# CS 8803-MDM Lecture 26

## *Integration and Sampling*

Alexander Gray

agray@cc.gatech.edu

Georgia Institute of Technology

# Today

1. Integration and Sampling
2. Monte Carlo Variance Reduction
3. Markov Chain Monte Carlo

# Integration and Sampling

Why integration, and why sampling.

# Integration

Suppose we want to find

$$I = \int b(x)dx. \tag{1}$$

If $x$ is low-dimensional, we can use standard quadrature techniques. However, quadrature techniques effectively grid up the space, so that their cost is exponential in the dimensionality $D$ of $x$.

# Integration

Now suppose we have the form

$$b(x) = a(x)f(x), \tag{2}$$

where $f$ is a probability density function. We get this form whenever we want to compute the expected value of a function $a(x)$, where $x \sim f$:

$$I = \mathbb{E}(a) = \int a(x)f(x)dx. \tag{3}$$

# Integration and Sampling

The law of large numbers ensures that the sample mean over iid samples from $f$ converges to the integral:

$$\widehat{I} = \frac{1}{S} \sum_{s}^{S} a(x_s) \rightarrow \mathbb{E}(a) \tag{4}$$

as $S \rightarrow \infty$. $\widehat{I}$ is an unbiased estimator of $I$. This is called *Monte Carlo integration*.

# Integration and Sampling

Its error is effectively its variance, which is

$$\frac{1}{S}\int \left(a(x) - \mathbb{E}(a)\right)^2 dx = \sigma_a^2/S. \tag{5}$$

An estimate of this is

$$\widehat{\sigma}^2 = \frac{1}{S-1}\sum_{s}^{S}\left(a(x_s) - \widehat{I}\right)^2. \tag{6}$$

# Integration and Sampling

Expectations are ubiquitous in statistics but this idea happens to be critical for making Bayesian statistics practicable.

Recall that for a dataset $\{x\} \equiv \{x_1, \ldots, x_N\}$, the likelihood is

$$f(\{x\}|\theta) = f(x_1, \ldots, x_N|\theta) = \prod_{i=1}^{N} f(x_i|\theta) = L(\theta), \qquad (7)$$

and the posterior is

$$f(\theta|\{x\}) = \frac{f(\{x\}|\theta)f(\theta)}{\int f(\{x\}|\theta)f(\theta)d\theta} = \frac{L(\theta)f(\theta)}{c} \propto L(\theta)f(\theta) \qquad (8)$$

where $c = \int f(\{x\}|\theta)f(\theta)d\theta$.

# Integration and Sampling

Bayesians want to compute the posterior mean

$$\overline{\theta} = \mathbb{E}(\theta) = \int \theta f(\theta|\{x\}) d\theta. \tag{9}$$

Note that this has the form we specified, where the integrand $\theta \sim f(\theta|\{x\})$ and $a(\theta) = \theta$. So if we can draw samples $\theta_1, \ldots, \theta_S$ from the posterior $f(\theta|\{x\})$,

$$\frac{1}{S} \sum_{s}^{S} \theta_s \rightarrow \mathbb{E}(\theta) \tag{10}$$

as $S \rightarrow \infty$.

# Integration and Sampling

Bayesians also want to compute the $1-\alpha$ posterior interval $(a,b)$ such that $\int_{-\infty}^{a} f(\theta|\{x\})d\theta = \int_{b}^{\infty} f(\theta|\{x\})d\theta = \alpha/2$ and

$$\mathbb{P}(\theta \in (a,b)|\{x\}) = \int_{a}^{b} f(\theta|\{x\})d\theta = 1-\alpha. \quad (11)$$

This can also be done by drawing samples $\theta_s$ from the posterior $f(\theta|\{x\})$. We can approximate the posterior $1-\alpha$ interval by $(\theta_{\alpha/2}, \theta_{1-\alpha/2})$ where $\theta_{\alpha/2}$ is the $\alpha/2$ sample quantile of $\theta_1, \ldots, \theta_S$.

One common problem: we often cannot directly generate samples from $f(\theta|\{x\})$.

# Monte Carlo Variance Reduction

General techniques for faster Monte Carlo.

# Integration and Sampling

Note that we can treat any $b(x)$ as if it has the form $a(x)f(x)$ since $b(x) = b(x) \cdot 1$, so we can use the uniform distribution over the domain of $x$ as $f(x)$. This is called *crude Monte Carlo*.

Note that the error of Monte Carlo integration doesn't depend explicitly on the dimension $D$. However, if we say, sample uniformly, and the function has most of its mass concentrated in a small part of a high-dimensional space, this will be inefficient.

# Stratified Sampling

If we believe we know that the function looks different in disjoint regions $R_1 \subset R, \ldots, R_K \subset R$ of the domain $R$, we may profitably use a *stratified sampling* approach, drawing samples $x_s \sim \text{unif}(R_k)$ from each region separately:

$$\widehat{I} = \sum_{k=1}^{K} \frac{|R_k|}{|R|} \frac{1}{S_k} \sum_{s}^{S_k} b(x_s) \rightarrow \mathbb{E}(a). \tag{12}$$

For this idea to yield an advantage, the differences between the mean values of $b$ in each region should be greater than the variation within each region. Though it can be very useful, it relies on certain knowledge of the function.

# Importance Sampling

We may have some other function $q(x)$ which we believe is similar to $f(x)$, and we know how to sample from $q(x)$. We can always write

$$\int a(x)f(x)dx = \int a(x)\frac{f(x)}{q(x)}q(x)dx \tag{13}$$

The estimator draws samples $x_s \sim q$ instead of $x_s \sim f$ and

$$\widehat{I} = \frac{1}{S}\sum_{s}^{S} a(x_s)\frac{f(x_s)}{q(x_s)} \rightarrow \mathbb{E}(a). \tag{14}$$

# Importance Sampling

For this idea, called *importance sampling*, to work, we have to have $q(x) > 0$ everywhere in the domain. Ideally $q$ overestimates $f$ where $f$ is small. Though often powerful, it relies on having knowledge of a good $q$ for $f$.

# Control Variates

Another idea is that of *control variates*. If we believe we know another function $q(x)$ which is similar to $b(x)$, *and* whose integral we know analytically, we can write

$$\int b(x)dx = \int q(x)dx + \int (b(x) - q(x))\, dx \tag{15}$$

where we compute the first part analytically and estimate the second part using samples.

# Markov Chain Monte Carlo

Another Monte Carlo method, based on the idea of Markov chains.

# Markov Chains

Consider a sequence of discrete random variables $Z_t$ such that

$$\mathbb{P}(Z_{t+1} = z_j | Z_0 = z_k, \ldots, Z_t = z_i) = \mathbb{P}(Z_{t+1} = z_j | Z_t = z_i), \tag{16}$$

*i.e.* the transition probabilities between different values (or "states") depend only on the variable's current state. We say the random variable $Z$ is a *Markov process* and the sequence $\{Z_t\}$ is a *Markov chain* generated by the process.

The transition probability between states

$$K(z_i, z_j) \equiv \mathbb{P}(i \to j) \equiv \mathbb{P}(Z_{t+1} = z_j | Z_t = z_i) \tag{17}$$

is called the *jump kernel* or *transition kernel*. Let $A$ be the matrix whose $(i,j)^{th}$ element is $\mathbb{P}(i \to j)$.

# Markov Chains

Let $\pi_t(j) \equiv \mathbb{P}(Z_t = z_j)$ denote the probability that $Z$ is in state $j$ at time $t$. We start the chain with some setting $\pi_0$. The probability that $Z$ has value $s_i$ at time $t+1$ is

$$\begin{aligned} \pi_{t+1}(i) &= \mathbb{P}(Z_{t+1} = z_i) && (18) \\ &= \sum_k \mathbb{P}(Z_{t+1} = z_i | Z_t = z_k)\mathbb{P}(Z_t = z_k) && (19) \\ &= \sum_k \mathbb{P}(k \to i)\pi_t(k) = \sum_k A(k,i)\pi_t(k), && (20) \end{aligned}$$

or in matrix form,

$$\pi_{t+1} = \pi_t A. \quad (21)$$

We see that we can iterate by successively multiplying $A$:

$$\pi_t = \pi_{t-1}A = (\pi_{t-2}A)A = \pi_{t-2}A^2 = \pi_0 A^t. \quad (22)$$

# Markov Chains

Defining the $\tau$-step probability that the process is in state $j$ given that it started in state $i$ $\tau$ steps ago,

$$\mathbb{P}_\tau(i \to j) \equiv \mathbb{P}(Z_{t+\tau} = z_j | Z_t = z_i), \tag{23}$$

we say that the chain is *irreducible* if there exists a positive $\tau$ such that $\mathbb{P}_\tau(i \to j) > 0$ for each $(i, j)$. In other words, one can always go from any state to any other in some finite number of steps.

We say the chain is *aperiodic* if the number of steps required to move between any two states is not required to be a multiple of some integer. In other words, the chain is not forced into some cycle of fixed length between certain states.

# Markov Chains

If a Markov chain is irreducible and periodic, it will reach a *stationary* or *equilibrium distribution* $\pi^*$:

$$\pi_0, \ldots, \pi_T \rightarrow \pi^* \tag{24}$$

as $T \rightarrow \infty$. The probability vectors $\pi$ change less and less, getting closer to $\pi^*$. The stationary distribution stops changing, *i.e.*

$$\pi^* = \pi^* A. \tag{25}$$

If a Markov chain is irreducible and aperiodic and has stationary distribution $\pi^*$, then it is *ergodic*:

$$\mathbb{P}_t(i \rightarrow j) \equiv \mathbb{P}(Z_t = z_j | Z_0 = z_i) \rightarrow \pi^*(j) \quad \forall i, j \tag{26}$$

and we can consider $Z_t$ to be iid draws from $\pi^*$.

# Markov Chains

A sufficient condition for having a unique stationary distribution is that

$$\pi^*(i)\mathbb{P}(i \to j) = \pi^*(j)\mathbb{P}(j \to i) \quad \forall i, j \tag{27}$$

which is called the *detailed balance* or *reversibility* condition.

Now to extend all of this to continuous state spaces, we just need a transition kernel which satisfies $\int K(z', z)dz = 1$ and now

$$\pi_{t+1}(z) = \int \pi_t(z')K(z', z)dz \tag{28}$$

$$\pi^*(z) = \int \pi^*(z')K(z', z)dz. \tag{29}$$

# Metropolis-Hastings Algorithm

Our goal is to create a Markov chain which converges to a density $f(x) = g(x)/C$ from which we'd like to be able to draw iid samples, where the normalizing constant $C$ may not be known. We can only evaluate $g$ at any point $x$, but we cannot directly sample from $g$.

We will create a sequence of points $z_0, \ldots, z_T$, and some subset of this sequence will be used as if they were iid samples from $f$. Then we can use these to do Monte Carlo integration. This methodology is called *Markov chain Monte Carlo* (MCMC).

# Metropolis-Hastings Algorithm

Here's the *Metropolis algorithm*. We assume the transition kernel is symmetric, *i.e.* $K(z', z) = K(z, z')$. Most often the Gaussian kernel $K(z', z) = N(z', \sigma)$ is used. Starting with $z_t$ set to some point $z_0$:

1. Obtain a new candidate point $z_c$ from the distribution $K(z_t, z)$.
2. Compute the ratio

$$\alpha = \frac{f(z_c) \cdot C}{f(z_t) \cdot C} = \frac{g(z_c)}{g(z_t)}. \tag{30}$$

   If $\alpha > 1$, *i.e.* the jump increases the density, accept the candidate point, setting $z_{t+1} = z_c$. Otherwise accept it with probability $\alpha$. Repeat.

# Metropolis-Hastings Algorithm

We generalize this to non-symmetric kernels to obtain the *Metropolis-Hastings algorithm*:

1. Obtain a new candidate point $z_c$ from the distribution $K(z_t, z)$.
2. Accept $z_c$ with probability

$$\alpha = \min\left(\frac{g(z_c)K(z_c, z_t)}{g(z_t)K(z_t, z_c)}, 1\right). \tag{31}$$

   Repeat.

After some *burn-in* period of $t^*$ steps, we assume the chain has converged to its stationary distribution. We keep only the samples after that point, and use them as if they were iid samples from $f$.

# Metropolis-Hastings: Convergence

To show that the Metropolis-Hastings algorithm generates a Markov chain whose stationary distribution is $f$, it is sufficient to show that its transition kernel satisfies the detailed balance condition with $f$.

The actual transition probability is a combination of the transition kernel and the acceptance step:

$$\mathbb{P}(z' \to z) = K(z', z)\alpha(z', z) = K(z', z) \min\left(\frac{g(z)K(z, z')}{g(z')K(z', z)}, 1\right). \quad (32)$$

# Metropolis-Hastings: Convergence

We can verify that the detailed balance condition $\mathbb{P}(z' \rightarrow z)g(z') = \mathbb{P}(z \rightarrow z')g(z)$ holds by checking each of the three cases for the relative sizes of $g(z)K(z, z')$ and $g(z')K(z', z)$ $(>, <, =)$, to see that the stationary distribution from this kernel corresponds to draws from $f$.

# Metropolis-Hastings: Practical Issues

So far we just know that if enough samples are taken, we can use it to simulate samples from $f$. To use it in practice, we have to address some issues:

- Where should we start the chain?
- What should the width of the Gaussian be?
- When has it converged to the stationary distribution?
- What is the error of the final integral?

For all of these questions, there is essentially no solid theory.

# Metropolis-Hastings: Practical Issues

We try to start the chain at a mode if we can. If there are multiple modes we may try multiple chains. People argue about whether it is better in general to use a single chain or multiple chains.

The width of the Gaussian is critical for performance: if too small, we can stay trapped in a local mode; if too large, we devolve to uniform sampling from the high-dimensional space. We use trial and error, examining many time series plots. We say the chain is “poorly mixing” if it does not seem to be making much progress, which is often reflected in low acceptance rates.

# Metropolis-Hastings: Practical Issues

Before convergence, there is error due to the fact that the sequence is correlated; an estimate of the Monte Carlo integration error using MCMC is

$$\widehat{\sigma}^2 = \frac{\sigma_a^2}{S}\left(1 + 2\sum_{l=1}^{N-1}\rho_l(a)\right) \tag{33}$$

where $\rho_l(a)$ is the lag-$l$ auto-correlation in the sequence $\{a(z_t)\}$. Plotting this autocorrelation can be used to examine the progress of MCMC.

# Metropolis-Hastings: Practical Issues

One way to test for stationarity is to choose a split point in the sequence and perform a hypothesis test to determine if the mean of the first part of the sequence is the same as the mean of the second part, and there are others.