Bayesian Learning

Lecture 8 - Markov Chain Monte Carlo. Metropolis-Hastings.

Mattias Villani

Department of Statistics
Stockholm University

mattiasvillani.com @matvil @matvil mattiasvillani

Lecture overview

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

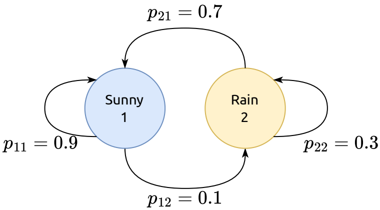
Stochastic process

- Let $\mathcal{S} = \{1, 2, \ldots, K\}$ be a finite set of **states**.
  - Weather: $\mathcal{S} = \{\text{sunny}, \text{ rain}\}$.
  - School grades: $\mathcal{S} = \{A, B, C, D, E, F\}$
- **Stochastic process**: collection of random variables $X_1, X_2, X_3, \ldots$, often over time.
- A **time series** with categories.
- Weather: $X_1 = \text{sunny}$, $X_2 = \text{sunny}$, $X_3 = \text{rain}$, $X_4 = \text{sunny}$.
- School grades: $X_1 = C$, $X_2 = C$, $X_3 = B$, $X_4 = A$, $X_5 = B$.

Markov chains

- **Markov chain**: probability distribution of tomorrow's state X_{t+1} depends (only) on today's state X_t:

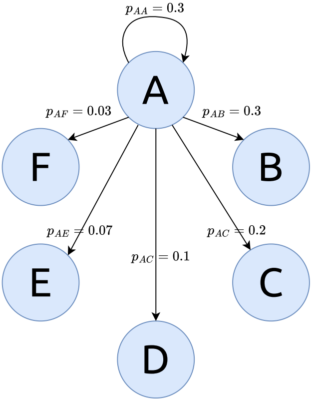
$$p_{ij} = \Pr(X_{t+1} = j | X_t = i)$$



- **Transition matrix** for weather example

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

Markov chains - Grades example



$$P = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0.3 & 0.3 & 0.2 & 0.1 & 0.07 & 0.03 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \end{pmatrix} \end{array}$$

Stationary distribution

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

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

- *h*-**step transition matrix** by **matrix power**

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

- **Unique equilibrium distribution** $\boldsymbol{\pi} = (\pi_1, ..., \pi_k)$ if chain is **ergodic**:
  - **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** as $h \to \infty$

$$\boldsymbol{P}^h \to \begin{pmatrix} \boldsymbol{\pi} \\ \boldsymbol{\pi} \\ \vdots \\ \boldsymbol{\pi} \end{pmatrix} = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_k \\ \pi_1 & \pi_2 & \cdots & \pi_k \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_k \end{pmatrix}$$

Stationary distribution, cont.

- Limiting **long-run distribution** as $h \to \infty$

$$\boldsymbol{P}^h \to \begin{pmatrix} \boldsymbol{\pi} \\ \boldsymbol{\pi} \\ \vdots \\ \boldsymbol{\pi} \end{pmatrix} = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_k \\ \pi_1 & \pi_2 & \cdots & \pi_k \\ \vdots & \vdots & & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_k \end{pmatrix}$$

- **Stationary distribution**

$$\boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{P}$$

- Draw starting point using $\boldsymbol{\pi}$. All future states of the Markov Chain will have distribution $\boldsymbol{\pi}$.

- Weather example:

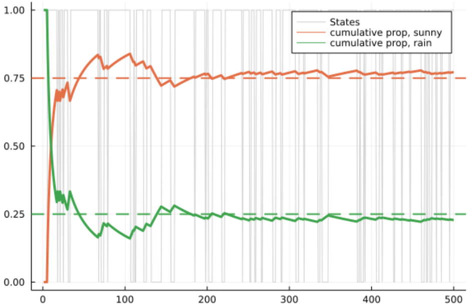
$$\boldsymbol{P} = \begin{pmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{pmatrix}, \boldsymbol{P}^2 = \begin{pmatrix} 0.84 & 0.16 \\ 0.42 & 0.58 \end{pmatrix}$$

$$\boldsymbol{P}^5 = \begin{pmatrix} 0.77 & 0.23 \\ 0.69 & 0.31 \end{pmatrix}, \boldsymbol{P}^{100} = \begin{pmatrix} 0.75 & 0.25 \\ 0.75 & 0.25 \end{pmatrix}$$

$$\boldsymbol{\pi} = (0.75, 0.25)$$

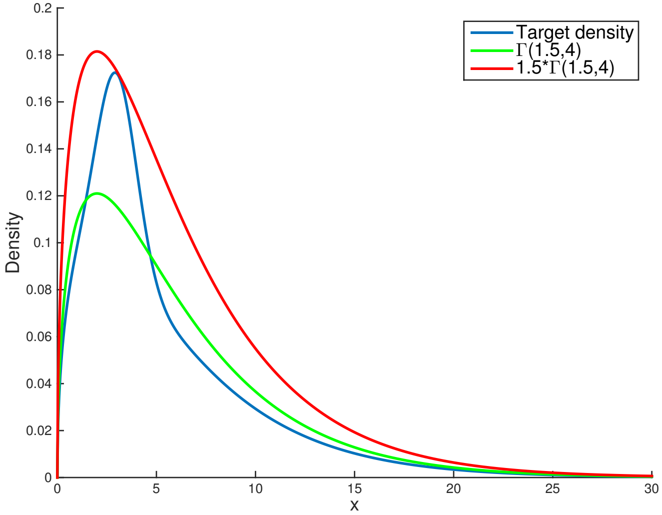
The basic MCMC idea

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



- Can be extended to **continuous random variables**.

Rejection sampling



Random walk Metropolis algorithm

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

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

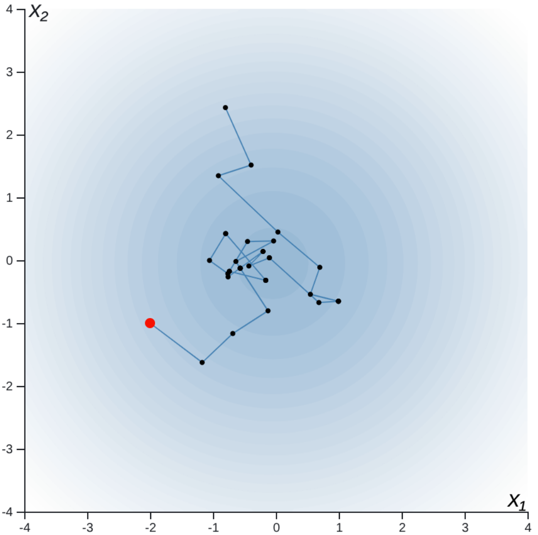
  $$\alpha = \min\left(1, \frac{p(\boldsymbol{\theta}^{\star}|\mathbf{y})}{p(\boldsymbol{\theta}^{(i-1)}|\mathbf{y})}\right)$$

  3. Draw $u \sim \mathrm{Uniform}(0,1)$

     If $u \leq \alpha$ set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{\star}$ (accept and move)

     If $u > \alpha$ set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$ (reject and stay)

Interactive - Random Walk Metropolis



Proportional form of posterior is enough!

- Assumption: we can compute $p(\boldsymbol{\theta}|\mathbf{y})$ for any $\boldsymbol{\theta}$.
- Proportionality constants in posterior cancel out in

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

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

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

Random walk Metropolis - choosing a proposal

- Common choices of $\boldsymbol{\Sigma}$ in proposal $N\left(\boldsymbol{\theta}^{(i-1)}, c \cdot \boldsymbol{\Sigma}\right)$:
  - $\boldsymbol{\Sigma} = \boldsymbol{I}$ (proposes 'off the cigar')
  - $\boldsymbol{\Sigma} = J_{\mathbf{y}}^{-1}(\tilde{\boldsymbol{\theta}})$ (propose 'along the cigar')
  - **Adaptive**. Start with $\boldsymbol{\Sigma} = \boldsymbol{I}$. Update $\boldsymbol{\Sigma}$ 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 $\boldsymbol{\theta}$-space
  - Proposals should **not be reject too often**.

The Metropolis-Hastings algorithm

- Generalization when the proposal density is not symmetric.

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

  1. **Sample proposal**: $\boldsymbol{\theta}^\star \sim q\left(\cdot|\boldsymbol{\theta}^{(i-1)}\right)$
  2. Compute the **acceptance probability**

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

  3. Draw $u \sim \mathrm{Uniform}(0, 1)$
     If $u \leq \alpha$ set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^\star$ (accept and move)
     If $u > \alpha$ set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$ (reject and stay)

The independence sampler

- **Independence sampler**: $q\left(\boldsymbol{\theta}^\star|\boldsymbol{\theta}^{(i-1)}\right) = q\left(\boldsymbol{\theta}^\star\right)$.
- **Proposal** is **independent of previous draw**.
- Example: a multivariate-t distribution (we want heavy tails)

$$\boldsymbol{\theta}^\star \sim t\left(\tilde{\boldsymbol{\theta}}, J_{\mathbf{y}}^{-1}(\tilde{\boldsymbol{\theta}}), \nu\right),$$

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

The efficiency of MCMC

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

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

- Autocorrelated $\theta^{(1)}, \theta^{(2)}, ..., \theta^{(m)}$ generated by **MCMC**

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

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

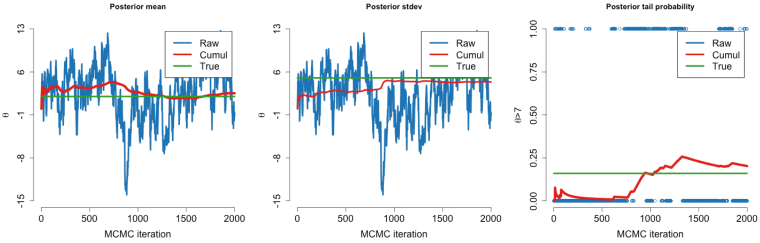
$$\mathrm{IF} = 1 + 2\sum_{k=1}^{K}\rho_k$$

- **Effective sample size** from MCMC

$$\mathrm{ESS} = \frac{m}{\mathrm{IF}}$$

Burn-in and convergence

- How long **burn-in**?
- **How long to sample** after burn-in?
- **Convergence diagnostics**
  - Raw plots of simulated sequences (**MCMC trajectories**)
  - **Cumulative estimates** plots
  - **Repeated runs** with different initial values
  - Potential scale reduction factor, R.



Burn-in and convergence