Simulation laboratory 5: Markov chain Monte Carlo methods

Pavel Ilinov

Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne

12 March 2025

Overview

Objective:

- Use Markov chain Monte Carlo (MCMC) methods to draw from complex distributions.

Implementation:

1. Gibbs sampling.
2. Metropolis-Hastings algorithm.

Ideas behind MCMC

- Goal: draw from complex target distribution
- Idea: target distribution is the stationary distribution of the Markov Chain
- Why it what is needed:
  1. once stationarity is reached: draw using transition matrix
  2. state is current draw
  3. by LLN fractions of draws $\approx$ target distribution

- How to construct transition matrix such that it has unique stationary distribution that equals target?

Transition matrix crafting

- Start with simple transition, e.g. symmetric random walk
- To achieve target as stationary: modify probabilities with accept/reject rule

Two variants of MCMC for today:

1. Gibbs: use conditional distribution, every draw is accepted
2. MH: use random walk, modify probabilities with accept/reject rule

Background

Motivation and intuition:

1. Draw from multivariate distributions from which direct sampling is difficult.
2. Construct conditional distributions from which sampling is easy.
3. Iteratively draw from conditional distributions.
4. Suppose we wish to sample $\theta_1, \theta_2 \sim p(\theta_1, \theta_2)$ but cannot do so directly.
5. However, we can sample $\theta_1 \sim p(\theta_1 \mid \theta_2)$ and $\theta_2 \sim p(\theta_2 \mid \theta_1)$.

Background

Algorithm:

1. Set $j = 0$.
2. Provide initial values $\left(\theta_1^{(0)}, \theta_2^{(0)}\right)$.
3. Set $j = j + 1$.
4. $\theta_1^{(j)} \sim p\left(\theta_1^{(j)} \mid \theta_2^{(j-1)}\right)$
5. $\theta_2^{(j)} \sim p\left(\theta_2^{(j)} \mid \theta_1^{(j)}\right)$
6. If j is less than the desired number of draws, return to step 3.

Exercise

Jupyter notebook:

1. Implement a Gibbs sampler to draw from a bivariate normal, i.e.

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \text{ with } \boldsymbol{\mu} = \mathbf{0}, \boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}, \rho = 0.8$$

2. Visually compare the empirical density of the draws to the theoretical density of the sampling distribution.

Note:

If:
$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right)$$

then
$$\theta_1 \mid \theta_2 \sim \mathcal{N}\left(\rho\theta_2, 1 - \rho^2\right)$$
$$\theta_2 \mid \theta_1 \sim \mathcal{N}\left(\rho\theta_1, 1 - \rho^2\right)$$

Background

- What if conditionals are unknown? Gibbs sampling is unfeasible
- Idea again: use stationary distribution of Markov chain
- Key elements
  1. State space: support of target distribution (density)
  2. Transition matrix (discrete case), Proposal density (continious case)
  3. Dynamics: current state $\Rightarrow$ new proposal using proposal density
  4. Decision: accept/reject
  5. Outcome: sampled states is target distribution

Background

Algorithm

1. Set $j = 0$
2. Let θ be current state.
3. Let $p(\theta)$ be target density.
4. Let $J(\theta^* \mid \theta)$ be proposal density
5. Set $j = j + 1$.
6. Propose a new state $\theta^* \mid \theta \sim J(\theta)$.
7. Calculate $\alpha = \frac{p(\theta^*)}{p(\theta)} \cdot \frac{J(\theta|\theta^*)}{J(\theta^*|\theta)}$.
8. Draw $r \sim \mathrm{Uniform}(0, 1)$
9. If $r \leq \alpha$, accept the new state and set $\theta = \theta^*$. Otherwise, reject the new state.
10. If j is less than the desired number of draws, return to step 5.

Convergence assessment

- Run multiple (at least two) Markov chains in *parallel* and initialise each chain with different starting values
- Discard the first half of each simulation sequence for *burn in* to attenuate the impact of adaption to starting values
- Each individual chain must reach *stationarity*.
- Chains must *mix* well
- **Today**: artificial split of single chain into two

Convergence assessment: Potential scale reduction factor $\widehat{R}$

- $\widehat{R}$ can be used to assess convergence for scalar estimands
- $\widehat{R}$ is calculated as a weighted average of the between- B and within W-sequence variances:

$$\widehat{R} = \sqrt{\frac{1}{W}\left(\frac{n-1}{n}W + \frac{1}{n}B\right)}$$

- $\widehat{R} < 1.1$ is acceptable, but $\widehat{R} \approx 1.0$ is preferred

Exercise

Jupyter notebook:

1. Simulate data from Cauchy distribution with location $\mu = 1$ and scale $\gamma = 1$
2. Implement a Metroplis-Hastings algorithms to infer the posterior distribution of the scale parameter of the Cauchy distribution.
3. Consider two possible proposal densities:
    1. $\mathrm{Normal}\left(\theta, \rho^2\right)$, where ρ is the step size.
    2. $\mathrm{Lognormal}\left(\log(\theta) - 0.5\rho^2, \rho\right)$, where ρ is a distance parameter. Note that $\mathrm{Lognormal}(\mu, \sigma)$ denotes a lognormal distribution with location μ and scale σ.
4. Evaluate the performance of the algorithms for different parametrisations of the proposal densities. Compute the potential scale reduction factors $\widehat{R}$

Exercise

Suppose:

- $x_i \sim \mathcal{C}(\mu, \gamma)$ for $i \in \{1, \ldots, N\}$
- $\gamma \sim \text{Gamma}\left(\alpha_0, \beta_0\right)$ with $\alpha_0 = \beta_0 = 0.001$.

Then

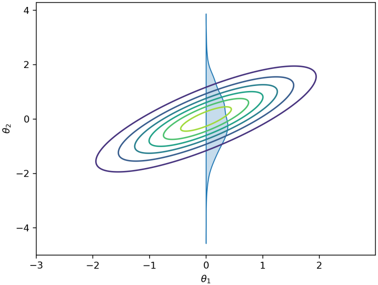
$$p\left(\gamma \mid x, \mu, \alpha_0, \beta_0\right) \propto \left(\prod_i p\left(x_i \mid \mu, \gamma\right)\right) p\left(\gamma \mid \alpha_0, \beta_0\right)$$

Note:

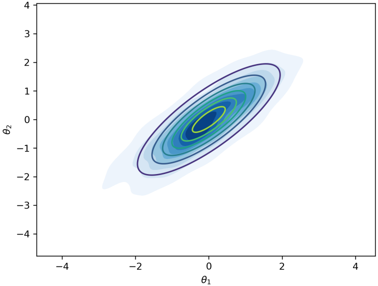
- PDF of Cauchy distribution: $f(x \mid \mu, \gamma) = \frac{1}{\pi\gamma\left[1+\left(\frac{x-\mu}{\gamma}\right)^2\right]}$
- Quantile function of Cauchy distribution: $Q = \mu + \gamma \cdot \tan\left[\pi\left(F - \frac{1}{2}\right)\right]$

Gibbs

Conditional density



Joint density



Metropolis Hastings

Performance of the Normal jumping distribution for the different parameterizations:

ρ	Post. mean	Post. std.	Acceptance r	$\hat{R}$
0.00001	0.100627	0.000265	0.967	2.050453
0.00010	0.129414	0.005499	0.830	2.831171
0.00100	0.600619	0.070020	0.866	3.087170
0.01000	1.038503	0.043489	0.928	1.002975
0.10000	1.028516	0.045275	0.454	1.001987
0.15000	1.030084	0.044200	0.349	0.999190
0.20000	1.024537	0.043835	0.256	1.003908
0.25000	1.033822	0.042426	0.222	1.001908
0.50000	1.033683	0.047204	0.106	1.036941
1.00000	1.037752	0.038093	0.044	1.002151

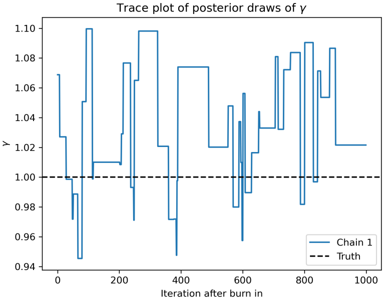
Metropolis Hastings

Performance of the LogNormal jumping distribution for the different parameterizations:

ρ	Post. mean	Post. std.	Acceptance r	$\hat{R}$
0.00001	0.100028	0.000015	0.999	1.673674
0.00010	0.100360	0.000130	0.974	2.598924
0.00100	0.137328	0.007989	0.764	2.953026
0.01000	1.052107	0.034412	0.921	1.006447
0.10000	1.032878	0.044744	0.433	1.001476
0.15000	1.024443	0.048473	0.317	1.002858
0.20000	1.028205	0.043205	0.258	1.012161
0.25000	1.025558	0.043160	0.205	1.026552
0.50000	1.034906	0.044986	0.107	1.010036
1.00000	1.033840	0.052076	0.046	1.003176

Metropolis Hastings

Normal jumping distribution



LogNormal jumping distribution