Lecture 9: Markov Chain Monte Carlo

UNIVERSITY OF
SAN FRANCISCO

James D. Wilson

MSAN 628

Computational Statistics

Simulation so far: Ingredients

- Target density: (posterior density) $p(\theta \mid y)$
  - posterior is calculated using Bayes rule from prior $\pi(\theta)$ and density $f(y \mid \theta)$
  - everything generalizes to *any* target density!
- Unnormalized density: $q(\theta \mid y) = \pi(\theta) f(y \mid \theta)$
  - the kernel of Bayes rule
- Proposal distribution: $g(\theta)$ - must be an integrable and non-negative function for all θ.
  - We'd like something "close" to $p(\theta \mid y)$
  - The further away $g(\theta)$ is from $p(\theta \mid y)$, the less efficient sampling will be!

Simulation Strategies so Far

- Monte Carlo: Approximate $\mathbb{E}[h(\theta) \mid y]$ given a known and ready-to-simulate-from posterior $p(\theta \mid y)$
- Rejection Sampling: Sample values of θ from a "funny" but known posterior $p(\theta \mid y)$
- Importance Sampling: Approximate $\mathbb{E}[h(\theta) \mid y]$ given an *unknown* posterior $p(\theta \mid y)$

Markov chain Monte Carlo Basics

Goal: Simulate values of θ from an *unknown* target density $p(\theta \mid y)$

Basic Strategy:

- Propose distribution $g_1(\theta)$ to approximate $p(\theta \mid y)$
- Simulate from $g_1(\theta) \rightarrow \theta^1$
- Based on θ^1, correct g_1 and propose new distribution $g_2(\theta)$ to approximate $p(\theta \mid y)$
- Simulate from $g_2(\theta) \rightarrow \theta^2$
- Repeat until proposal distribution is "close enough"

Key Idea: The draw of $\theta^s \mid \theta^1, \ldots, \theta^{s-1}$ forms a Markov chain! And the proposal distribution improves at each step.

Markov chain Monte Carlo Basics

- We generate a Markov chain of samples in the following manner:
  - Initialize with some arbitrary θ^0
  - For $t > 0$, draw θ^t from a transition distribution $T_t(\theta^t \mid \theta^{t-1})$ that depends on the previous draw. (Note that this may depend on the time and hence be time inhomogeneous)
- The transition probability distribution, T_t, *must* be chosen so that the Markov chain converges to a unique stationary distribution that is the posterior distribution $p(\theta \mid y)$

Markov chain Monte Carlo Basics

Elevator Pitch

At time t, we draw θ from the transition distribution $T_t(\theta^t \mid \theta^{t-1})$ which describes a Markov chain whose stationary distribution is the target posterior distribution $p(\theta \mid y)$. This is a method of successive approximation.

The hard part: coming up with an appropriate T_t that leads to $p(\theta \mid y)$

Other Important Considerations

- MCMC algorithms get better with each sample, but it is really difficult to quantify how much better.
- This means that in practice, one generally tosses out the first m samples $\theta^1, \ldots, \theta^m$ and then assumes that the remaining samples are "good". Here, the first m samples are called the burn-in
- In general, one always needs to assess convergence of the samples of θ to a stationary distribution. (more on this later)

The Gibbs Sampler

Suppose that θ is d-dimensional. Then draws of θ_j^t at each stage depend on the other values in the vector at time t, θ_{-j}^t.

The Gibbs Sampler

- Randomly draw θ^0 (say from the prior distribution)
- For $t > 0$
  - For all $j = 1, \ldots, d$, sample θ_j^t from $p(\theta_j \mid \theta_{-j}^{t-1}, y)$

This is the best case scenario. We suppose that we have a nice representation of the conditional distribution of one entry in the vector given the other values.

Toy Example of the Gibbs Sampler

$$\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} \mid y \sim N\left(\begin{pmatrix}y_1\\ y_2\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

- In practice we won't know the posterior distribution above.
- But we do know (thanks to the properties of the multivariate Gaussian distribution) that
  - $\theta_1 \mid \theta_2, y \sim N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2)$
  - $\theta_2 \mid \theta_1, y \sim N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2)$
- So, the Gibbs sampler just relies on alternatively sampling from these normal distributions once we initiate with a random guess θ^0. I run this in R for an example.

The Metropolis Algorithm

- The Metropolis algorithm is an adaptation of a random walk with an acceptance/rejection rule used to converge to the desired target distribution.
- It relies on a proposal distribution $J_t(\theta^t \mid \theta^{t-1})$ that is symmetric. Namely,

$$J_t(\theta_a \mid \theta_b) = J_t(\theta_b \mid \theta_a)$$

for all θ_a, θ_b, and t.

The Metropolis Algorithm

- Initiate with some reasonable θ^0
- For $t > 0$
  - Simulate θ^* from $J_t(\theta \mid \theta^{t-1})$
  - Calculate the ratio:

$$r = \frac{p(\theta^* \mid y)}{p(\theta^{t-1} \mid y)} = \frac{q(\theta^* \mid y)}{q(\theta^{t-1} \mid y)}$$

  - Set

$$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

Example of the Metropolis Algorithm

Target Distribution:

$$\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} \mid y \sim N\left(\begin{pmatrix}y_1\\ y_2\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

Again, we are supposing that we don't know this distribution.

We make the following proposal density:

$$J_t(\theta^t \mid \theta^{t-1}) = N(\theta^t \mid \theta^{t-1}, I),$$

where $N(\theta^t \mid \theta^{t-1}, I)$ is the bivariate normal density evaluated at θ^t. It is readily verified that this density is symmetric, as needed.

The Metropolis-Hastings Algorithm

- The major weakness of the Metropolis algorithm is that it requires a *symmetric* proposal distribution $J_t(\theta^t \mid \theta^{t-1})$, which in practice can be very difficult to specify.
- The Metropolis-Hastings algorithm generalizes the Metropolis algorithm by no longer requiring J_t to be symmetric.
- The algorithm is very similar to the Metropolis algorithm, with the exception that the ratio r is defined differently.

The Metropolis-Hastings Algorithm

- Initiate with some reasonable θ^0
- For $t > 0$
  - Simulate θ^* from $J_t(\theta \mid \theta^{t-1})$
  - Calculate the ratio:

$$r = \frac{p(\theta^* \mid y)}{p(\theta^{t-1} \mid y)} \frac{J_t(\theta^* \mid \theta^{t-1})}{J_t(\theta^{t-1} \mid \theta^*)} = \frac{q(\theta^* \mid y)}{q(\theta^{t-1} \mid y)} \frac{J_t(\theta^{t-1} \mid \theta^*)}{J_t(\theta^* \mid \theta^{t-1})}$$

  - Set

$$\theta^t = \begin{cases} \theta^* & \text{with probability } \min(r, 1) \\ \theta^{t-1} & \text{otherwise} \end{cases}$$

Example of Metropolis-Hastings

Target Distribution:

$$\begin{pmatrix}\theta_1\\ \theta_2\end{pmatrix} \mid y \sim N\left(\begin{pmatrix}y_1\\ y_2\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)$$

Again, we are supposing that we don't know this distribution.

We make the following proposal density:

$$J_t(\theta^t \mid \theta^{t-1}) = N(\theta^t \mid 0.8\theta^{t-1}, I),$$

where $N(\theta^t \mid 2\theta^{t-1}, I)$ is the bivariate normal density evaluated at θ^t. Note here that our proposal density is *not* symmetric! So, the Metropolis algorithm cannot be used.

Comparison of MCMC algorithms

- Gibbs: Requires the knowledge of conditional probabilities of a θ_j given θ_{-j}. The most efficient, but it is often difficult to derive.
- Metropolis: Rejection algorithm with a *symmetric* transition distribution / proposal distribution $J(\theta^t \mid \theta^{t-1})$
- Metropolis-Hastings: Rejection algorithm with a non-symmetric transition / proposal distribution $J(\theta^t \mid \theta^{t-1})$. The most general of the three MCMC methods
- All of the methods seek simulated values from an unknown posterior distribution $p(\theta \mid y)$