Markov Chain Monte Carlo

John C. Chao

Econ 721 Lecture Notes

November 16, 2021

MCMC

- **Problem to be considered:** Suppose we are able to evaluate a possibly un-normalized density function π^*. Our goal is to draw a sample from the (normalized) probability density function

$$\pi(x) = \frac{\pi^*(x)}{C},$$

where the normalization constant

$$C = \int_{\mathcal{S}} \pi^*(x)\, dx$$

may be unknown to us.

MCMC

- **Application to Bayesian Econometrics or Statistics:** Let $L(\theta; y)$ be the likelihood function of a statistical experiment with data y and unknown parameter (vector) $\theta \in \Theta$ and let $p(\theta)$ denote the prior density. Then, the target distributon of interest might be the posterior distribution of θ given the data y whose density (up to a normalization constant) can be represented by

$$\pi(\theta|y) \propto L(\theta; y)\, p(\theta) = \pi^*(\theta|y)$$

- The use of Markov Chain Monte Carlo (MCMC) methods allow us to overcome the following difficulties typically encountered in the implementation of Bayesian procedures

(i) The state space $\mathcal{S}$ is typically high-dimensional.

(ii) Direct simulation from π is too complex to be feasible.

(iii) Computing the normalization constant C is as difficult as the entire simulation problem.

Markov Chain Monte Carlo

- **The MCMC Approach:** Let $\pi(\cdot)$ be a target density on some state space $\mathcal{S}$ (e.g., $\mathcal{S} \subseteq \mathbb{R}^d$). The MCMC approach requires us to construct a Markov chain on $\mathcal{S}$, i.e., a Markov chain with transition probability

  $$P(x, dy) \text{ for } x, y \in \mathcal{S}$$

  such that $\pi(\cdot)$ is its stationary distribution, so that

  $$\int_{x \in \mathcal{S}} \pi(dx) P(x, dy) = \pi(dy).$$

  The hope is that if we run the Markov chain for a long time (starting from some initial value in the state space); then, for n sufficiently large the distribution of X_n will be approximately that of the stationary distribution $\pi(\cdot)$.

MCMC

- **Metropolis-Hastings Algorithm:** The Metropolis-Hasting algorithm is a particular type of MCMC which requires the choice of a proposal distribution $q(y|x)$ which is a friendly distribution from which we know how to generate a sample. Given $q(y|x)$, the Metropolis-Hasting algorithm creates a sequence of observations $X_0, X_1, X_2, ...$ based on the following algorithm.
- **Algorithm:** Choose X_0 arbitrarily. Suppose we have generated $X_0, ..., X_n$; then, to generate X_{n+1}, we proceed as follows:

1. Generate a proposal or candidate value $Y_{n+1} \sim q(y|x)$
2. Evaluate

$$\alpha = \alpha(X_n, Y_{n+1})$$

where

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y) q(x|y)}{\pi(x) q(y|x)}\right\}$$

Metropolis-Hastings Algorithm

3. Set

$$X_{n+1} = \begin{cases} Y_{n+1} & \text{with prob } \alpha \\ X_n & \text{with prob } 1-\alpha \end{cases}$$

i.e., we accept Y_{n+1} as the new value X_{n+1} with probability $\alpha = \alpha\left(X_n, Y_{n+1}\right)$ or reject Y_{n+1} and set $X_{n+1} = X_n$ (the old value) with probability $1 - \alpha\left(X_n, Y_{n+1}\right)$.

- **Remarks:**

(i) A simple way to carry out step 3 above is to generate U ∼Unif$(0,1)$. If $U < \alpha$. set $X_{n+1} = Y_{n+1}$; otherwise, set $X_{n+1} = X_n$.

Metropolis-Hastings Algorithm

- **Remarks:**

(ii) A common choice for $q\left(y|x\right)$ is to specify it to be the pdf of $N\left(x,\omega^{2}\right)$ for some $\omega>0$. This means that the proposal is drawn from a normal distribution centered at the current value x. Since in this case

$$q\left(y|x\right)=\frac{1}{\omega\sqrt{2\pi}}\exp\left\{-\frac{1}{2\omega^{2}}\left(y-x\right)^{2}\right\}$$

we see that the proposal density is symmetric, i.e., $q\left(y|x\right)=q\left(x|y\right)$. Hence, in this case, α simplifies to

$$\begin{aligned}\alpha &= \alpha\left(X_{n},Y_{n+1}\right)=\min\left\{1,\frac{\pi\left(Y_{n+1}\right)q\left(X_{n}|Y_{n+1}\right)}{\pi\left(X_{n}\right)q\left(Y_{n+1}|X_{n}\right)}\right\}\\ &= \min\left\{1,\frac{\pi\left(Y_{n+1}\right)}{\pi\left(X_{n}\right)}\right\}.\end{aligned}$$

Metropolis-Hastings Algorithm

- **Remarks (con't):**

(iii) Note also that since $\alpha\left(x,y\right)$ only depends on the ratio $\pi\left(y\right)/\pi\left(x\right)=\pi^{*}\left(y\right)/\pi^{*}\left(x\right)$, we would not need to know the normalization constant C in order to implement this algorithm.

- **Claim:** Given that

$$\alpha\left(x,y\right)=\min\left\{1,\frac{\pi\left(y\right)q\left(x|y\right)}{\pi\left(x\right)q\left(y|x\right)}\right\}$$

the resulting Markov chain is reversible with respect to $\pi\left(\cdot\right)$.

Metropolis-Hastings Algorithm

- **Proof of Claim (Sketched):** We need to show that

$$\pi(dx) P(x, dy) = \pi(dy) P(y, dx) \text{ for all } x, y \in \mathcal{S}.$$

It suffices to assume that $x \neq y$ since otherwise it is trivial. Note that $P(x, dy)$ is the probability of jumping from x to y. To do so requires two things: (i) $y \in dy$ is generated in accordance with the conditional distribution $q(y|x)$ and (ii) y is accepted. Hence, we have roughly

$$\begin{aligned}
\pi(dx) P(x, dy) &= \pi(x) dx \alpha(x, y) q(y|x) dy \\
&= \pi(x) q(y|x) \min\left\{1, \frac{\pi(y) q(x|y)}{\pi(x) q(y|x)}\right\} dxdy \\
&= \min\{\pi(x) q(y|x), \pi(y) q(x|y)\} dxdy \\
&= \pi(y) q(x|y) \min\left\{\frac{\pi(x) q(y|x)}{\pi(y) q(x|y)}, 1\right\} dxdy \\
&= \pi(y) dy \alpha(y, x) q(x|y) dx \\
&= \pi(dy) P(y, dx).
\end{aligned}$$

Metropolis-Hastings Algorithm

- **Example:** Suppose that the target distribution is the Cauchy with pdf given by

$$\pi(x) = \frac{1}{\pi}\frac{1}{1+x^2}$$

Here, we can take $q(y|x)$ to be $N(x, \omega^2)$ and because $q(y|x) = q(x|y)$ in this case, we have

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\} = \min\left\{1, \frac{1+x^2}{1+y^2}\right\}$$

Hence, the Metropolis-Hastings algorithm in this case is to draw

$$Y_{n+1} \sim N(x, \omega^2)$$

and set

$$X_{n+1} = \begin{cases} Y_{n+1} & \text{with prob } \alpha(X_n, Y_{n+1}) \\ X_n & \text{with prob } 1-\alpha(X_n, Y_{n+1}) \end{cases}$$

Metropolis-Hastings Algorithm

- **Example (con't):** Wasserman (2004) showed the results of an experiment where three chains of length $T = 1000$ were generated using

$$\omega = 0.1,\ 1,\ 10.$$

He found that setting $\omega = 0.1$ requires the chain to take steps that were too small, so that it does not "explore" much of the state space. On the other hand, setting $\omega = 10$ often results in proposals that are in the tails of the distribution, leading to small values of $\alpha\left(X_n, Y_{n+1}\right)$ and, thus, frequent rejection of the proposals. Hence, the chain ends up "getting stuck" in the same place quite often. The choice $\omega = 1$ turns out to avoid the two extremes and results in a chain that performs much better. Hence, we can think of ω as a tuning parameter whose selection will affect the efficiency of the algorithm.

Gibbs Sampling

- **Two-Variables Case:** Starting at $\left(X_0, Y_0\right)$ and suppose we have drawn $\left(X_0, Y_0\right), \ldots, \left(X_n, Y_n\right)$; then, the Gibbs sampler for getting $\left(X_{n+1}, Y_{n+1}\right)$ is

$$\begin{aligned} X_{n+1} &\sim f\left(x \middle| Y_n\right), \\ Y_{n+1} &\sim f\left(y \middle| X_{n+1}\right). \end{aligned}$$

the Gibbs sampling process then involves iteration on this step until we obtain the needed sample.

Gibbs Sampling

- **General Case:** Suppose that the target distribution is $\pi\left(x\right)$ where x is d-dimensional, say $x \in \mathcal{S} \subseteq \mathbb{R}^d$. Let

$$\begin{aligned} x &= \left(x_1, x_2, ..., x_d\right)', \\ x^{(n)} &= \left(x_1^{(n)}, x_2^{(n)}, ..., x_d^{(n)}\right)' \text{ - } x \text{ obtained in the } n^{th} \text{ iteration} \\ x_{[-i]} &= \left(x_1, ..., x_{i-1}, x_{i+1}, ..., x_d\right)', \\ x_{[-i]}^{(n)} &= \left(x_1^{(n)}, ..., x_{i-1}^{(n)}, x_{i+1}^{(n)}, ..., x_d^{(n)}\right)' \end{aligned}$$

and let $\pi\left(x|x_{[-i]}^{(n)}\right)$ be the conditional density of x given $x_{[-i]}^{(n)}$.

Gibbs Sampling

- There are two versions of Gibbs sampling.

1. **Random-Scan Gibbs Sampler:** Given that in the n^{th} iteration we obtain $x^{(n)}$, we perform the following steps to obtain $x^{(n+1)}$.
   - (i) Randomly select a coordinate $i \in \{1, 2, ..., d\}$ according to some probability vector $(p_1, .., p_d)$, e.g. $(p_1, .., p_d) = (1/d, ..., 1/d)$.
   - (ii) Draw $x_i^{(n+1)}$ from the conditional distribution $P_i = P\left(x_{[-i]}^{(n)}, dx_i\right) = \pi\left(x_i | x_{[-i]}^{(n)}\right) dx_i$ and leave the remaining components unchanged, i.e., let

$$x^{(n)} = \left(x_1^{(n)}, ..., x_{i-1}^{(n)}, x_i^{(n+1)}, x_{i+1}^{(n)} ..., x_d^{(n)}\right)'$$

Gibbs Sampling

2. **Systematic-Scan Gibbs Sampler:** Given that in the n^{th} iteration we obtain $x^{(n)}$, we draw

$$\begin{aligned}
x_1^{(n+1)} &\sim \pi\left(x_1 | x_{[-1]}^{(n)}\right) \\
x_2^{(n+1)} &\sim \pi\left(x_2 | x_1^{(n+1)}, x_3^{(n)}, ..., x_d^{(n)}\right) \\
&\vdots \\
x_d^{(n+1)} &\sim \pi\left(x_d | x_1^{(n+1)}, x_2^{(n+1)}, ..., x_{d-1}^{(n+1)}\right)
\end{aligned}$$

- **Remark:** When $d = 2$, the systematic-scan Gibbs sampler reduces to

$$\begin{aligned}
x_1^{(n+1)} &\sim \pi\left(x_1 | x_2^{(n)}\right), \\
x_2^{(n+1)} &\sim \pi\left(x_2 | x_1^{(n+1)}\right)
\end{aligned}$$

We do this repeatedly from some initial value $x_2^{(0)}$ to get the sequence $x_1^{(1)}, x_2^{(1)}, x_1^{(2)}, x_2^{(2)}, ..., x_1^{(T)}, x_2^{(T)}$.

Gibbs Sampling

- **Example:** Let

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right)$$

where $|\rho| < 1$ so that

$$\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} > 0 \text{ (i.e., it is positive definite).}$$

In this case, the Markov chain is generated by iterating

$$\begin{aligned} x_1^{(n+1)} | x_2^{(n)} &\sim N\left(\rho x_2^{(n)}, 1 - \rho^2 \right), \\ x_2^{(n+1)} | x_1^{(n+1)} &\sim N\left(\rho x_1^{(n+1)}, 1 - \rho^2 \right). \end{aligned}$$

Gibbs Sampling

- **Example (con't):** The marginal distribution of $x^{(n)} = \left(x_1^{(n)}, x_2^{(n)} \right)'$ can be shown to be

$$\begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix}$$

$$\sim \quad N\left(\begin{pmatrix} \rho^{2n-1} x_2^{(0)} \\ \rho^{2n} x_2^{(0)} \end{pmatrix}, \begin{pmatrix} 1-\rho^{2(2n-1)} & \rho\left(1-\rho^{2(2n-1)}\right) \\ \rho\left(1-\rho^{2(2n-1)}\right) & 1-\rho^{4n} \end{pmatrix} \right)$$

so that

$$\begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \end{pmatrix} \xrightarrow{d} N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right) \quad \text{as } n \longrightarrow \infty.$$

Note also that the rate of convergence is very fast in this case.

Metropolis within Gibbs Algorithm:

- To implement the Gibbs sampling algorithm, we must be able to draw from the conditional distributions. If that is not the case; then, we can still implement the Gibbs sampling algorithm by drawing each observation using a Metropolis-Hastings step using q as a proposal distribution to draw x and $\widetilde{q}$ a proposal distribution to draw y.
- **Metropolis within Gibbs Algorithm:** Choose X_0 arbitrarily. Suppose we have generated $X_0, X_1, ..., X_n$; then, to generate X_{n+1}, we proceed as follows

1. Generate a proposal or candidate value $Z \sim q\left(z|X_n\right)$
2. Evaluate

$$\alpha_X = \alpha\left(X_n, Y_n\right) = \min\left\{1, \frac{\pi\left(Z, Y_n\right) q\left(X_n|Z\right)}{\pi\left(X_n, Y_n\right) q\left(Z|X_n\right)}\right\}$$

Metropolis within Gibbs Algorithm:

- **Metropolis within Gibbs Algorithm (con't):**

3. Set

$$X_{n+1} = \begin{cases} Z & \text{with prob } \alpha_X \\ X_n & \text{with prob } 1 - \alpha_X \end{cases}$$

4. Generate a proposal or candidate value $Z \sim \widetilde{q}\left(z \middle| Y_n\right)$
5. Evaluate

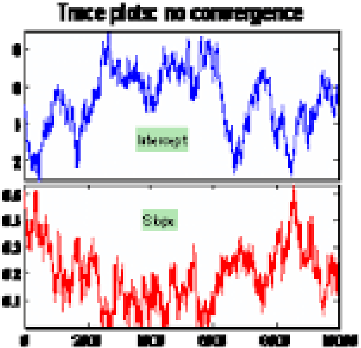
$$\alpha_Y = \alpha\left(X_{n+1}, Y_n\right) = \min\left\{1, \frac{\pi\left(X_{n+1}, Z\right)\widetilde{q}\left(Y_n | Z\right)}{\pi\left(X_{n+1}, Y_n\right)\widetilde{q}\left(Z \middle| Y_n\right)}\right\}$$

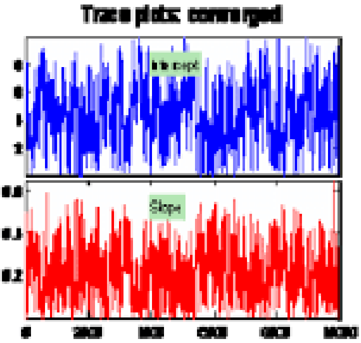
6. Set

$$Y_{n+1} = \begin{cases} Z & \text{with prob } \alpha_Y \\ Y_n & \text{with prob } 1 - \alpha_Y \end{cases}$$

Convergence Diagnostics

- The theory of Markov chains tells us thta an irreducible, aperiodic Markov chain will eventually converge to its stationary distribution. However, a more practical question that needs to be answered is how do we know that our chain has approximately converged after T draws? Below are some of the common methods used to perform convergence diagnostics.
- **Traceplots:**





Convergence Diagnostics

- **Autocorrelations:** Another way to assess convergence is to look at the sample autocorrelations between the draws of the Markov chain. Let

$$\widehat{\rho}_k = \frac{\sum_{t=1}^{T-k} \left(X_t - \overline{X}\right)\left(X_{t+k} - \overline{X}\right)}{\sum_{t=1}^{T} \left(X_t - \overline{X}\right)^2}$$

  When k is large, we would expect the k^{th} lag autocorrelation to be small, so if $\widehat{\rho}_k$ is still relatively high when k is large, this indicates that the chain is mixing slowly as there is a high level of correlation in our draws.

- **Gelman and Rubin Multiple Sequence Diagnostics:** Steps (for each parameter)

(i) Run $m \geq 2$ chains of length $2n$ from overdispersed starting values.

(ii) Discard the first n draws in each chain

Convergence Diagnostics

(iii) Calculate the within-chain and between chain variances as follows:
Within Chain Variance:

$$W = \frac{1}{m} \sum_{j=1}^{m} s_j^2$$

where

$$s_j^2 = \frac{1}{n-1} \sum_{t=1}^{n} \left(\theta_{tj} - \overline{\theta}_j \right)^2 \text{ with } \overline{\theta}_j = \frac{1}{n} \sum_{t=1}^{n} \theta_{tj}$$

Note that W is likely to underestimate the true variance of the stationary distribution since with a finite number of draws our chains have probably not reached all the points of the stationary distribution

Convergence Diagnostics

(iii) **Between Chain Variance:**

$$B = \frac{n}{m-1} \sum_{j=1}^{m} \left(\bar{\theta}_j - \bar{\bar{\theta}} \right)^2$$

where

$$\bar{\bar{\theta}} = \frac{1}{m} \sum_{j=1}^{m} \bar{\theta}_j .$$

It follows that B/n is an estimate of the between chain variance.
Estimated Variance of the Stationary Distribution:

$$\widehat{\sigma}_\theta^2 = \left(1 - \frac{1}{n} \right) W + \frac{1}{n} B$$

Because of overdispersion of the initial values, $\widehat{\sigma}_\theta^2$ is likely to overestimate the true variance. On the other hand, if the initial distribution happens to be the stationary distribution, then this is an unbiased estimator.

Convergence Diagnostics

(iii) **Potential Scale Reduction Factor:**

$$\widehat{R} = \frac{\widehat{\sigma}_\theta}{\sqrt{W}}$$

If $\widehat{R}$ is high (say, greater than 1.1 or 1.2); then, we need to run our chains out longer in order to ensure approximate convergence.

- **Remarks:**

(i) If we have more than one parameter, then we need to calculate $\widehat{R}$ for each parameter.

(ii) We should run our chains long enough so that all the $\widehat{R}$'s are small enough.

(iii) We can then combine the *mn* total draws from our chains to produce one chain from the stationary distribution.

Convergence Diagnostics

- **Geweke's Method (based on Geweke, 1992)**
  Suppose we are interested in estimating by MCMC the function $g(\theta)$ of the parameter θ. Geweke proposes using the test statistic

$$\mathbb{T} = \frac{\overline{g}_{n_A} - \overline{g}_{n_B}}{\widehat{\sigma}}$$

where

$$\begin{aligned} \overline{g}_{n_A} &= \frac{1}{n_A} \sum_{t=1}^{n_A} g\left(\theta^{(t)}\right), \overline{g}_{n_B} = \frac{1}{n_B} \sum_{t=n-n_B+1}^{n} g\left(\theta^{(t)}\right) \\ \widehat{\sigma}^2 &= \frac{\widehat{S}_{n_A}(0)}{n_A} + \frac{\widehat{S}_{n_B}(0)}{n_B} \end{aligned}$$

Under the null hypothesis that $\left\{g\left(\theta^{(t)}\right)\right\}$ is strictly stationary and ergodic, we should have

$$\mathbb{T} \xrightarrow{d} N(0,1)$$

as n_A, n_B, and $n \to \infty$ such that $n_A \sim n$ and $n_B \sim n$.