Bayesian Learning

Lecture 8 - Markov Chain Monte Carlo and Metropolis-Hastings

Bertil Wegmann

Department of Computer and Information Science
Linköping University

LINKÖPING UNIVERSITY

Lecture overview

- Markov Chain Monte Carlo
- Metropolis-Hastings
- MCMC - efficiency, burn-in and convergence

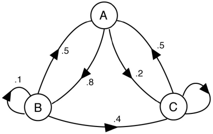
Markov chains

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

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

- School grades: $X_1 = C, X_2 = C, X_3 = B, X_4 = A, X_5 = B$.
- **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 by matrix power

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

- **Unique** equilibrium distribution $\pi = (\pi_1, ..., \pi_k)$ if chain is
  - irreducible (possible to get to any state from any state)
  - aperiodic (does not get stuck in predictable cycles)
  - positive recurrent (expected time of returning 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

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

- **Stationary distribution**

$$\pi = \pi P$$

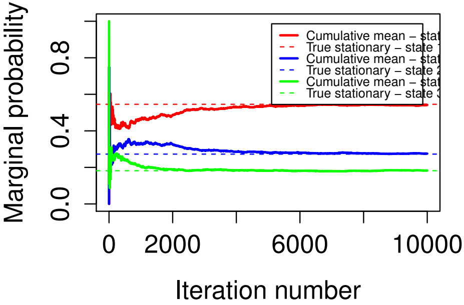
- Example:

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

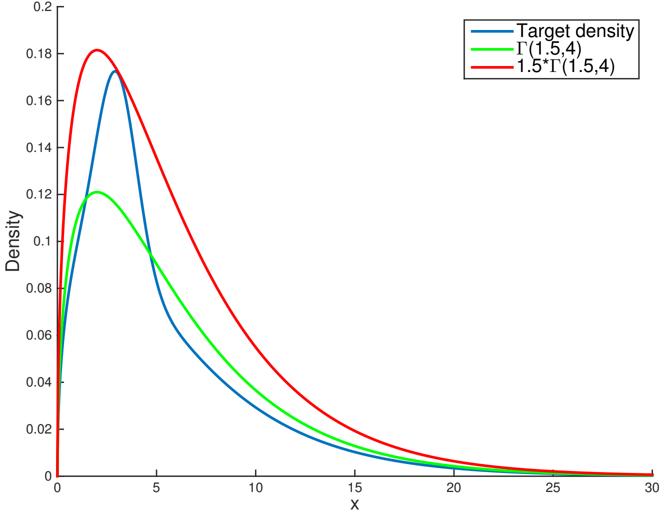
$$\pi = (0.545, 0.272, 0.181)$$

The basic MCMC idea

- Simulate from discrete distribution $p(x)$ when $x \in \{s_1, .., s_k\}$.
- **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 proposal**: $\theta_p | \theta^{(i-1)} \sim N\left(\theta^{(i-1)}, c \cdot \Sigma\right)$
  2. Compute the **acceptance probability**

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

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

Random walk Metropolis, cont.

- Assumption: we can compute $p(\theta_p|\mathrm{y})$ for any θ.

- Proportionality constants in posterior cancel out in

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

- In particular:

$$\frac{p(\theta_p|\mathrm{y})}{p(\theta^{(i-1)}|\mathrm{y})} = \frac{p(\mathrm{y}|\theta_p)p(\theta_p)/p(y)}{p(\mathrm{y}|\theta^{(i-1)})p(\theta^{(i-1)})/p(y)} = \frac{p(\mathrm{y}|\theta_p)p(\theta_p)}{p(\mathrm{y}|\theta^{(i-1)})p(\theta^{(i-1)})}$$

- **Proportional form of posterior is enough!**

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

Random walk Metropolis, cont.

- Common choices of Σ in proposal $N\left(\theta^{(i-1)}, c \cdot \Sigma\right)$:
  - $\Sigma = I$ (proposes 'off the cigar')
  - $\Sigma = J^{-1}_{\hat{\theta},\mathbf{y}}$ (propose 'along the cigar')
  - **Adaptive**. Start with $\Sigma = I$. Update Σ from initial run.

- Set c so average acceptance probability is 25-30%.

- **Good proposal**:
  - **Easy to sample**
  - **Easy to compute** α
  - Proposals should take reasonably **large steps** in θ-space
  - Proposals should **not be rejected 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 proposal**: $\theta_p \sim q\left(\cdot|\theta^{(i-1)}\right)$
  2. Compute the **acceptance probability**
  $$\alpha = \min\left(1, \frac{p(\mathrm{y}|\theta_p)p(\theta_p)}{p(\mathrm{y}|\theta^{(i-1)})p(\theta^{(i-1)})}\frac{q\left(\theta^{(i-1)}|\theta_p\right)}{q\left(\theta_p|\theta^{(i-1)}\right)}\right)$$
  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:
$$\theta_p \sim t_v\left(\hat{\theta}, J^{-1}_{\hat{\theta},\mathrm{y}}\right),$$
where $\hat{\theta}$ and , $J_{\hat{\theta},\mathrm{y}}$ 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 $p(\theta|\mathrm{y})$.

Metropolis-Hastings within Gibbs

- Gibbs sampling from $p(\theta_1, \theta_2, \theta_3 | \mathrm{y})$
  - Sample $p(\theta_1 | \theta_2, \theta_3, \mathrm{y})$
  - Sample $p(\theta_2 | \theta_1, \theta_3, \mathrm{y})$
  - Sample $p(\theta_3 | \theta_1, \theta_2, \mathrm{y})$
- When a full conditional is not easily sampled we can simulate from it using MH.
- Example: at ith iteration, propose θ_2 from $q(\theta_2 | \theta_1, \theta_3, \theta_2^{(i-1)}, \mathrm{y})$. Accept/reject.
- Gibbs sampling is a special case of MH when $q(\theta_2 | \theta_1, \theta_3, \theta_2^{(i-1)}, \mathrm{y}) = p(\theta_2 | \theta_1, \theta_3, \mathrm{y})$, which gives $\alpha = 1$. Always accept.

The efficiency of MCMC

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

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

- Autocorrelated $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(N)}$ 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**

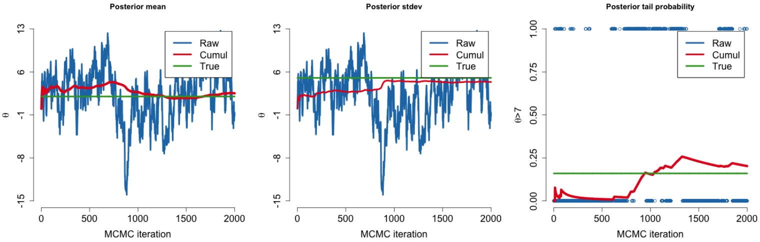
$$\mathrm{IF} = 1 + 2\sum_{k=1}^{\infty}\rho_k$$

- **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?
- **Thinning**? Keeping every h draw reduces autocorrelation.
- **Convergence diagnostics**
  - Raw plots of simulated sequences (trajectories)
  - CUSUM plots
  - ESS
  - Potential scale reduction factor, $\hat{R}$.



Burn-in and convergence