Lecture 8: Rejection Sampling and Markov chain review

Jacob Steinhardt

September 22, 2020

Last Time

- Latent variable models
  - Bayesian hierarchical model (COVID meta-analysis)
  - Hidden Markov model (ice cores)
  - Election forecasting model

This time: approximate inference via sampling algorithms

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \ldots))$

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \ldots))$

Want some way of "querying" the distribution. E.g.:

- What is the variance?
- What is the probability that $x_2 > x_1$?

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \ldots))$

Want some way of "querying" the distribution. E.g.:

- What is the variance?
- What is the probability that $x_2 > x_1$?

If we just have the pdf, unclear how to do this. Instead, suppose we have samples $x^{(1)}, \ldots, x^{(S)} \sim p$.

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \dots))$

Want some way of "querying" the distribution. E.g.:

- What is the variance?
- What is the probability that $x_2 > x_1$?

If we just have the pdf, unclear how to do this. Instead, suppose we have samples $x^{(1)}, \dots, x^{(S)} \sim p$.

- Can approximate any statistic f: $\mathbb{E}_{x \sim p}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})$

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \ldots))$

Want some way of "querying" the distribution. E.g.:

- What is the variance?
- What is the probability that $x_2 > x_1$?

If we just have the pdf, unclear how to do this. Instead, suppose we have samples $x^{(1)}, \ldots, x^{(S)} \sim p$.

- Can approximate any statistic f: $\mathbb{E}_{x \sim p}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})$
  - $f(x) = (x, x^2)$ (variance)
  - $f(x_1, x_2) = \mathbb{I}[x_2 > x_1]$

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \ldots))$

Want some way of "querying" the distribution. E.g.:

- What is the variance?
- What is the probability that $x_2 > x_1$?

If we just have the pdf, unclear how to do this. Instead, suppose we have samples $x^{(1)}, \ldots, x^{(S)} \sim p$.

- Can approximate any statistic f: $\mathbb{E}_{x \sim p}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})$
  - $f(x) = (x, x^2)$ (variance)
  - $f(x_1, x_2) = \mathbb{I}[x_2 > x_1]$
- Interpretable, efficient way to represent a distribution

Sampling: General Idea

Have a distribution $p(x)$ or $(p(x_1, x_2, \ldots))$

Want some way of "querying" the distribution. E.g.:

- What is the variance?
- What is the probability that $x_2 > x_1$?

If we just have the pdf, unclear how to do this. Instead, suppose we have samples $x^{(1)}, \ldots, x^{(S)} \sim p$.

- Can approximate any statistic f: $\mathbb{E}_{x \sim p}[f(x)] \approx \frac{1}{S} \sum_{s=1}^{S} f(x^{(s)})$
  - $f(x) = (x, x^2)$ (variance)
  - $f(x_1, x_2) = \mathbb{I}[x_2 > x_1]$
- Interpretable, efficient way to represent a distribution
- How many samples to get error ε?

Sampling Algorithms

Eventual target: Metropolis-Hastings algorithm (MCMC)

- Named among the “top 10 algorithms of the 20th century”

Sampling Algorithms

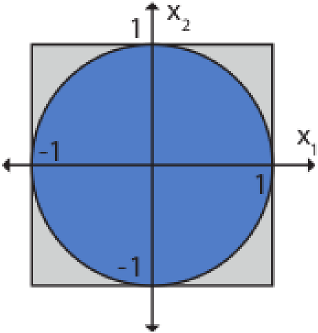
Eventual target: Metropolis-Hastings algorithm (MCMC)

- Named among the “top 10 algorithms of the 20th century”

First, need some build-up:

- Rejection sampling
- Markov chains

Warm-up: Sampling from unit circle



How to sample uniformly from the blue region?

Rejection sampling

[Jupyter demos]

Rejection sampling

[on board: general algorithm and normalization constant]

Rejection sampling

Input:

- Proposal distribution $q(x)$ (that we can sample from)
- Target distribution $p(x)$ (unnormalized; must satisfy $p(x) \leq q(x)$ for all x)

Rejection sampling

Input:

- Proposal distribution $q(x)$ (that we can sample from)
- Target distribution $p(x)$ (unnormalized; must satisfy $p(x) \leq q(x)$ for all x)

Algorithm:

- For $s = 1, \ldots, S$:
  - Sample $x \sim q$
  - With probability $p(x)/q(x)$, accept x and add to list of samples
  - Otherwise, reject

Rejection sampling

Input:

- Proposal distribution $q(x)$ (that we can sample from)
- Target distribution $p(x)$ (unnormalized; must satisfy $p(x) \leq q(x)$ for all x)

Algorithm:

- For $s = 1, \ldots, S$:
  - Sample $x \sim q$
  - With probability $p(x)/q(x)$, accept x and add to list of samples
  - Otherwise, reject

Pros: simple, can use with many pairs of densities, provides exact samples

Rejection sampling

Input:

- Proposal distribution $q(x)$ (that we can sample from)
- Target distribution $p(x)$ (unnormalized; must satisfy $p(x) \leq q(x)$ for all x)

Algorithm:

- For $s = 1, \ldots, S$:
  - Sample $x \sim q$
  - With probability $p(x)/q(x)$, accept x and add to list of samples
  - Otherwise, reject

Pros: simple, can use with many pairs of densities, provides exact samples
Cons: can be slow (curse of dimensionality)

Markov chains

Markov Chains

Markov chain: sequence $x_1, x_2, \ldots, x_T$ where distribution of x_t depends only on x_{t-1}

Defined by *transition distribution* $A(x^{\text{new}} \mid x^{\text{old}})$, together with initial state x_1

Examples:

- Random walk on a graph
- Repeatedly shuffling a deck of cards
- Process defined by

$$x_1 = 0, \quad x_t \mid x_{t-1} \sim N(0.9x_{t-1}, 1)$$

Markov Chains: Stationary Distribution

All "nice enough" Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1, and converges to some distribution $\bar{p}(x)$ as $T \to \infty$.

Markov Chains: Stationary Distribution

All “nice enough” Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1, and converges to some distribution $\bar{p}(x)$ as $T \to \infty$.

$\bar{p}(x)$ is called the *stationary distribution*, and the technical condition for “nice enough” is that the Markov chain is *ergodic*.

Markov Chains: Stationary Distribution

All "nice enough" Markov chains have the property that if T is large enough, the distribution over x_T is almost independent of x_1, and converges to some distribution $\bar{p}(x)$ as $T \to \infty$.

$\bar{p}(x)$ is called the *stationary distribution*, and the technical condition for "nice enough" is that the Markov chain is *ergodic*.

The distribution $\bar{p}(x)$ is also what we get if we count how many times x_t visits each state, as $T \to \infty$.

Markov Chains: Mixing Time

The *mixing time* is how long it takes for x_T to be close to the stationary distribution (we won't define this formally).

Markov Chains: Mixing Time

The *mixing time* is how long it takes for x_T to be close to the stationary distribution (we won't define this formally).

Example: card shuffling

- Mixing time is how many shuffles we need for deck to be "almost random"

Markov Chains: Mixing Time

The *mixing time* is how long it takes for x_T to be close to the stationary distribution (we won't define this formally).

Example: card shuffling

- Mixing time is how many shuffles we need for deck to be "almost random"

Other examples:

- Random walk on complete graph with n vertices
- Random walk on path of length n

The Annals of Applied Probability
1992, Vol. 2, No. 2, 294–313

TRAILING THE DOVETAIL SHUFFLE TO ITS LAIR

BY DAVE BAYER[1] AND PERSI DIACONIS[2]

Columbia University and Harvard University

We analyze the most commonly used method for shuffling cards. The main result is a simple expression for the chance of any arrangement after any number of shuffles. This is used to give sharp bounds on the approach to randomness: $\frac{3}{2}\log_2 n + \theta$ shuffles are necessary and sufficient to mix up n cards.

Key ingredients are the analysis of a card trick and the determination of the idempotents of a natural commutative subalgebra in the symmetric group algebra.

1. Introduction. The dovetail, or riffle shuffle is the most commonly used method of shuffling cards. Roughly, a deck of cards is cut about in half and then the two halves are riffled together. Figure 1 gives an example of a riffle shuffle for a deck of 13 cards.

A mathematically precise model of shuffling was introduced by Gilbert and Shannon [see Gilbert (1955)] and independently by Reeds (1981). A deck of n cards is cut into two portions according to a binomial distribution; thus, the chance that k cards are cut off is $\binom{n}{k}/2^n$ for $0 \le k \le n$. The two packets are then riffled together in such a way that cards drop from the left or right heaps

Markov chains: recap

- Governed by proposal distribution $A(x^{\text{new}} \mid x^{\text{old}})$
- Stationary distribution: limiting distribution of x_T
- Mixing time: how long it takes to get to stationary distribution