Bayesian Learning - Lecture 8

Mattias Villani

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University

Lecture overview

- Markov Chain Monte Carlo - the general idea
- Metropolis-Hastings
- MCMC in practice

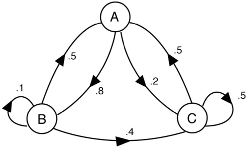
Markov chains

- Let $\mathcal{S} = \{s_1, s_2, ..., s_k)$ be a finite set of **states**.
  - Weather: $\mathcal{S} = \{\text{sunny}, \ \text{rain}\}$.
  - Journal rankings: $\mathcal{S} = \{A+, A, B, C, D, E\}$
- **Markov chain** is a stochastic process $\{X_t\}_{t=1}^{T}$ with random **state transitions**

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

- Example realization journal ranking:
  $X_1 = C, X_2 = C, X_3 = B, X_4 = A+, X_5 = B$.
- **Transition matrix** for weather example

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix} = \begin{pmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{pmatrix}$$



Stationary distribution

- ***h*-step transition probabilities**

$$P_{ij}^{(h)} = \Pr(X_{t+h} = s_j | X_t = s_i)$$

- ***h*-step transition matrix**

$$P^{(h)} = P^h$$

- The chain has a **unique equilibrium stationary distribution** $\pi = (\pi_1, ..., \pi_k)$ if it is
  - **irreducible** (possible to get from any state from any state)
  - **aperiodic** (does not get stuck in predictable cycles)
  - **positive recurrent** (expected time of returning to any state is finite)
- Limiting (long-run) distribution

$$P^t \to \begin{pmatrix} \pi \\ \pi \\ \vdots \\ \pi \end{pmatrix} = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_k \\ \pi_1 & \pi_2 & \cdots & \pi_k \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_k \end{pmatrix} \text{ as } t \to \infty$$

Stationary distribution, cont.

- Limiting (long-run) distribution

$$P^t \rightarrow \begin{pmatrix} \pi \\ \pi \\ \vdots \\ \pi \end{pmatrix} = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_k \\ \pi_1 & \pi_2 & \cdots & \pi_k \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_k \end{pmatrix} \text{ as } t \rightarrow \infty$$

- **Stationary distribution**

$$\pi = \pi P$$

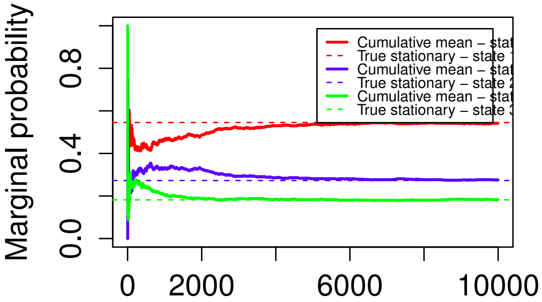
- Example:

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

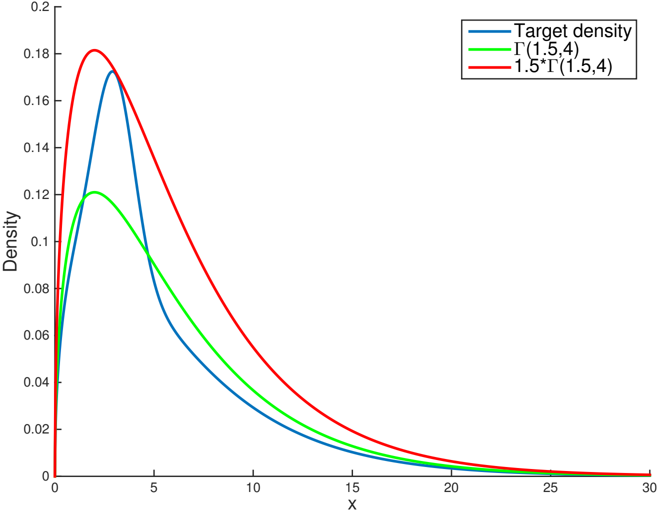
$$\pi = (0.545, 0.272, 0.181)$$

The basic MCMC idea

- Aim: to simulate from a discrete distribution $p(x)$ when $x \in \{s_1, s_2, ..., s_k\}$.
- **MCMC**: **simulate a Markov Chain** with a **stationary distribution** that is exactly $p(x)$.
- How to set up the transition matrix P? **Metropolis-Hastings**!



Rejection sampling



Random walk Metropolis algorithm

- Initialize $\theta^{(0)}$ and iterate for $i = 1, 2, \ldots$

  1. Sample $\theta_p | \theta^{(i-1)} \sim N\left(\theta^{(i-1)}, c \cdot \Sigma\right)$ (the **proposal distribution**)
  2. Compute the **acceptance probability**

  $$\alpha = \min\left(1, \frac{p(\theta_p|\mathbf{y})}{p(\theta^{(i-1)}|\mathbf{y})}\right)$$

  3. With probability α set $\theta^{(i)} = \theta_p$ and $\theta^{(i)} = \theta^{(i-1)}$ otherwise.

Random walk Metropolis, cont.

- Assumption: we can compute $p(\theta_p|\mathbf{y})$ for any θ.
- Proportionality constant in $p(\theta_p|\mathbf{y})$ does not matter. It will cancel in α

$$\alpha = \min\left(1, \frac{c \cdot p(\theta_p|\mathbf{y})}{c \cdot p(\theta^{(i-1)}|\mathbf{y})}\right) = \min\left(1, \frac{p(\theta_p|\mathbf{y})}{p(\theta^{(i-1)}|\mathbf{y})}\right)$$

- So we many use tattoo-version: $p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)$

$$\alpha = \min\left(1, \frac{p\left(\mathbf{y}|\theta_p\right) p\left(\theta_p\right)}{p\left(\mathbf{y}|\theta^{(i-1)}\right) p\left(\theta^{(i-1)}\right)}\right)$$

- We can generalize the proposal $\theta_p|\theta^{(i-1)} \sim N\left(\theta^{(i-1)}, c \cdot \Sigma\right)$ to

$$\theta_p|\theta^{(i-1)} \sim q\left(\cdot|\theta^{(i-1)}\right)$$

where $q\left(\cdot|\theta^{(i-1)}\right)$ is symmetric is its arguments

$$q\left(y|x\right) = q\left(x|y\right)$$

Random walk Metropolis, cont.

- Common choices of Σ in proposal $N\left(\theta^{(i-1)}, c \cdot \Sigma\right)$:
  - $\Sigma = I$ (may propose 'off the cigar')
  - $\Sigma = J_{\hat{\theta},\mathbf{y}}^{-1}$ (propose 'along the cigar')
  - Adaptive. Start with $\Sigma = I$ and then recompute Σ from an initial simulation run.
- c is set so that average acceptance probability is roughly 25-30%.
- A good proposal:
  - **Easy to sample**
  - **Easy to compute** α
  - Proposals should take reasonably **large steps** in θ-space
  - Proposals should **not be reject too often**.

The Metropolis-Hastings algorithm

- Generalization when the proposal density is not symmetric.

- Initialize $\theta^{(0)}$ and iterate for $i = 1, 2, \ldots$

  1. Sample $\theta_p \sim q\left(\cdot|\theta^{(i-1)}\right)$ (the **proposal distribution**)
  2. Compute the **acceptance probability**
  $$\alpha = \min\left(1, \frac{p(\mathbf{y}|\theta_p)p(\theta_p)}{p(\mathbf{y}|\theta^{(i-1)})p(\theta^{(i-1)})}\frac{q\left(\theta^{(i-1)}|\theta_p\right)}{q\left(\theta_p|\theta^{(i-1)}\right)}\right)$$
  3. With probability α set $\theta^{(i)} = \theta_p$ and $\theta^{(i)} = \theta^{(i-1)}$ otherwise.

The independence sampler

- **Independence sampler**: $q\left(\theta_p|\theta^{(i-1)}\right) = q\left(\theta_p\right)$.
- Proposal is independent of previous draw.
- Example:

$$\theta_p \sim t_v\left(\hat{\theta}, J^{-1}_{\hat{\theta},\mathbf{y}}\right),$$

  where $\hat{\theta}$ and , $J_{\hat{\theta},\mathbf{y}}$ are computed by numerical optimization.
- Can be very **efficient**, but has a tendency to **get stuck**.
- Make sure that $q\left(\theta_p\right)$ has **heavier tails** than $p(\theta|\mathbf{y})$.

The efficiency of MCMC

- $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}$ are **dependent** (autocorrelated).
- How efficient is my MCMC compared to iid sampling?
- If $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}$ are iid with variance σ^2, then

$$\mathrm{Var}(\bar{\theta}) = \frac{\sigma^2}{N}.$$

- If $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}$ are generated by MCMC

$$\mathrm{Var}(\bar{\theta}) = \frac{\sigma^2}{N}\left(1 + 2\sum_{k=1}^{\infty}\rho_k\right)$$

  where $\rho_k = Corr(\theta^{(i)}, \theta^{(i+k)})$ is the autocorrelation at lag k.
- **Inefficiency factor**

$$\mathrm{IF} = 1 + 2\sum_{k=1}^{\infty}\rho_k$$

- **Effective sample size** from MCMC

$$\mathrm{ESS} = N/\mathrm{IF}$$

Burn-in and convergence

- **How long** **burn-in**?
- **How long to sample** after burn-in?
- To **thin** of not to thin? Only keeping every h draw reduces autocorrelation.
- **Convergence diagnostics**
  - Raw plots of simulated sequences (trajectories)
  - CUSUM plots + Local means
  - Potential scale reduction factor, R.