Bayesian Learning - Lecture 9

Mattias Villani

Division of Statistics
Department of Computer and Information Science
Linköping University

Lecture overview

- Markov Chain Monte Carlo
- Metropolis-Hastings

Markov chains

- **Markov chain**

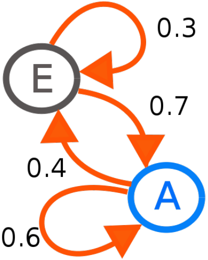
$$\Pr(X_{t+1} = x | X_t = x_t, ..., X_1 = x_1) = \Pr(X_{t+1} = x | X_t = x_t)$$

- Markov chain with two states: i and j. **Transition probabilities**:

$$p_{ij} = \Pr(X_{t+1} = j | X_t = i)$$

- Example

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{pmatrix}$$



Stationary distribution

- Initial probabilities: $\alpha_0 = Pr(X_0 = x)$.
- Marginal distribution of the chain at time t

$$\alpha_0 P^t$$

- Stationary distribution

$$\pi = \pi P$$

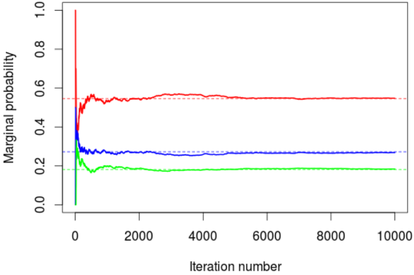
$$P^t \to \begin{pmatrix} \pi \\ \pi \\ \vdots \\ \pi \end{pmatrix}$$

- [π is the normalized left eigenvector corresponding to the eigenvalue 1]
- Example:

$$P = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}$$

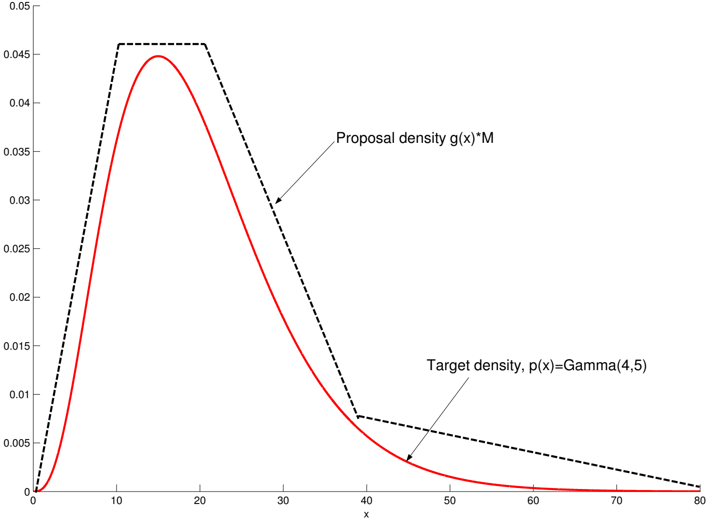
$$\pi = (0.545, 0.272, 0.181)$$

Simulating the stationary distribution



- Suppose we want to simulate from a discrete distribution $p(x)$ for $x \in \{s_1, s_2, ..., s_k\}$.
- Basic idea of MCMC: simulate from a Markov chain with a stationary distribution that is exactly $p(x)$.
- How to set up the transition matrix P? Metropolis-Hastings.

Rejection sampling



The Metropolis Algorithm

- Initialize with $\theta = \theta_0$
- For $t = 1, 2, ...$
  - Sample a proposal draw $\theta^* | \theta^{(t-1)} \sim J_t(\theta^*, \theta^{(t-1)})$
  - Accept θ^* with probability

$$r(\theta^*, \theta^{(t-1)}) = \min\left[\frac{p(\theta^*|y)}{p(\theta^{(t-1)}|y)}, 1\right].$$

  - If the proposal is accepted, set $\theta^{(t)} = \theta^*$, otherwise set $\theta^{(t)} = \theta^{(t-1)}$.

Metropolis algorithm, cont.

- We must be able to compute the posterior density $p(\theta|y)$ for any θ.
- The Metropolis algorithm works even if $p(\theta|y)$ is only known up to a proportionality constant as it simply cancels in $r(\theta^*, \theta^{(t-1)})$.
- The proposal, or jumping, distribution $J_t(\theta^*|\theta^{(t-1)})$ may vary from iteration to iteration.
- $J_t(\theta^*, \theta^{(t-1)})$ must be symmetric, i.e.

$$J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a) \text{ for all } \theta_a, \theta_b \text{ and } t.$$

- Every proposal that θ^* that lies uphill $(p(\theta^*|y) \geq p(\theta^{(t-1)}|y))$ is accepted with certainty. Downhill moves accepted with prob. $r(\theta^*, \theta^{(t-1)})$.

Metropolis - Choosing the proposal distribution

- Common choice of proposal distribution:

$$J_t(\theta^*|\theta^{(t-1)}) = N\left(\theta^{(t-1)}, \Sigma\right)$$

  where $\Sigma = c^2 \cdot J^{-1}_{\tilde{\theta},\mathbf{x}}$ and $\cdot J_{\tilde{\theta},\mathbf{x}}$ is the observed information matrix at the posterior mode (numerical optimization).
- c is a tuning constant set so that average acceptance probability is something like 0.3 (see Section 11.9).
- A good proposal $J_t(\theta^*|\theta^{(t-1)})$ should have the following properties
  - Easy to sample
  - Easy to compute $r(\theta^*, \theta^{(t-1)})$
  - Takes reasonably large jumps in the parameter space
  - The jumps are not rejected too frequently.

Practical Implementation of MCMC Algorithms

- The autocorrelation in the simulated sequence $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}$ makes it somewhat problematic to define the effective number of simulation draws.
- Inefficiency factor:
$$\mathrm{IF} = 1 + 2\sum_{i=1}^{\infty} \rho_i,$$
where ρ_i is the autocorrelation at lag i.
- Effective sample size:
$$\mathrm{ESS} = \mathrm{N}/\mathrm{IF}.$$
- When do we stop sampling?
- How many *burn-in* iterations to discard?
- Several short sequences or a single long sequence? To thin out or not to thin out?
- Software issues.

Convergence diagnostics

- Raw plots of the simulated sequences (trajectories)
- CUSUM plots (+ Local)
- Anova-type tests. After convergence, it should not matter if we compute the marginal posterior variance of from:
  1. one big posterior sample which merges all the m parallel sequences together
  2. each of the parallel sequences separately and then average the m estimates.
- Potential scale reduction factor:

$$R = \frac{\text{Variance under setting 1}}{\text{Variance under setting 2}}$$

$R \downarrow 1$ as $N \to \infty$.

The Metropolis-Hastings algorithm

- Generalization of the Metropolis algorithm to non-symmetric proposals.
- The acceptance probability is slightly more complicated

$$r(\theta^*, \theta^{(t-1)}) = \min\left[\frac{p(\theta^*|y)\,/\,J_t(\theta^*|\theta^{(t-1)})}{p(\theta^{(t-1)}|y)\,/\,J_t(\theta^{(t-1)}|\theta^*)}, 1\right].$$

- Gibbs sampling is a special case of the MH algorithm where the proposal is the full conditional posterior and $r(\theta^*, \theta^{(t-1)}) = 1$ for any $(\theta^*, \theta^{(t-1)})$ pair.
- Metropolis-Hastings-within-Gibbs hybrid algorithms.