MSA101/MVE187 2021 Lecture 5 More basic simulation methods Introduction to Markov chain Monte Carlo (MCMC) methods

Petter Mostad

Chalmers University

September 12, 2022

Review and overview

- Prediction variable y_{pred}, data y_{data}, parameter θ.
- Make predictions using

$$\pi(y_{\text{pred}} \mid y_{\text{data}}) = \int \pi(y_{\text{pred}} \mid \theta)\pi(\theta \mid y_{\text{data}})\, d\theta$$

- One possibility: Generate sample from posterior $\pi(\theta \mid y_{\text{data}})$ and use Monte Carlo integration.
- Today:
  - Importance sampling: A better sample for the Monte Carlo integration.
  - Sampling Importance Resampling: More efficient sampling.
  - Approximating a function (the posterior?) using Laplace approximation.
  - Main feature: Introduction to (review of?) Markov chains and the Metropolis Hastings algorithm: Generating a sample from the posterior.

Importance sampling

- Review: Monte Carlo integration approximates

$$\mathrm{E}_f(h(x)) = \int h(x)f(x)\,dx$$

  where $f(x)$ is a probability density function by simulating $x_1, \ldots, x_m$ according to f and taking the average of $h(x_1), \ldots, h(x_m)$. The result has accuracy $\sqrt{Var_f(h(X))/m}$.

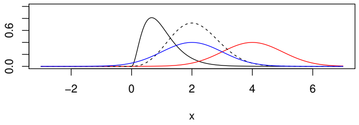
- Instead, we may re-write the integral as

$$\int \left[\frac{h(x)f(x)}{g(x)}\right] g(x)\,dx$$

  and simulate x_i according to g, taking the averages of $h(x_1)f(x_1)/g(x_1), \ldots, h(x_m)f(x_m)/g(x_m)$.

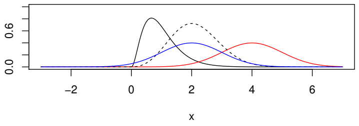
- A good idea if $Var_g(h(X)f(X)/g(X))$ is much smaller than $Var_f(h(X))$.

Importance sampling: Example



- In figure above, the black curve is $h(x)$ and the red curve is the density $f(x)$. The dotted curve is a scaled version of their product.
- Simulating points according to the red curve to compute the integral under the dotted curve will not be efficient.
- Better to simulate using the blue curve, which approximates the dotted curve, and then adjust using the quotient of the densities, as shown in the previous overhead.

Sampling Importance Resampling (SIR)



- In the figure above, assume you want to sample from a density proportional to the dotted curve: $\pi(x) \propto_x v(x)$.
- An *approximate* procedure starts with generating a sample

$$x_1, x_2, \ldots, x_N$$

according to the density given by the blue curve $g(x)$.
- Then one *resamples* from this sample (with replacement) using probability weights

$$w_i = \frac{v(x_i)/g(x_i)}{\sum_{j=1}^{N} v(x_j)/g(x_j)}.$$

The Laplace approximation

- For many simple models, the posterior $\pi(\theta \mid \text{data})$ for the parameter will have a shape that is close to a normal distribution.
- NOTE: For "scale parameters" (e.g., a standard deviation σ or a precision τ) reparametrization with a logarithm (e.g., $\theta_1 = \log(\sigma)$ and $\theta_2 = \log(\tau)$) with often make the posterior more normal-like.
- So, sometimes using some (multivariate) normal approximation for the true posterior distribution is a good enough approximation.
- If we use the normal density that has the same mode as the actual posterior and the same second derivatives of its logged density as that of the actual logged posterior, we call it the *Laplace approximation*.
- The Laplace approximation can be found for example by numerical differentiation of the logged posterior density, which needs to be known only up to an additive constant. See the R function `laplace` in the R package `LearnBayes`.

The (multivariate) Laplace as a Taylor approximation

Assume we have a density written

$$\pi(\theta) = C \cdot \exp(h(\theta))$$

for some known function h and unknown constant C. If $\hat{\theta}$ is the mode of the density, the second-degree Taylor approximation gives

$$h(\theta) \approx h(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^t H(\hat{\theta})(\theta - \hat{\theta})$$

where $H(\theta)$ is the Hessian matrix of second derivatives. We get

$$\pi(\theta) \approx C \cdot \exp(h(\hat{\theta})) \exp\left(-\frac{1}{2}(\theta - \hat{\theta})^t ((-H(\hat{\theta}))^{-1})^{-1}(\theta - \hat{\theta})\right).$$

This means that $\pi(\theta)$ might be approximated by a multivariate normal distribution with expectation $\hat{\theta}$ and covariance matrix $-H(\hat{\theta})^{-1}$. If we integrate both sides with respect to θ we get

$$C \approx \frac{1}{\exp(h(\hat{\theta}))|2\pi(-H(\hat{\theta}))^{-1}|^{1/2}}.$$

Example: Importance sampling using a multivariate normal approximation

The output from a machine depends on a vector $\theta = (\theta_1, \ldots, \theta_5)$ of external parameters as follows:

$$f(\theta) = \exp(-10((\theta_1 - 3)^2 + \theta_2^2 + \theta_3^2 + \theta_4^2 + \theta_5^2))$$

θ varies over time. What is the expected output?

- A data analysis has produced a posterior. For simplicity we assume $\theta \mid \text{data} \sim \text{Normal}_5(0, I)$.
- Computing the expectation directly by simulation can give large errors.
- Instead, find a Laplace approximation to the function you want to integrate and use this as an instrumental density in importance sampling. Faster convergence towards accuracy!

Why do we still need more simulation methods?

- Given a density function known up to a proportionality constant, we have looked at Rejection sampling, SIR, and importance sampling to generate (and use) samples.
- However, to give good accuracy, these methods require an approximate instrumental density $g(\theta)$.
- We now introduce *Markov chain Monte Carlo* (MCMC) which can much more easily and generally give accurate results.
- To study it we first need to review(?) a bit about Markov chains.

Review(?) of Markov chains

- Definition: A (discrete time, time-homogeneous) Markov chain with kernel K is a sequence of random variables $X^{(0)}, X^{(1)}, X^{(2)}, \ldots$ satisfying, for all t,

$$\pi(X^{(t)} \mid X^{(0)}, X^{(1)}, \ldots, X^{(t-1)}) = \pi(X^{(t)} \mid X^{(t-1)}) = K(X^{(t-1)}, X^{(t)})$$

- Example: In the case of a state space with n possible values, a distribution is represented by a vector of length n summing to 1, and the *transition probabilities* are given in an $(n \times n)$ matrix K.
- A *limiting distribution* is the probability distribution (if it exists) $\lim_{t\to\infty} X^{(t)}$.
- A *stationary distribution* f is one satisfying

$$f(y) = \int K(x, y) f(x)\, dx.$$

In the discrete case, a stationary distribution becomes a probability vector v so that $vK = v$, i.e., a left eigenvector for K.

Ergodic Markov chains

- For Markov chains with discrete state spaces we have:
  - A Markov chain is *irreducible* if for any pairs of states x and y there is an n so that the probability that the chain moves from x to y in n steps is nonzero.
  - If a chain starts at x the (random) number of steps T before it revisits x is called the return time. A state is called positive recurrent if the expectation of T is finite.
  - The *period* of a state x is the greatest common divisor of the numbers m so that $\Pr(T = m) > 0$. In an irreducible chain all states have the same period. If this period is 1 the chain is called *aperiodic*.
  - A Markov chain is called *ergodic* if it is irreducible, aperiodic, and all states have a finite expected return time.
- For Markov chains with continuous state spaces, ergodicity is based on similar definitions.

Fundamental limit theorem for ergodic Markov chains

- If $X_0, X_1, X_2, \ldots$ is an ergodic Markov chain then there exists a *unique* positive stationary distribution which is the *limiting distribution* for the chain.
- In other words, if we run an ergodic Markov chain long enough, its values will eventually be an approximate sample from the limiting distribution, which can be identified as the unique distribution that is stationary for the chain.

How to use this for MCMC

- The MCMC algorithm constructs a Markov chain which has a stationary distribution equal to the target density we would like to generate a sample from.
- To use MCMC one needs to check that the constructed Markov chain is ergodic, but this is usually simple.
- Running the Markov chain will then eventually create values which are an approximate sample from the target distribution.
- Will this approximate sample give an approximately correct computation for the prediction?

The Ergodic theorem

- This theorem says that, when $X^{(0)}, \dots, X^{(t)}, \dots,$ is sampled from an ergodic Markov chain with stationary distribution f, we have that

$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} h(X^{(t)}) = E_f[h(X)]$$

- When the sample is instead a random sample from f, this is the law of large numbers; we then also have the extension to the Central Limit Theorem, telling us how fast the convergence is.
- In the ergodic case, we still have convergence, but we don't know as easily how fast it is.

MCMC simulation: General idea

- We have a "target density" $f(x)$ (known only up to a proportionality constant) and we would like to generate a sample (or approximate sample) from this density.
- We use the *Metropolis-Hastings algorithm* to construct a Markov chain $x_0, x_1, \ldots$ which has the target density as a stationary distribution.
- After checking that the chain is ergodic, we know that if we simulate long enough, the chain will provide an approximate sample which can be used for Bayesian inference and predictions with Monte Carlo integration.

The Metropolis-Hastings algorithm

Given a probability density f that we want to simulate from. Construct a *proposal function* $q(y \mid x)$ which for every x gives a probability density for a proposed new value y. The algorithm starts with a choice of an initial value $x^{(0)}$ for x, and then simulates $x^{(t+1)}$ given $x^{(t)}$ for $t \geq 0$. Specifically, given $x^{(t)}$,

- Simulate a new value y according to $q(y \mid x^{(t)})$.
- Compute the acceptance probability

$$\rho(x^{(t)}, y) = \min\left(\frac{f(y)q(x^{(t)} \mid y)}{f(x^{(t)})q(y \mid x^{(t)})}, 1\right).$$

- Set

$$x^{(t+1)} = \begin{cases} y & \text{with probability } \rho(x^{(t)}, y) \\ x^{(t)} & \text{with probability } 1 - \rho(x^{(t)}, y) \end{cases}$$

Proving that the Metropolis-Hastings algorithm works

- The missing ingredient is to prove that the Metropolis-Hastings (MH) algorithm has the target density as a stationary distribution.
- We do this by showing
  - The MH chain satisfies the *detailed balance condition* relative to the target density.
  - If a chain satisfies the detailed balance condition relative to a density f then f is a stationary distribution.

The detailed balance condition

- A Markov chain satisfies the *detailed balance condition* relative to a density f if, for all x, y,

  $$f(x)K(x,y) = f(y)K(y,x)$$

  where $K(x,y)$ is the kernel of the Markov chain. The chain is then called a *time reversible* Markov chain.
- If a chain satisfies detailed balance relative to f, then f must be a stationary distribution.
- Proof by integrating over x:

  $$\int K(x,y)f(x)\,dx = \int K(y,x)f(y)\,dx = f(y).$$

The chain defined by Metropolis-Hastings satisfies the detailed balance condition relative to $f(x)$

- Assume first that $\rho(x,y) < 1$ (with $x \neq y$). Then

$$\begin{aligned} f(x)K(x,y) &= f(x)q(y \mid x)\rho(x,y) = f(x)q(y \mid x)\frac{f(y)q(x \mid y)}{f(x)q(y \mid x)} \\ &= f(y)q(x \mid y) = f(y)q(x \mid y)\rho(y,x) = f(y)K(y,x) \end{aligned}$$

  The next to last step is because $\rho(y,x) = 1$ when $\rho(x,y) < 1$.
- If we start with $\rho(x,y) = 1$ the situation is clearly symmetrical, and we get the same result.

Note that...

- ...the Metropolis-Hastings algorithm *only* requires knowledge of the target density $f(x)$ up to a constant not involving x, as the density only appears in the quotient $f(y)/f(x)$ in the algoritm.
- ...the Metropolis-Hastings algorithm *only* requires knowledge of the proposal density up to a constant, for the same reason.
- ...similarly, smart versions of the Metropolis-Hastings algorithm uses proposal flunctions so that many factors in the acceptance probability
$$\frac{f(y)q(x \mid y)}{f(x)q(y \mid x)}$$
cancel each other.