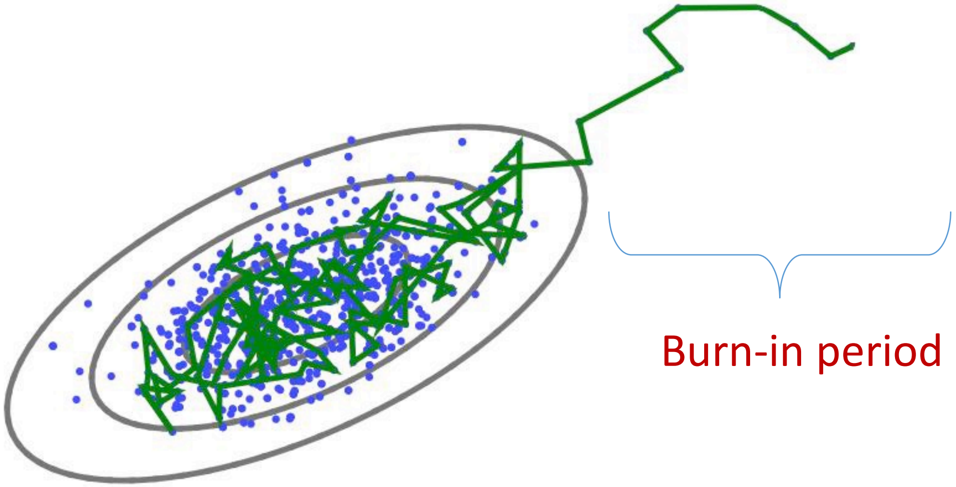
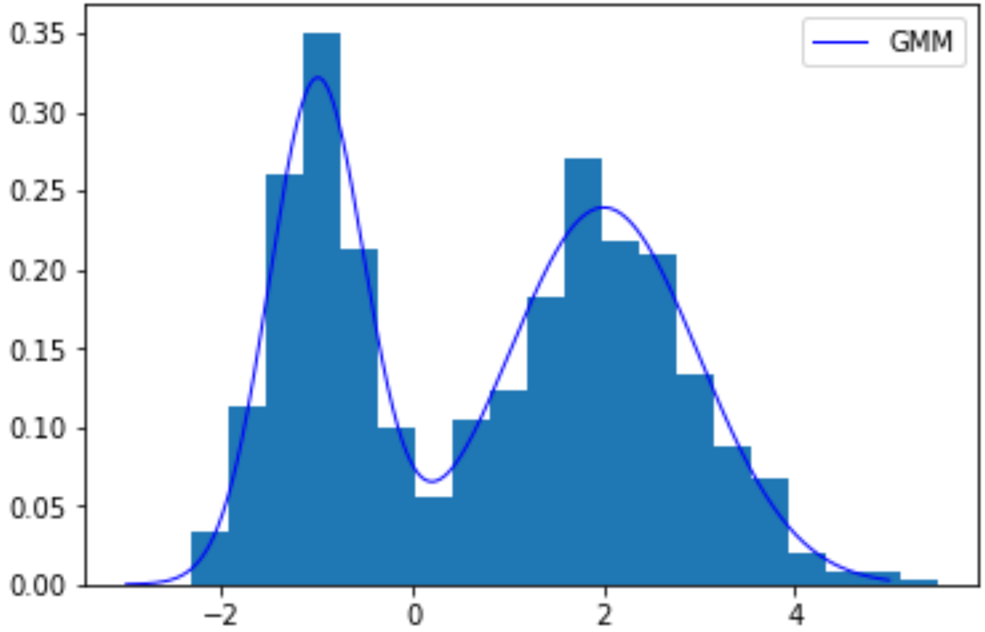


Image source: http://wiki.ubc.ca/Course:CPSC522/MCMC

Tutorial Sheet



NUS National University of Singapore | School of Computing

MCMC Theory

Markov Chains, Stationary Distributions, Ergodicity

What is the connection between Markov chains and MCMC?

Why does the Metropolis-Hasting algorithm work?

Ergodic Theorem for Markov Chains

If $X_0, X_1, \ldots, X_N$ is an irreducible, homogenous, aperiodic discrete Markov Chain with stationary distribution π, then:

$$\frac{1}{N}\sum_i^N f(X_i) \xrightarrow{a.s.} \mathbb{E}[f(X)] \text{ as } N \to \infty$$

where $X \sim \pi$, and

$$p(x_N = x | x_0) \to \pi(x) \;\forall x, x_0 \in \mathcal{X} \text{ as } N \to \infty$$

The key idea: Metropolis-Hastings constructs a irreducible, homogenous, aperiodic Markov Chain with stationary distribution $\tilde{p}$

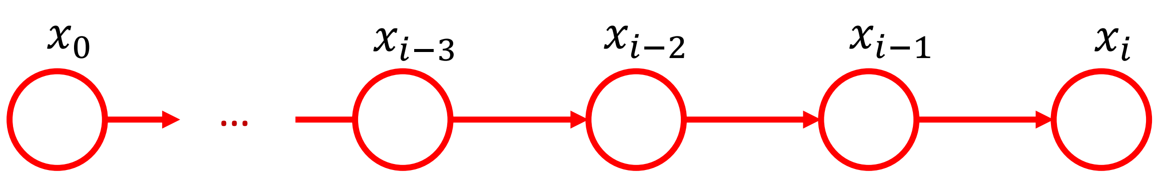
What is a Markov Chain?

- Intuitive to introduce Markov chains on finite state spaces, where $x^{(i)}$ can only take s discrete values $x^{(i)} \in \mathcal{X} = \{x_1, x_2, \dots, x_s\}$.

- The stochastic process $x^{(i)}$ is called a Markov chain if:

$$p\left(x^{(i)} \mid x^{(i-1)}, \dots, x^{(1)}\right) = T\left(x^{(i)} \mid x^{(i-1)}\right)$$

$s \times s$ matrix



- Current state $x^{(i)}$ is conditionally independent of all previous states given most recent state $x^{(i-1)}$.

Properties of a Markov Chain

1. **Homogeneous chain**:

- Chain is homogeneous if $T \triangleq T\left(x^{(i)} \mid x^{(i-1)}\right)$ remains invariant $\forall\, i$, with $\underbrace{\sum_{x^{(i)}} T\left(x^{(i)} \middle| x^{(i-1)}\right) = 1}_{\text{Sum of each row in } T \text{ equals to 1}}$ for any i.

- That is, the evolution of the chain in a space $\mathcal{X}$ depends solely on the current state of the chain and a fixed transition matrix.

Properties of a Markov Chain

2. Stationary and limiting distributions:

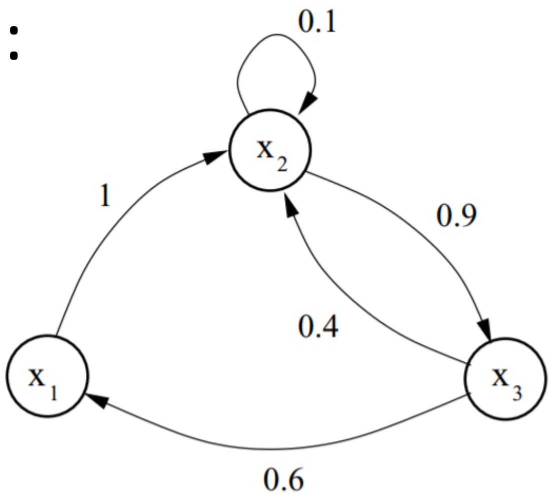
- A probability vector $\pi = p(x)$ defined on $\mathcal{X}$ is a stationary (invariant) distribution (w.r.t T) if

$$\pi T = \pi.$$

- A limiting distribution π, is a distribution over the states such that whatever the starting distribution π_0, the Markov chain converges to π.

Properties of a Markov Chain

Example:



Transition graph for the Markov chain example with $\mathcal{X} = \{x_1, x_2, x_3\}$.

$$T = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0.1 & 0.9 \\ 0.6 & 0.4 & 0 \end{bmatrix}$$

If initial state is $\mu\left(x^{(1)}\right) = (0.5, 0.2, 0.3)$ (can be any state), it follows that:

$$\mu\left(x^{(2)}\right) = \mu\left(x^{(1)}\right)T = (0.18, 0.64, 0.18)$$

$$\vdots$$

$$\mu\left(x^{(t)}\right) = \mu\left(x^{(t-1)}\right)T = (0.2213, 0.4098, 0.3689)$$

$$\mu\left(x^{(t+1)}\right) = \mu\left(x^{(t+2)}\right)T = (0.2213, 0.4098, 0.3689)$$

Converges to stationary distribution!

Image source: "An introduction to MCMC for Machine Learning", Christophe Andrieu at. al.

Properties of a Markov Chain

3. Irreducibility:

- A Markov chain is irreducible if for any state of the Markov chain, there is a positive probability of visiting all other states, i.e.

$$\forall a, b \in \mathcal{X}, \qquad \exists t \geq 0$$

$$s.t. \quad p(x_t = b \mid x_0 = a) > 0$$

Properties of a Markov Chain

4. Aperiodicity:

- The Markov chain should not get trapped in cycles, i.e.

$$\gcd\{t : p(x_t = a \mid x_0 = a) > 0\} = 1, \qquad \forall\, a \in \mathcal{X}$$

greatest common divisor

0 1 2 3 4 8 12 16 t

$p(x_t=a)$ 0 0.1 0.2 0 0 — 0 — 0.3 —

$p(x_t=a)$ 0 — 0 0.1 0 — 0 0.4 ~~0~~ 0.6

Ergodic Theorem for Markov Chains

- A Markov chain is ergodic if it is irreducible and aperiodic.

- **Ergodicity is important**: it implies we can reach the stationary/limiting distribution π, no matter the initial distribution π_0.

- All good MCMC algorithms must satisfy ergodicity, so that we cannot initialize in a way that will never converge.

Detailed Balance (Reversibility)

- A probability vector $\pi = p(x)$ defined on $\mathcal{X}$ satisfies detailed balance w.r.t T if:

$$\pi_a T_{ab} = \pi_b T_{ba}, \qquad \forall a, b \in x$$

Remark 1: Detailed balance $\Longrightarrow$ stationary distribution, i.e. $\pi T = \pi$.

Proof:

$$\begin{aligned} \pi_b &= \sum_a \pi_a T_{ab} & \\ &= \sum_a \pi_b T_{ba} & \text{(detailed balance)} \\ &= \pi_b \underbrace{\boxed{\sum_a T_{ba}}}_{=1} & \text{(sum over row of } T_{ba}\text{)} \\ &= \pi_b, \qquad \forall b \in x & \text{(stationary distribution)} \end{aligned}$$

Detailed Balance (Reversibility)

- A probability vector $\pi = p(x)$ defined on $\mathcal{X}$ satisfies detailed balance w.r.t T if:

$$\pi_a T_{ab} = \pi_b T_{ba}, \qquad \forall a, b \in x$$

Remark 2: Detailed balance = "reversibility"

- terminology: we say that a Markov chain is "reversible" if it has a stationary distribution π that satisfies detailed balance w.r.t T.

Ergodic Theorem for Markov Chains

If $X_0, X_1, \dots, X_N$ is an irreducible, homogenous, aperiodic discrete Markov Chain with stationary distribution π, then:

$$\frac{1}{N}\sum_i^N f(X_i) \xrightarrow{a.s.} \mathbb{E}[f(X)] \text{ as } N \to \infty$$

where $X \sim \pi$, and

$$p(x_N = x | x_0) \to \pi(x)\ \forall x, x_0 \in \mathcal{X} \text{ as } N \to \infty$$

The key idea: Metropolis-Hastings constructs a irreducible, homogenous, aperiodic Markov Chain with stationary distribution $\tilde{p}$

Why does Metropolis-Hastings work?

- Recall that we draw a sample x' according to $q(x'|x)$, and then accept/reject according to $\mathcal{A}(x', x)$.

- In other words, the transition kernel is:

$$T(x' \mid x) = q(x' \mid x)\mathcal{A}(x' \mid x)$$

- We shall show that $\tilde{p}$ satisfies detailed balance wrt T

Why does Metropolis-Hastings work?

- Recall that:

$$\mathcal{A}(x', x) = \min\left\{1, \frac{\tilde{p}(x')q(x|x')}{\tilde{p}(x)q(x'|x)}\right\}$$

- Notice this implies the following:

if $\mathcal{A}(x', x) < 1$, then $\dfrac{\tilde{p}(x)q(x'|x)}{\tilde{p}(x')q(x|x')} > 1$

and thus $\mathcal{A}(x, x') = 1$

Why does Metropolis-Hastings work?

- Now suppose $\mathcal{A}(x', x) < 1$ and $\mathcal{A}(x, x') = 1$, we have:

$$\mathcal{A}(x', x) = \frac{\tilde{p}(x')q(x|x')}{\tilde{p}(x)q(x'|x)}$$

$$\mathcal{A}(x', x)\tilde{p}(x)q(x'|x) = \tilde{p}(x')q(x|x')$$

$$\mathcal{A}(x', x)\tilde{p}(x)q(x'|x) = \mathcal{A}(x, x')\tilde{p}(x')q(x|x')$$

$$\tilde{p}(x)T(\, x' \mid x\,) = \tilde{p}(x')T(x \mid x')$$

This is the detailed balance condition!

Why does Metropolis-Hastings work?

$$\tilde{p}(x)T(\, x' \mid x\,) = \tilde{p}(x')T(x \mid x')$$

- In other words, the Metropolis-Hasting algorithm has stationary distribution $\tilde{p}(x)$.
  - Recall we defined $\tilde{p}(x)$ to be the (un-normalized) true distribution of x.
- Have to show ergodicity, i.e., irreducibility and aperiodicity to show convergence.
  - Irreducible if the support of q includes the support of $\tilde{p}$
  - Aperiodic since there is always a chance of rejection

Ergodic Theorem for Markov Chains

If $X_0, X_1, \ldots, X_N$ is an irreducible, homogenous, aperiodic discrete Markov Chain with stationary distribution π, then:

$$\frac{1}{N}\sum_i^N f(X_i) \xrightarrow{a.s.} \mathbb{E}[f(X)] \text{ as } N \to \infty$$

where $X \sim \pi$, and

$$p(x_N = x | x_0) \to \pi(x)\ \forall x, x_0 \in \mathcal{X} \text{ as } N \to \infty$$

The key idea: Metropolis-Hastings constructs a irreducible, homogenous, aperiodic Markov Chain with stationary distribution $\tilde{p}$

NUS National University of Singapore | School of Computing

Metropolis & Gibbs Sampling

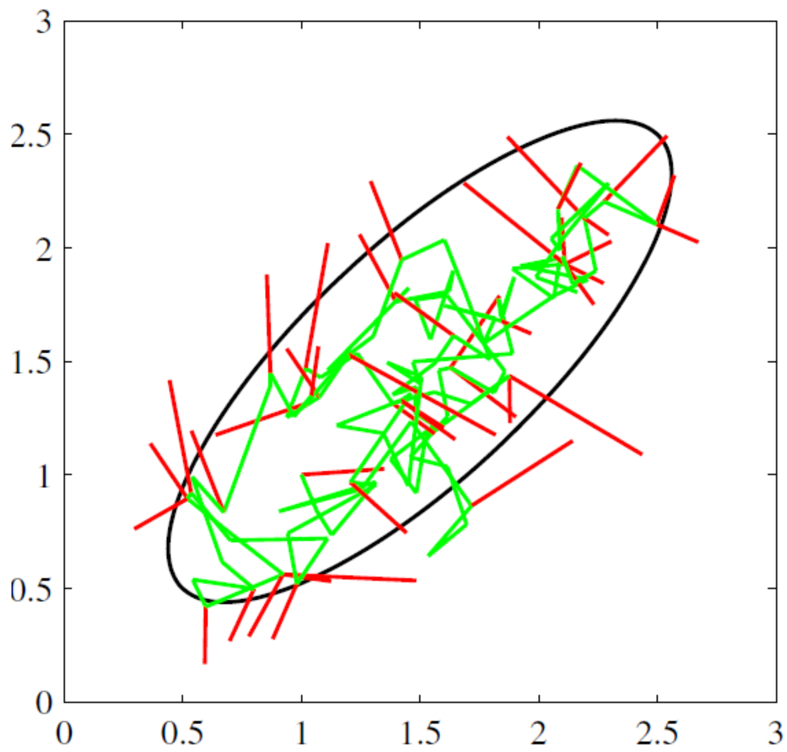
Metropolis Algorithm

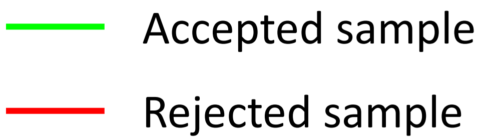
- Metropolis algorithm is a special case of the Metropolis-Hasting algorithm.
- Proposal distribution is a random walk, i.e. $q(x|x')$= $q(x'|x)$, e.g. an isotropic Gaussian distribution.
- Acceptance probability of Metropolis algorithm is given by:

$$\mathcal{A}(x',x) = \min\left\{1, \frac{\tilde{p}(x')\cancel{q(x|x')}}{\tilde{p}(x)\cancel{q(x'|x)}}\right\} = \min\left\{1, \frac{\tilde{p}(x')}{\tilde{p}(x)}\right\}$$

Metropolis Algorithm

- Illustration of using Metropolis algorithm (proposal distribution: isotropic Gaussian) to sample from a Gaussian distribution:

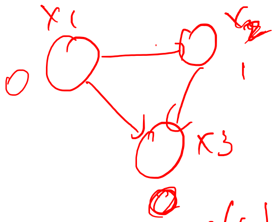




150 candidate samples are generated, 43 are rejected.

Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Gibbs Sampling



$\tilde{x}_1 = 1 \sim p(x_1 | x_2 = 1, x_3 = 0)$

$\tilde{x}_2 \sim p(x_2 | x_1 = 1, x_3 = 0)$

$= 1$

$\tilde{x}_3 \sim p(x_3 | x_1 = 1, x_2 = 1)$

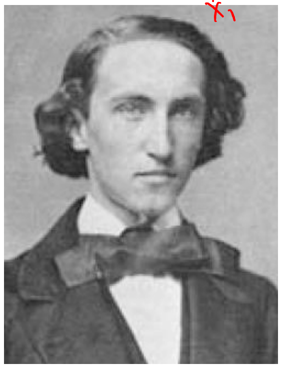
$= 1$

$\tilde{x}_1 \sim p(x_1 | x_2 = 1, x_3 = 1)$

$\vdots$

Algorithm : Gibbs Sampling

1. Initialize $\{x_i : i = 1, \ldots, M\}$
2. For $\tau = 1, \ldots, T$:
3. Sample $x_1^{\tau+1} \sim p(x_1 | x_2^{(\tau)}, x_3^{(\tau)}, \ldots, x_M^{(\tau)})$.
4. Sample $x_2^{\tau+1} \sim p(x_2 | x_1^{(\tau+1)}, x_3^{(\tau)}, \ldots, x_M^{(\tau)})$.
5. Sample $x_j^{\tau+1} \sim p\left(x_j \middle| x_1^{(\tau+1)}, \ldots, x_{j-1}^{(\tau+1)}, x_{j+1}^{(\tau)}, \ldots, x_M^{(\tau)}\right)$.
6. Sample $x_M^{\tau+1} \sim p(x_M | x_1^{(\tau+1)}, x_2^{(\tau+1)}, \ldots, x_{M-1}^{(\tau+1)})$



Josiah Willard Gibbs
1839–1903

Image Source: “Pattern Recognition and Machine Learning”, Christopher Bishop

Gibbs Sampling

- We have the expressions for the full conditionals:

$$p(\, x_j \,|x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_M).$$

- Let $q_j(x'|x) = p\left(x_j'\middle|x_{\backslash \mathrm{j}}\right)$
  - $x_{\backslash \mathrm{j}}$ denotes all variables except x_j
  - Note: $x'_{\backslash \mathrm{j}} = x_{\backslash \mathrm{j}}$ since those values don't change

Gibbs Sampling

- Gibbs sampling is a special case of the Metropolis-Hasting algorithm where the acceptance probability is always one.

Why:

$$\mathcal{A}(x', x) = \min\left\{1, \frac{p(x')q_j(x|x')}{p(x)q_j(x'|x)}\right\}$$

$$= \min\left\{1, \frac{\cancel{p(x_j'|x_{\setminus j}')}\,\cancel{p(x_{\setminus j}')}\,\cancel{p(x_j \mid x_{\setminus j}')}}{\cancel{p(x_j|x_{\setminus j})}\,\cancel{p(x_{\setminus j})}\,\cancel{p(x_j' \mid x_{\setminus j})}}\right\}$$

Note that $x'_{\setminus j} = x_{\setminus j}$ because these components are kept fixed during the sampling step:

$$\Longrightarrow \mathcal{A}(x', x) = \min\left\{1, \cancelto{1}{\frac{p(x_j'|x_{\setminus j}')p(x_{\setminus j})p(x_j \mid x_{\setminus j})}{p(x_j|x_{\setminus j})p(x_{\setminus j})p(x_j' \mid x_{\setminus j}')}}\right\} = 1$$

Gibbs Sampling: Markov Blankets

- The conditional $p(\, x_j \,|x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_N)$ looks intimidating, but recall Markov Blankets:

$$p(\, x_j \,|x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_N) = p(\, x_j \mid \underbrace{\mathrm{MB}(x_j)}_{\text{Markov blanket of } x_j} \,)\,.$$

- Bayesian network: the Markov blanket of X_j is the set containing its parents, children, and co-parents.

- MRF: the Markov Blanket of X_j is its immediate neighbors.

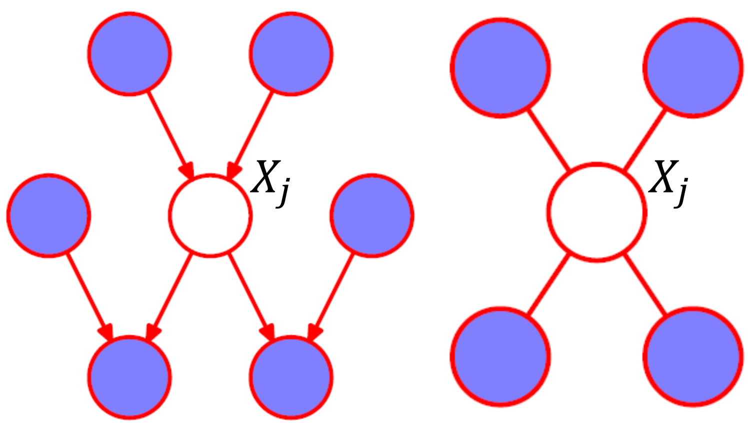


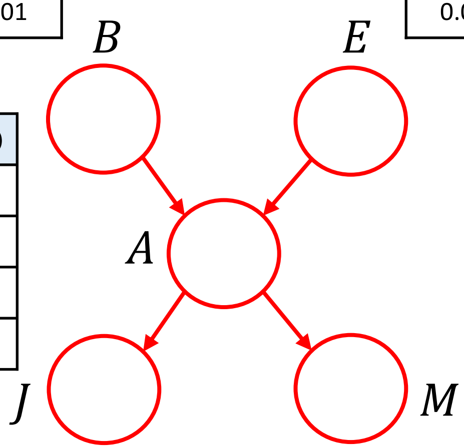
Image Source: "Pattern Recognition and Machine Learning", Christopher Bishop

Gibbs Sampling: Example on a Bayesian Network

B: Burglary, E: Earthquake, A: Alarm, J: John Calls, M: Mary Calls

$p(B)$
0.001

$p(E)$
0.002



B	E	$p(A \mid B, E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01

A	$p(J \mid A)$
T	0.90
F	0.05

A	$p(M \mid A)$
T	0.70
F	0.01

- Assume we sample variables in the order B, E, A, J, M.
- Initialize all variables at $t = 0$ to False.

t	B	E	A	J	M
0	F	F	F	F	F
1					
2					
3					
4					

Gibbs Sampling: Example on a Bayesian Network

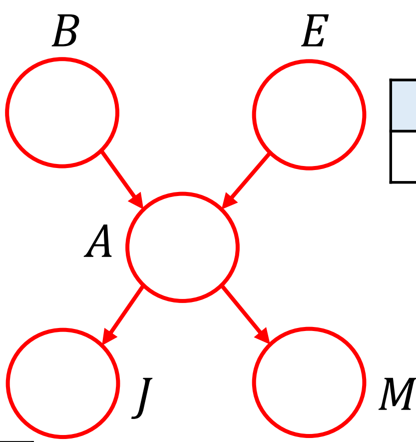
- Sampling $p(B|\, A, E)$ at $t = 1$: using Bayes rule, we have

$$p(\,B \mid A, E\,) \propto p(\,A \mid B, E\,)\, p(\,B\,)$$

- $(A, E) = (F, F)$, we compute the following, and sample $B = F$

$$p(\,B = T\ |A = F, E = F\,) \propto (0.06)(0.001) = 0.00006$$
$$p(\,B = F\ |A = F, E = F\,) \propto (0.99)(0.999) = 0.98901$$

$p(B)$
0.001



$p(E)$
0.002

B	E	$p(A \mid B, E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01

A	$p(J\ \|A)$
T	0.90
F	0.05

A	$p(M\ \|A)$
T	0.70
F	0.01

t	B	E	A	J	M
0	F	F	F	F	F
1	F				
2					
3					
4					

Gibbs Sampling: Example on a Bayesian Network

- Sampling $p(E|A,B)$ at $t = 1$: using Bayes rule, we have

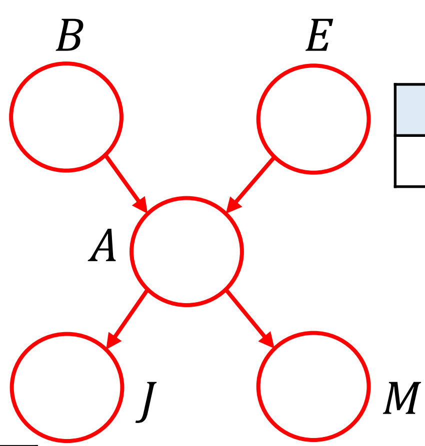
$$p(\,E \mid A, B\,) \propto p(\,A \mid B, E\,)\, p(\,E\,)$$

- $(A, B) = (F, F)$, we compute the following, and sample $E = T$

$$p(\,E = T \mid A = F, B = F\,) \propto (0.71)(0.002) = 0.00142$$
$$p(\,E = F \mid A = F, B = F\,) \propto (0.99)(0.998) = 0.98802$$

$p(B)$
0.001

$p(E)$
0.002



B	E	$p(A \mid B, E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01

A	$p(J \mid A)$
T	0.90
F	0.05

A	$p(M \mid A)$
T	0.70
F	0.01

t	B	E	A	J	M
0	F	F	F	F	F
1	F	T			
2					
3					
4					

Gibbs Sampling: Example on a Bayesian Network

- Sampling $p(A|B,E,J,M)$ at $t = 1$: using Bayes rule

$$p(A \mid B,E,J,M) \propto p(J|A)p(M|A)p(A|B,E)$$

- $(B,E,J,M) = (F,T,F,F)$, we compute the following, and sample $A = F$

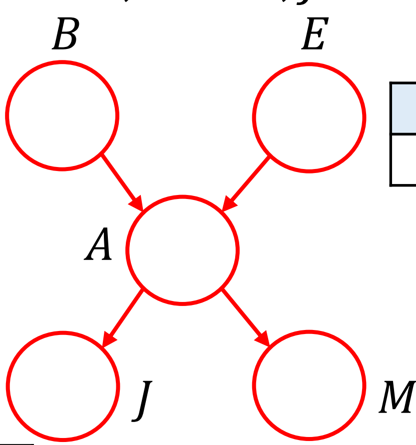
$$p(A=T|B=F,E=T,J=F,M=F) \propto (0.1)(0.3)(0.29) = 0.0087$$

$$p(A=F|B=F,E=T,J=F,M=F) \propto (0.95)(0.99)(0.71) = 0.6678$$

$p(B)$
0.001

$p(E)$
0.002

B	E	$p(A \mid B,E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01



A	$p(J \mid A)$
T	0.90
F	0.05

A	$p(M \mid A)$
T	0.70
F	0.01

t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F		
2					
3					
4					

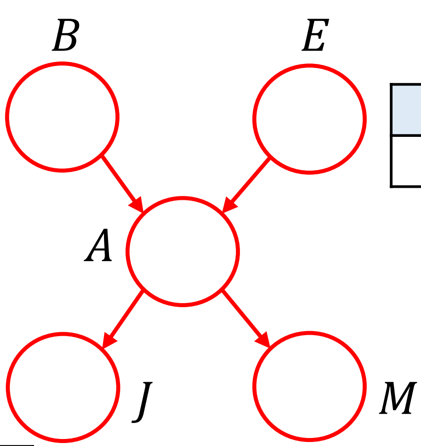
Gibbs Sampling: Example on a Bayesian Network

- Sampling $p(J|A)$ at $t = 1$: no need to apply Bayes rule
- $A = F$, we compute the following, and sample $J = T$

$$p(J = T|A = F) \propto 0.05$$
$$p(J = F|A = F) \propto 0.95$$

$p(B)$
0.001

B	E	$p(A \mid B, E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01



$p(E)$
0.002

A	$p(J \mid A)$
T	0.90
F	0.05

A	$p(M \mid A)$
T	0.70
F	0.01

t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	
2					
3					
4					

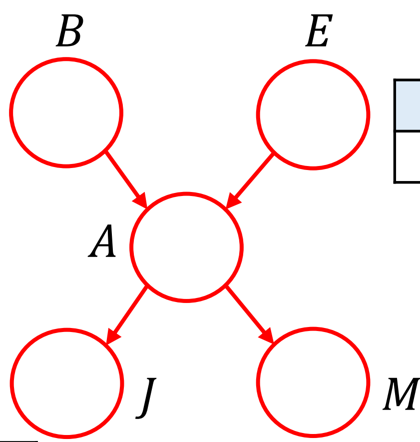
Gibbs Sampling: Example on a Bayesian Network

- Sampling $p(M|A)$ at $t = 1$: no need to apply Bayes rule
- $A = F$, we compute the following, and sample $M = F$

$$p(M = T|A = F) \propto 0.01$$
$$p(M = F|A = F) \propto 0.99$$

$p(B)$
0.001

B	E	$p(A \mid B, E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01



$p(E)$
0.002

A	$p(J \mid A)$
T	0.90
F	0.05

A	$p(M \mid A)$
T	0.70
F	0.01

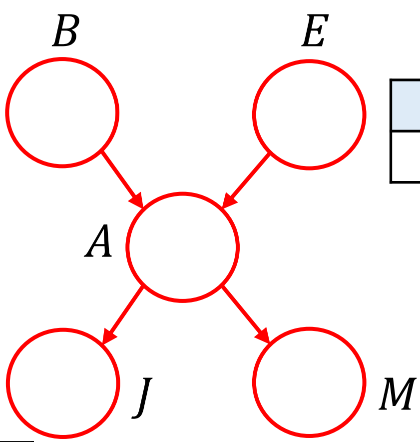
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2					
3					
4					

Gibbs Sampling: Example on a Bayesian Network

- Now $t = 2$, and we repeat the procedure to sample new values of B, E, A, J, M …
- And similarly for $t = 3, 4$, etc.

$p(B)$
0.001

$p(E)$
0.002



B	E	$p(A \mid B, E)$
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01

A	$p(J \mid A)$
T	0.90
F	0.05

A	$p(M \mid A)$
T	0.70
F	0.01

t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3	T	F	T	F	T
4	T	F	T	F	F

Appendix

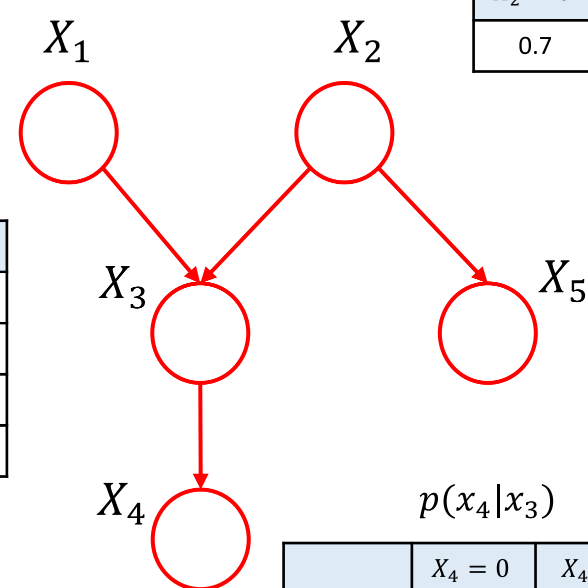
Importance Sampling: Example on a Bayesian Network

$p(x_1)$

$X_1 = 0$	$X_1 = 1$
0.6	0.4

$p(x_2)$

$X_2 = 0$	$X_2 = 1$
0.7	0.3



$p(x_3|x_1,x_2)$

	$X_3 = 0$	$X_3 = 1$	$X_3 = 2$
$X_1 = 0, X_2 = 0$	0.3	0.4	0.3
$X_1 = 0, X_2 = 1$	0.05	0.25	0.7
$X_1 = 1, X_2 = 0$	0.9	0.08	0.02
$X_1 = 1, X_2 = 1$	0.5	0.3	0.2

$p(x_5|x_2)$

	$X_5 = 0$	$X_5 = 1$
$X_2 = 0$	0.95	0.05
$X_2 = 1$	0.2	0.8

$p(x_4|x_3)$

	$X_4 = 0$	$X_4 = 1$
$X_3 = 0$	0.1	0.9
$X_3 = 1$	0.4	0.6
$X_3 = 2$	0.99	0.01

How do we compute

$p(x_1, x_4, x_5 \mid x_2 = 1, x_3 = 1)$?

X_1: Difficulty, X_2: Intelligence, X_3: Grade, X_4: Letter, X_5: SAT score

Importance Sampling: Example on a Bayesian Network

- How do we compute $p(x_1, x_4, x_5 \,|x_2 = 1, x_3 = 1)$?

Importance Sampling!!!

$$p(x_1, x_4, x_5 \,|x_2 = 1, x_3 = 1) = \frac{p(x_1, x_4, x_5, x_2 = 1, x_3 = 1)}{p(x_2 = 1, x_3 = 1)}$$

$$= \frac{p(x_F, x_E)}{p(x_E)}$$

$$= \frac{1}{Z_p}\tilde{p}(x)$$

We don't want to evaluate Z_p

Target distribution

- What should we use as the proposal distribution $q(x)$?

Importance Sampling: Example on a Bayesian Network

$p(x_1)$

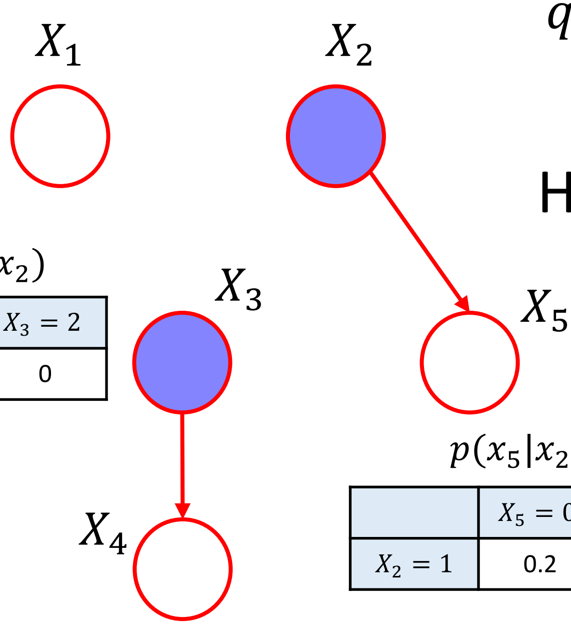
$X_1 = 0$	$X_1 = 1$
0.6	0.4

$p(x_2 = 1)$

$X_2 = 0$	$X_2 = 1$
0	1

$p(x_3 = 1|x_1, x_2)$

$X_3 = 0$	$X_3 = 1$	$X_3 = 2$
0	1	0



$p(x_5|x_2 = 1)$

	$X_5 = 0$	$X_5 = 1$
$X_2 = 1$	0.2	0.8

$p(x_4|x_3 = 1)$

	$X_4 = 0$	$X_4 = 1$
$X_3 = 1$	0.4	0.6

Proposal distribution:

$$q(x_1, x_4, x_5) = p(x_1)p(x_4|x_3 = 1)p(x_5|x_2 = 1)$$

How do we sample from $q(x_1, x_4, x_5)$?

$$x_1 \sim p(x_1)$$

$$x_4 \sim p(x_4|x_3 = 1)$$

$$x_5 \sim p(x_5|x_2 = 1)$$

e.g. randomly generate a number within [0,1] (uniform distribution), i.e. $n = \text{rand}$; $x_1 = 0$ if $n < 0.6$, $x_1 = 1$ otherwise.

Importance Sampling: Example on a Bayesian Network

- For each sample $x^{(l)}$, we evaluate the weight as:

$$w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\tilde{p}(x^{(l)})/q(x^{(l)})}{\sum_m \tilde{p}(x^{(m)})/q(x^{(m)})}.$$

- Example:

$x^{(l)}: \{x_1 = 0, x_4 = 1, x_5 = 1\}$ obtained from sampling, we have

$$\tilde{p}(x^{(l)}) = p(x_1 = 0, x_4 = 1, x_5 = 1, x_2 = 1, x_3 = 1)$$

$$= p(x_1 = 0)p(x_2 = 1)p(x_3 = 1 | x_2 = 1, x_1 = 0)$$

$$p(x_4 = 1 \mid x_3 = 1)p(x_5 = 1 \mid x_2 = 1)$$

$$= (0.6)(0.3)(0.08)(0.6)(0.8)$$

$$= 0.006912$$

Importance Sampling: Example on a Bayesian Network

- For each sample $x^{(l)}$, we evaluate the weight as:

$$w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\tilde{p}(x^{(l)})/q(x^{(l)})}{\sum_m \tilde{p}(x^{(m)})/q(x^{(m)})}.$$

- Example:

  $x^{(l)}: \{x_1 = 0, x_4 = 1, x_5 = 1\}$ obtained from sampling, we have

$$q(x^{(l)}) = p(x_1 = 0)p(x_4 = 1|x_3 = 1)p(x_5 = 1|x_2 = 1)$$

$$= (0.6)(0.6)(0.8) = 0.288$$

$$\Rightarrow \frac{\tilde{p}(x^{(l)})}{q(x^{(l)})} = \frac{0.006912}{0.288} = 0.024$$

- Finally, denominator (hence each weight w_l) can be computed from all M samples.

Importance Sampling: Example on a Bayesian Network

- We can compute $p(x_1, x_4, x_5 \mid x_2 = 1, x_3 = 1)$ from all the weights and samples:

Sum of all weights from samples at $\{x_1 = 0, x_4 = 0, x_5 = 0\}$

$$p(x_1 = 0, x_4 = 0, x_5 = 0 \mid x_2 = 1, x_3 = 1) = \frac{\boxed{\sum_m w_m \delta(x^{(m)} = \{x_1 = 0, x_4 = 0, x_5 = 0\})}}{\boxed{\sum_m w_m}}$$

normalizer: ensure probability sums to 1

$$\vdots$$

$$p(x_1 = 1, x_4 = 1, x_5 = 1 \mid x_2 = 1, x_3 = 1) = \frac{\sum_m w_m \delta(x^{(m)} = \{x_1 = 1, x_4 = 1, x_5 = 1\})}{\sum_m w_m}$$

- In summary, we get:

$$p(x_F \mid x_E) = \frac{\sum_m w_m \delta(x^{(m)})}{\sum_m w_m}$$