Markov Chain Monte Carlo

Natalie Packham
Berlin School of Economics & Law

Bayesian Statistics Explorations

26 November 2019

Contents

Overview

- Monte Carlo simulation: directly sample random numbers from a distribution of interest (e.g. normal distribution).
- There are applications – especially those involving Bayesian inference – where this is not possible.
- Idea of Markov Chain Monte Carlo (MCMC): cleverly choose a Markov chain whose stationary distribution corresponds to the distribution of interest.
- By running simulations of this Markov chain for long enough, construct a sample the approximates the distribution of interest.
- Exposition here uses material from (Lemieux, 2009), (Stachurski, 2016), (Fahrmeir et al., 2009).

Contents

Markov chain

- A standard reference is (Norris, 1998).
- Let λ be a distribution on a countable set I (the state-space).
- A matrix $P = (p_{ij} : i, j \in I)$ is stochastic if every row $(p_{ij} : j \in I)$ is a distribution; this is the so-called transition matrix.

Definition

The process $(X_n)_{n\geq 0}$ is a Markov chain with initial distribution λ and transition matrix P if

(i) X_0 has distribution λ, i.e., $\mathbf{P}(X_0 = j) = \lambda_j$;

(ii) for $n \geq 0$, conditional on $X_n = i$, X_{n+1} has distribution $(p_{ij} : j \in I)$;

(iii) $\mathbf{P}(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \ldots, X_n = i) = \mathbf{P}(X_{n+1} = j | X_n = i)$.

Invariant / stationary distribution

Definition

The distribution λ is invariant (stationary) for P if

$$\lambda P = \lambda.$$

Theorem

Let $(X_n)_{n\geq 0}$ be Markov(λ, P). If λ is invariant for P, then $(X_{m+n})_{n\geq 0}$ is also Markov(λ, P).

Convergence to equilibrium

Definition

A Markov chain is irreducible if it is possible to get to any state from any other state.

Definition

The state $i \in I$ is aperiodic if

$$p_{ii}^{(n)} = \mathbf{P}(X_{n+m} = i | X_m = i) > 0, \qquad \text{for sufficiently large } n.$$

A Markov chain is aperiodic if all states are aperiodic.

Theorem

Let P be irreducible and aperiodic, and suppose that P has an invariant distribution π. Let λ be any distribution. Suppose that $(X_n)_{n \geq 0}$ is Markov(λ, P). Then

$$\mathbf{P}(X_n = j) \to \pi_j \text{ as } n \to \infty \text{ for all } j.$$

Ergodic Markov process

Definition

A Markov chain is ergodic if it is aperiodic, irreducible and if it has an invariant distribution.

Ergodic Theorem

- Ergodic Theorems: limiting behaviour of averages over time.
- Example: Strong law of large numbers.

Theorem (Ergodic Theorem)

Let P be irreducible and let λ be any distribution. If $X = (X_n)_{n\geq 0}$ is Markov(λ, P) and if P has an invariant distribution, then, for any bounded function $f : I \to \mathbb{R}$ we have

$$\mathbf{P}\left(\left(\frac{1}{n}\sum_{k=0}^{n-1} f(X_k) \to \overline{f} \text{ as } n \to \infty\right)\right) = 1,$$

where

$$\overline{f} = \sum_{i\in I} \pi_i f_i,$$

and where $\pi = (\pi_i : i \in I)$ is the unique invariant distribution.

Stochastic kernel

- In the following, we use the notion of a stochastic kernel, which describes the transition behaviour of Markov processes in general.
- A transition matrix is a stochastic kernel for a discrete system.
- A transition density describes the transition behaviour of a Markov chain with state space a subset of $\mathbb{R}^d$.

Definition

Let $\mathcal{B}(S)$ be the Borel subsets of $S \subset \mathbb{R}^K$. A stochastic kernel is a function $Q : S \times \mathcal{B}(S) \to [0, 1]$ such that

(i) $Q(\boldsymbol{s}, \cdot)$ is a probability measure on $\mathcal{B}(S)$ for all $\boldsymbol{s} \in S$ and

(ii) $g_B(\boldsymbol{s}) := Q(\boldsymbol{s}, B)$ is $\mathcal{B}$-measurable for each $B \in \mathcal{B}(S)$.

Contents

Markov Chain Monte Carlo

- MCMC is a way of simulating from a given density π on $S \subset \mathbb{R}^d$. The idea is to construct a stochastic kernel P on S such that
  (i) π is a stationary distribution for P,
  (ii) P is sufficiently ergodic that its sample path averages converge to expectations under π.

MCMC: Metropolis-Hastings algorithm, ideas

- The Metropolis-Hastings algorithm starts with a proposal density $q = q(s, s')$.
- Draws from the proposal density are called proposals.
- Each proposal is
  - either accepted by moving to the new state;
  - or rejected by staying at the existing state.
- The probability of accepting is structured so that the chain tends to stay in regions where π puts most probability mass.
- As a consequence, for a sequence (X_t) generated by this process:

$$\text{fraction of time spent in } B = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}_{\{X_t \in B\}} \approx \pi(B) \quad \text{for large } T.$$

MCMC: Metropolis-Hastings algorithm

- Assume that the acceptance probability is $\alpha = \alpha(X_t, Y)$, where
  - X_t is the current state,
  - Y is the proposal.
- The algorithm draws Y from $q(X_t, \cdot)$ and $U \sim U(0,1)$ independently.
- If $U \leq \alpha(X_t, Y)$, then $X_{t+1} = Y$, else $X_{t+1} = X_t$.
- It is easily checked that the stochastic kernel of (X_t) has the form

$$P(s, B) = \int_B p(s, s')\,\mathrm{d}s' + (1 - \lambda(s))\mathbf{1}_{\{s \in B\}},$$

where $p(s, s') := q(s, s')\alpha(s, s')$ and $\lambda(s) := \int p(s, s')\,\mathrm{d}s'$.

MCMC: Metropolis-Hastings algorithm

- In the Metropolis-Hastings algorithm, the acceptance probability function α is defined as

$$\alpha(s, s') := \min\left\{\frac{\pi(s')q(s', s)}{\pi(s)q(s, s')}, 1\right\},$$

with $\alpha(s, s') = 1$ if $\pi(s)q(s, s') = 0$.

Theorem

In the setting above, π is a stationary distribution for P.

- If $q(s, s') = q(s', s)$, e.g. $q(s, s') = \phi(s - s')$ for some function ϕ, sampling from π boils down to being able to calculate density values of π.

Bayesian estimation

- If π is a posterior density in Bayesian estimation,

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})},$$

  with $p(\boldsymbol{x}) = \int(p(x|\boldsymbol{\theta}')p(\boldsymbol{\theta}')\,\mathrm{d}\boldsymbol{\theta}'$, then calculation of the integral drops out due to the ratio in α.

- Here,
  - $p(\boldsymbol{\theta})$ is the prior density,
  - $p(\cdot|\boldsymbol{\theta})$ the likelihood (joint density of the data given $\boldsymbol{\theta}$)
  - $p(\cdot|\boldsymbol{x})$ is the posterior density.

Contents

MCMC example

- This follows the example from Section B.5 of (Fahrmeir et al., 2009).
- Let $Y_1, \dots, Y_n$ be a sample of independent, Poisson distributed random variables with parameter λ.
- λ is unknown and to be estimated using Bayesian inference and MCMC.
- The joint distribution of the sample $\boldsymbol{y} = (y_1, \dots, y_n)$ is

$$p(\boldsymbol{y}|\lambda) = \prod_{i=1}^{n} f(y_i; \lambda),$$

where $f(\cdot; \lambda)$ denotes the Poisson probability function with parameter λ.

MCMC example

- Choice of prior: The Gamma distribution is a conjugate prior
  - (i.e., prior and posterior have the same distribution type),

  so we choose $\lambda \sim G(a, b)$.
- Other prior distributions (e.g. uniform, normal, ...) give good results, too.
- The posterior is
$$p(\lambda|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\lambda)p(\lambda)}{\int p(\boldsymbol{y}|\lambda)p(\lambda)\,\mathrm{d}\lambda}.$$
- Using MCMC, the denominator, which is a constant that does not depend on λ, drops out.
- The posterior density is
$$\boldsymbol{y}|\lambda \sim G\left(a + \sum_{i=1}^{n} y_i, b + n\right).$$

MCMC example

- Drawing the sample and setting up likelihood and prior functions:

```
>>> import numpy as np
>>> import scipy as sp
>>> import scipy.stats as scs
>>> import matplotlib.pyplot as plt
>>> plt.style.use('seaborn')  # sets the plotting style
>>> np.random.seed(583920)

>>> a=0.1
>>> y = scs.poisson.rvs(1, size=125) # sample data
>>> y.mean()
0.984

>>> def l(lamb): # likelihood
...     return scs.poisson.pmf(y, lamb).prod()
...
>>> def prior(lamb):
...     return scs.gamma.pdf(lamb, a) # gamma prior
```

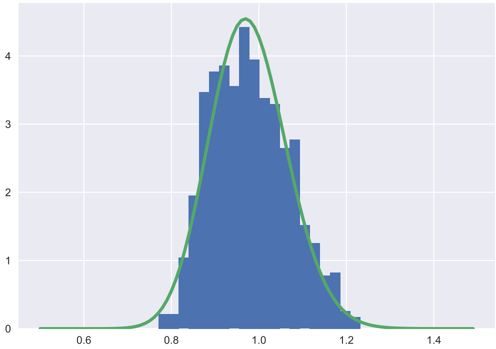
MCMC example

- Running the MCMC algorithm and plotting the posterior density:

```
...
>>> T = 1000
>>> sigma = 0.05
>>> x = np.zeros(T)
>>> z = scs.norm.rvs(size=T) # simulate proposal from normal proposal density
>>> u = scs.uniform.rvs(size=T)
>>> x[0] = 1
>>> for t in range(1,T):
...     s = x[t-1] + sigma * z[t-1] # proposal
...     alpha = l(s) * prior(s) / (l(x[t-1]) * prior(x[t-1])) \
...             if (l(x[t-1]) * prior(x[t-1])) > 0 else 1
...     x[t] = s if u[t] <= alpha else x[t-1]
...
>>> u = np.arange(0.5,1.5,0.01)
>>> _ = plt.hist(x, bins=20, density=True);
>>> alpha = a + y.sum()
>>> beta = 1 + len(y)
>>> _ = plt.plot(u, scs.gamma.pdf(u, alpha) * sp.exp(u * (1-beta)) \
...     * beta**alpha, linewidth=3);
>>> plt.savefig('mcmc_pic.pdf')
>>> print(x.mean())
0.9806439285842388
```

MCMC example

- Output produced:



References

Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. *Regression*. Springer, 2nd edition, 2009.

Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer, 2009.

J. R. Norris. *Markov Chains*. Cambridge University Press, 1998.

John Stachurski. *A Primer in Econometric Theory*. MIT Press, 2016.

Thank you!

Prof. Dr. Natalie Packham
Professor of Mathematics and Statistics
Berlin School of Economics and Law
Badensche Str. 52
10825 Berlin

Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law