Markov chain Monte Carlo

Geir Storvik

Geilo Winter school 2023

UiO : Universitetet i Oslo

Norsk Regnesentral
NORWEGIAN COMPUTING CENTER

Outline

1. Theoretical properties
2. Hamiltonian MC
3. Pseudo-marginal MCMC
4. Reversible jump MCMC
5. Additional topics

Theoretical properties

Convergence issues of MCMC

- Theoretical properties:

$$\boldsymbol{x}^{(t)} \xrightarrow{\mathcal{D}} \pi(\boldsymbol{x}), \quad \text{as } t \to \infty$$

$$\hat{\theta}_1 = \frac{1}{L} \sum_{t=1}^{L} h(\boldsymbol{x}^{(t)}) \to E^p[h(\boldsymbol{x})] \quad \text{as } L \to \infty$$

Convergence issues of MCMC

- Theoretical properties:

$$\boldsymbol{x}^{(t)} \xrightarrow{\mathcal{D}} \pi(\boldsymbol{x}), \quad \text{as } t \to \infty$$

$$\hat{\theta}_1 = \frac{1}{L}\sum_{t=1}^{L} h(\boldsymbol{x}^{(t)}) \to E^p[h(\boldsymbol{x})] \quad \text{as } L \to \infty$$

- Note: We also have

$$\hat{\theta}_2 = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)}) \to E^p[h(\boldsymbol{x})] \quad \text{as } L \to \infty$$

  - Advantage: Remove those variables with distribution very different from $\pi(\boldsymbol{x})$
  - Disadvantage: Need more samples
- Question: How to specify D and L?
  - D: Large enough so that $\boldsymbol{x}^{(t)} \approx \pi(\boldsymbol{x})$ for $t > D$ (bias small)
  - L: Large enough so that $\text{Var}[\hat{\theta}_2]$ is small enough

Central limit theorems

- Under additional requirements (see e.g Robert and Casella (1999, ch 6)) we have

$$\sqrt{L}(\hat{\theta} - \theta) \approx N(0, \gamma^2)$$

- Essential requirement:

$$\gamma^2 = \mathrm{Var}_\pi[h(X_0) + 2\sum_{k=1}^{\infty} \mathrm{Cov}_\pi[h(X_0), h(X_k)] < \infty$$

- Interpretation: Correlations should decay fast enough

Mixing

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\mathrm{Var}[\hat{\theta}] = \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \mathrm{Var}[h(\boldsymbol{x}^{(t)})] + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})]\right]$$

Mixing

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\mathrm{Var}[\hat{\theta}] = \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \mathrm{Var}[h(\boldsymbol{x}^{(t)})] + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})]\right]$$

Assume D large, so "converged":

$$\mathrm{Var}[h(\boldsymbol{x}^{(t)})] \approx \sigma_h^2, \quad \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})] \approx \sigma_h^2 \rho(t-s)$$

Mixing

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\mathrm{Var}[\hat{\theta}] = \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \mathrm{Var}[h(\boldsymbol{x}^{(t)})] + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})]\right]$$

Assume D large, so "converged":

$$\mathrm{Var}[h(\boldsymbol{x}^{(t)})] \approx \sigma_h^2, \quad \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})] \approx \sigma_h^2 \rho(t-s)$$

gives

$$\begin{aligned}\mathrm{Var}[\hat{\theta}] &\approx \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \sigma_h^2 + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \sigma_h^2 \rho(t-s)\right] \\ &= \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1} \frac{L-k}{L}\rho(k)\right]\end{aligned}$$

Mixing

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\mathrm{Var}[\hat{\theta}] = \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \mathrm{Var}[h(\boldsymbol{x}^{(t)})] + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})]\right]$$

  Assume D large, so "converged":

$$\mathrm{Var}[h(\boldsymbol{x}^{(t)})] \approx \sigma_h^2, \quad \mathrm{Cov}[h(\boldsymbol{x}^{(s)}), h(\boldsymbol{x}^{(t)})] \approx \sigma_h^2 \rho(t-s)$$

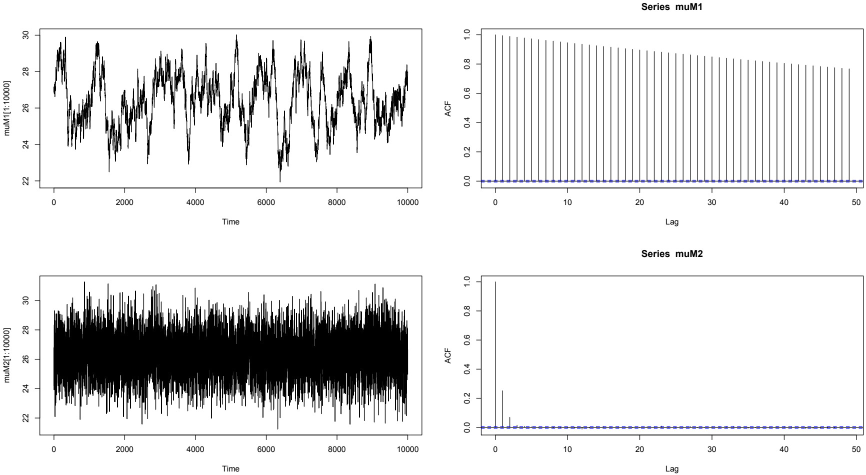
  gives

$$\mathrm{Var}[\hat{\theta}] \approx \frac{1}{L^2}\left[\sum_{t=D+1}^{D+L} \sigma_h^2 + 2\sum_{s=D+1}^{D+L-1}\sum_{t=s+1}^{D+L} \sigma_h^2\rho(t-s)\right]$$

$$= \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1}\frac{L-k}{L}\rho(k)\right]$$

- Good mixing: $\rho(k)$ decreases fast with k!

Example


Series muM1
Series muM2
muM1[1:10000]
muM2[1:10000]
Time
ACF
Lag

How to assess convergence?

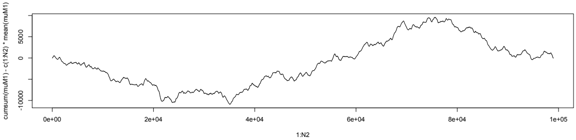
- Graphical diagnostics:
  - Sample paths:
    - Plot $h(\boldsymbol{x}^{(t)})$ as function of t
    - Useful with different $h(\cdot)$ functions!

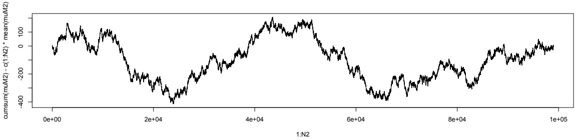
How to assess convergence?

- Graphical diagnostics:
  - Sample paths:
    - Plot $h(\boldsymbol{x}^{(t)})$ as function of t
    - Useful with different $h(\cdot)$ functions!
  - Cusum diagnostics
    - Plot $\sum_{i=1}^{t}[h(\boldsymbol{x}^{(i)}) - \hat{\theta}_n]$ versus t
    - Wiggly and small excursions from 0: Indicate chain is mixing well

How to assess convergence?

- Graphical diagnostics:
  - Sample paths:
    - Plot $h(\boldsymbol{x}^{(t)})$ as function of t
    - Useful with different $h(\cdot)$ functions!
  - Cusum diagnostics
    - Plot $\sum_{i=1}^{t}[h(\boldsymbol{x}^{(i)}) - \hat{\theta}_n]$ versus t
    - Wiggly and small excursions from 0: Indicate chain is mixing well





The Gelman-Rubin diagnostic

- Motivated from analysis of variance
- Assume J chains run in parallel
- jth chain: $x_j^{(D+1)}, \ldots, x_j^{(D+L)}$ (first D discarded)

The Gelman-Rubin diagnostic

- Motivated from analysis of variance
- Assume J chains run in parallel
- jth chain: $x_j^{(D+1)}, \ldots, x_j^{(D+L)}$ (first D discarded)
- Define

$$\bar{x}_j = \frac{1}{L} \sum_{t=D+1}^{D+L} x_j^{(t)} \qquad \bar{x}_\cdot = \frac{1}{J} \sum_{j=1}^{J} \bar{x}_j$$

$$B = \frac{L}{J-1} \sum_{j=1}^{J} (\bar{x}_j - \bar{x}_\cdot)^2$$

$$W = \frac{1}{J} \sum_{j=1}^{J} s_j^2 \qquad s_j^2 = \frac{1}{L-1} \sum_{t=D+1}^{D+L} (x_j^{(t)} - \bar{x}_j)^2$$

The Gelman-Rubin diagnostic

- Motivated from analysis of variance
- Assume J chains run in parallel
- jth chain: $x_j^{(D+1)}, \ldots, x_j^{(D+L)}$ (first D discarded)
- Define

$$\bar{x}_j = \frac{1}{L} \sum_{t=D+1}^{D+L} x_j^{(t)} \qquad \bar{x}_\cdot = \frac{1}{J} \sum_{j=1}^{J} \bar{x}_j$$

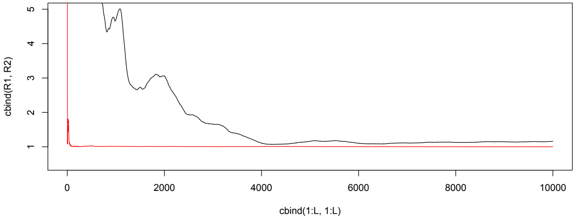
$$B = \frac{L}{J-1} \sum_{j=1}^{J} (\bar{x}_j - \bar{x}_\cdot)^2$$

$$W = \frac{1}{J} \sum_{j=1}^{J} s_j^2 \qquad s_j^2 = \frac{1}{L-1} \sum_{t=D+1}^{D+L} (x_j^{(t)} - \bar{x}_j)^2$$

- If converged, both B and W estimates $\sigma^2 = \mathrm{Var}_f[X]$
- Diagnostic: $R = \frac{\frac{L-1}{L} W + \frac{1}{L} B}{W}$
- "Rule": $\sqrt{R} < 1.1$ indicate D and L are sufficient

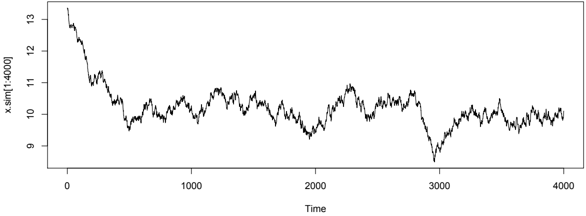
Example

- $D = 100, L = 1000$: $\sqrt{R_1} = 1.588$, $\sqrt{R_2} = 1.002$,
- $D = 1000, L = 1000$: $\sqrt{R_1} = 1.700$, $\sqrt{R_2} = 1.004$,
- $D = 1000, L = 10000$: $\sqrt{R_1} = 1.049$, $\sqrt{R_2} = 1.0008$



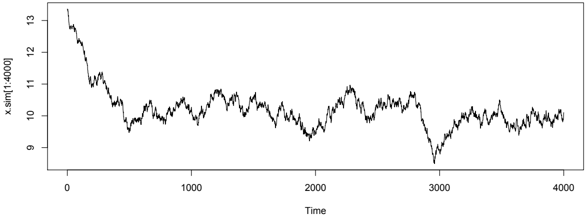
Apparent convergence

- $p(x) = 0.7 \cdot N(7, 0.5^2) + 0.3 \cdot N(10, 0.5^2)$
- Metropolis-Hastings with proposal $N(x^{(t)}, 0.05^2)$
- First 4000 samples (400 discarded)

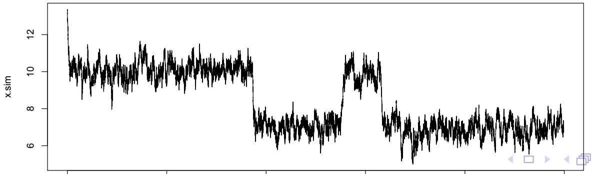


Apparent convergence

- $p(x) = 0.7 \cdot N(7, 0.5^2) + 0.3 \cdot N(10, 0.5^2)$
- Metropolis-Hastings with proposal $N(x^{(t)}, 0.05^2)$
- First 4000 samples (400 discarded)



- Full 10000 samples



M-H: Choice of proposal distribution

- Independence chain:
  - $g(\cdot) \approx p(\cdot)$
  - High acceptance rate
  - Tail properties most important: f/g should be bounded

M-H: Choice of proposal distribution

- Independence chain:
  - $g(\cdot) \approx p(\cdot)$
  - High acceptance rate
  - Tail properties most important: f/g should be bounded
- Random walk proposal
  - Tune variance so that acceptance rate is between 25 and 50%

Effective sample size for MCMC

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\text{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1} \frac{L-k}{L}\rho(k)\right] \xrightarrow{L\to\infty} \frac{\sigma_h^2}{L}[1 + 2\sum_{k=1}^{\infty} \rho(k)]$$

Effective sample size for MCMC

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\mathrm{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1}\frac{L-k}{L}\rho(k)\right] \overset{L\to\infty}{\to} \frac{\sigma_h^2}{L}[1 + 2\sum_{k=1}^{\infty}\rho(k)]$$

- If independent samples:

$$\mathrm{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}$$

Effective sample size for MCMC

- For $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

$$\mathrm{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}\left[1 + 2\sum_{k=1}^{L-1}\frac{L-k}{L}\rho(k)\right] \xrightarrow{L\to\infty} \frac{\sigma_h^2}{L}[1 + 2\sum_{k=1}^{\infty}\rho(k)]$$

- If independent samples:

$$\mathrm{Var}[\hat{\theta}] = \frac{\sigma_h^2}{L}$$

- Effective sample size: $\frac{L}{1+2\sum_{k=1}^{\infty}\rho(k)}$
- Use empirical estimates $\hat{\rho}(k)$
- Usual to truncate the summation when $\hat{\rho}(k) < 0.1$.

Number of chains

- Assume possible to perform N iterations
  - One long chain of length N, or
  - J parallel chains, each of length N/J?

Number of chains

- Assume possible to perform N iterations
  - One long chain of length N, or
  - J parallel chains, each of length N/J?
- Burnin:
  - One long chain: Only need to discard D samples
  - Parallel chains: Need to discard $J \cdot D$ samples

Number of chains

- Assume possible to perform N iterations
  - One long chain of length N, or
  - J parallel chains, each of length N/J?
- Burnin:
  - One long chain: Only need to discard D samples
  - Parallel chains: Need to discard $J \cdot D$ samples
- Check of convergence
  - Easier with many parallel chains

Number of chains

- Assume possible to perform N iterations
  - One long chain of length N, or
  - J parallel chains, each of length N/J?
- Burnin:
  - One long chain: Only need to discard D samples
  - Parallel chains: Need to discard $J \cdot D$ samples
- Check of convergence
  - Easier with many parallel chains
- Efficiency
  - Parallel chains give more independent samples

Number of chains

- Assume possible to perform N iterations
  - One long chain of length N, or
  - J parallel chains, each of length N/J?
- Burnin:
  - One long chain: Only need to discard D samples
  - Parallel chains: Need to discard $J \cdot D$ samples
- Check of convergence
  - Easier with many parallel chains
- Efficiency
  - Parallel chains give more independent samples
- Computational issues
  - Possible to utilize multiple cores with parallel chains

Example

- Number of positive tests of Covid-19 Nov 1 - Nov 7 2021 (county level):

Day	1	2	3	4	5	6	7	8	9	10
Cases	845	331	76	47	1105	126	186	58	156	258
Population	693494	479892	265238	241235	1241165	371385	419396	307231	636531	468702

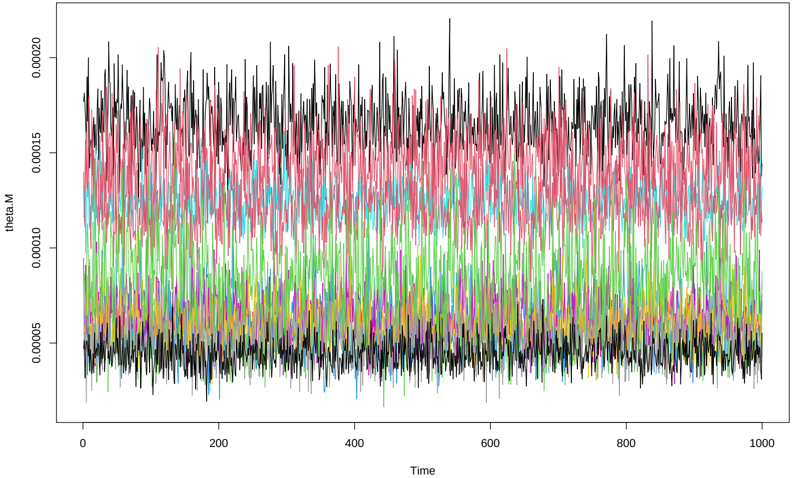
- Model:

$$\begin{aligned} y_j \sim& \text{Poisson}(N_j\theta_j) \\ \theta_j \sim& \text{Gamma}(a, b) \\ p(b|a) \sim& \text{Gamma}(\alpha, \beta) \qquad a \text{ assumed fixed} \end{aligned}$$

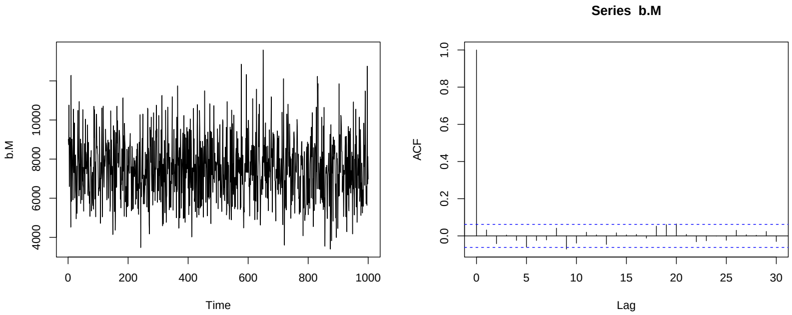
- Can show

$$\begin{aligned} p(\boldsymbol{\theta}_{1:11}|a, b, \mathbf{y}) =& \prod_{j=1}^{11} \text{Gamma}(a + y_j, b + N_j) \\ p(b|a, \boldsymbol{\theta}, \mathbf{y}) =& \text{Gamma}(\alpha + 11a, \beta + \sum_{j=1}^{1} 1\theta_j) \end{aligned}$$

Res-covid


0.00005
0.00010
0.00015
0.00020
theta.M
0
200
400
600
800
1000
Time

Res-covid


b.M
4000
6000
8000
10000
0
200
400
600
800
1000
Time
Series b.M
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
Lag

Data uncertainty and Monte Carlo uncertainty

- Parameter: $\theta = E^p[h(\boldsymbol{x})]$
- Estimator: $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:

Data uncertainty and Monte Carlo uncertainty

- Parameter: $\theta = E^p[h(\boldsymbol{x})]$
- Estimator: $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:
- Two types of uncertainty
  - Variability in $h(\boldsymbol{x})$: $\sigma_h^2 = \mathrm{Var}^p[h(\boldsymbol{x})]$
    - Estimator: $\hat{\sigma}_h^2 = \frac{1}{L}\sum_{t=D+1}^{D+L}[h(\boldsymbol{x}^{(t)}) - \hat{\theta}]^2$

Data uncertainty and Monte Carlo uncertainty

- Parameter: $\theta = E^p[h(\boldsymbol{x})]$
- Estimator: $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:
- Two types of uncertainty
  - Variability in $h(\boldsymbol{x})$: $\sigma_h^2 = \text{Var}^p[h(\boldsymbol{x})]$
    - Estimator: $\hat{\sigma}_h^2 = \frac{1}{L}\sum_{t=D+1}^{D+L}[h(\boldsymbol{x}^{(t)}) - \hat{\theta}]^2$
  - MC variability in $\hat{\theta}$:
    - Estimator: Divide data into batches of size $b = \lfloor L^{1/a} \rfloor$, make estimates $\hat{\theta}$ within each batch and variance from these

Data uncertainty and Monte Carlo uncertainty

- Parameter: $\theta = E^p[h(\boldsymbol{x})]$
- Estimator: $\hat{\theta} = \frac{1}{L}\sum_{t=D+1}^{D+L} h(\boldsymbol{x}^{(t)})$:
- Two types of uncertainty
  - Variability in $h(\boldsymbol{x})$: $\sigma_h^2 = \text{Var}^p[h(\boldsymbol{x})]$
    - Estimator: $\hat{\sigma}_h^2 = \frac{1}{L}\sum_{t=D+1}^{D+L}[h(\boldsymbol{x}^{(t)}) - \hat{\theta}]^2$
  - MC variability in $\hat{\theta}$:
    - Estimator: Divide data into batches of size $b = \lfloor L^{1/a} \rfloor$, make estimates $\hat{\theta}$ within each batch and variance from these
- Recommendation: Specify L so that MC variability is less than 5% of variability in $h(\boldsymbol{x})$.

Uncertainty in Bayesian setting

- Interest in $\boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\boldsymbol{y})$
- Monte Carlo estimate of mean: $\hat{\boldsymbol{\mu}}_\theta = \frac{1}{L}\sum_{i=1}^{L} \boldsymbol{\theta}^i$
- Can show:

$$
\begin{aligned}
E[\hat{\mu}_\theta] =& \boldsymbol{\mu}_\theta = E[\boldsymbol{\theta}|\boldsymbol{y}] \\
E[(\boldsymbol{\theta} - \hat{\boldsymbol{\mu}}_\theta)^2] =& \left(1 + \tfrac{1}{L}\right) E[(\boldsymbol{\theta} - \boldsymbol{\mu}_\theta)^2]
\end{aligned}
$$

Hamiltonian MC

Hamiltonian MC

- Common trick in Monte Carlo: Introduce auxiliary variables
- Hamiltonian MC (Neal et al., 2011):

$$
\begin{aligned}
\pi(\boldsymbol{q}) &\propto \exp(-U(\boldsymbol{q})) && \text{Distribution of interest} \\
\pi(\boldsymbol{q}, \boldsymbol{p}) &\propto \exp(-U(\boldsymbol{q}) - 0.5\boldsymbol{p}^T\boldsymbol{p}) && \text{Extended distribution} \\
&= \exp(-H(\boldsymbol{q}, \boldsymbol{p})) && H(\boldsymbol{q}, \boldsymbol{p}) = U(\boldsymbol{q}) + 0.5\boldsymbol{p}^T\boldsymbol{p}
\end{aligned}
$$

Hamiltonian MC

- Common trick in Monte Carlo: Introduce auxiliary variables
- Hamiltonian MC (Neal et al., 2011):

$$\begin{aligned} \pi(\boldsymbol{q}) &\propto \exp(-U(\boldsymbol{q})) && \text{Distribution of interest} \\ \pi(\boldsymbol{q}, \boldsymbol{p}) &\propto \exp(-U(\boldsymbol{q}) - 0.5\boldsymbol{p}^T\boldsymbol{p}) && \text{Extended distribution} \\ &= \exp(-H(\boldsymbol{q}, \boldsymbol{p})) && H(\boldsymbol{q}, \boldsymbol{p}) = U(\boldsymbol{q}) + 0.5\boldsymbol{p}^T\boldsymbol{p} \end{aligned}$$

- Note
  - $\boldsymbol{q}$ and $\boldsymbol{p}$ are independent
  - $\boldsymbol{p} \sim N(\boldsymbol{0}, \boldsymbol{I})$.
  - Usually dim($\boldsymbol{p}$)= dim($\boldsymbol{q}$)

Hamiltonian MC

- Common trick in Monte Carlo: Introduce auxiliary variables
- Hamiltonian MC (Neal et al., 2011):

$$\begin{aligned} \pi(\boldsymbol{q}) &\propto \exp(-U(\boldsymbol{q})) && \text{Distribution of interest} \\ \pi(\boldsymbol{q},\boldsymbol{p}) &\propto \exp(-U(\boldsymbol{q}) - 0.5\boldsymbol{p}^T\boldsymbol{p}) && \text{Extended distribution} \\ &= \exp(-H(\boldsymbol{q},\boldsymbol{p})) && H(\boldsymbol{q},\boldsymbol{p}) = U(\boldsymbol{q}) + 0.5\boldsymbol{p}^T\boldsymbol{p} \end{aligned}$$

- Note
  - $\boldsymbol{q}$ and $\boldsymbol{p}$ are independent
  - $\boldsymbol{p} \sim N(\mathbf{0}, \boldsymbol{I})$.
  - Usually dim($\boldsymbol{p}$)= dim($\boldsymbol{q}$)
- Algorithm ($\boldsymbol{q}$) current value
  1. Simulate $\boldsymbol{p} \sim N(\mathbf{0}, \boldsymbol{I})$
  2. Generate $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ such that $H(\boldsymbol{q}^*, \boldsymbol{p}^*) \approx H(\boldsymbol{q}, \boldsymbol{p})$
  3. Accept $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ by a Metropolis-Hastings step

Hamiltonian MC

- Common trick in Monte Carlo: Introduce auxiliary variables
- Hamiltonian MC (Neal et al., 2011):

$$
\begin{aligned}
\pi(\boldsymbol{q}) &\propto \exp(-U(\boldsymbol{q})) && \text{Distribution of interest}\\
\pi(\boldsymbol{q},\boldsymbol{p}) &\propto \exp(-U(\boldsymbol{q}) - 0.5\boldsymbol{p}^T\boldsymbol{p}) && \text{Extended distribution}\\
&= \exp(-H(\boldsymbol{q},\boldsymbol{p})) && H(\boldsymbol{q},\boldsymbol{p}) = U(\boldsymbol{q}) + 0.5\boldsymbol{p}^T\boldsymbol{p}
\end{aligned}
$$

- Note
  - $\boldsymbol{q}$ and $\boldsymbol{p}$ are independent
  - $\boldsymbol{p} \sim N(\boldsymbol{0}, \boldsymbol{I})$.
  - Usually dim($\boldsymbol{p}$)= dim($\boldsymbol{q}$)
- Algorithm ($\boldsymbol{q}$) current value
  1. Simulate $\boldsymbol{p} \sim N(\boldsymbol{0}, \boldsymbol{I})$
  2. Generate $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ such that $H(\boldsymbol{q}^*, \boldsymbol{p}^*) \approx H(\boldsymbol{q}, \boldsymbol{p})$
  3. Accept $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ by a Metropolis-Hastings step
- Step 1 is a Gibbs sampling step!
- Main challenge: Generate $(\boldsymbol{q}^*, \boldsymbol{p}^*)$

Hamiltonian dynamics

- Consider $(\boldsymbol{q}, \boldsymbol{p})$ as a time-process $(\boldsymbol{q}(t), \boldsymbol{p}(t))$
- Hamiltonian dynamics: Change through

$$
\begin{aligned}
\frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \\
\frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i}
\end{aligned}
$$

Hamiltonian dynamics

- Consider $(\boldsymbol{q}, \boldsymbol{p})$ as a time-process $(\boldsymbol{q}(t), \boldsymbol{p}(t))$
- Hamiltonian dynamics: Change through

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

This gives

$$\frac{dH}{dt} = \sum_{i=1}^{d}\left[\frac{\partial H}{\partial q_i}\frac{dq_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt}\right]$$
$$= \sum_{i=1}^{d}\left[\frac{\partial H}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial q_i}\right] = 0$$

Hamiltonian dynamics

- Consider $(\boldsymbol{q}, \boldsymbol{p})$ as a time-process $(\boldsymbol{q}(t), \boldsymbol{p}(t))$
- Hamiltonian dynamics: Change through

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

This gives

$$\frac{dH}{dt} = \sum_{i=1}^{d} \left[\frac{\partial H}{\partial q_i}\frac{dq_i}{dt} + \frac{\partial H}{\partial p_i}\frac{dp_i}{dt} \right] = \sum_{i=1}^{d} \left[\frac{\partial H}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial H}{\partial p_i}\frac{\partial H}{\partial q_i} \right] = 0$$

- If we can change $(\boldsymbol{q}, \boldsymbol{p})$ exactly by the Hamiltonian dynamics, H will not change!
- In practice, only possible to make numerical approximations

Hamiltonian dynamics - Eulers method

- Assume

$$
\begin{aligned}
p_i(t+\varepsilon) =& p_i(t) + \varepsilon \frac{dp_i}{dt}(t) \\
=& p_i(t) - \varepsilon \frac{\partial U}{\partial q_i}(q_i(t)) \\
q_i(t+\varepsilon) =& q_i(t) + \varepsilon \frac{dq_i}{dt}(t) \\
=& q_i(t) + \varepsilon p_i(t)
\end{aligned}
$$

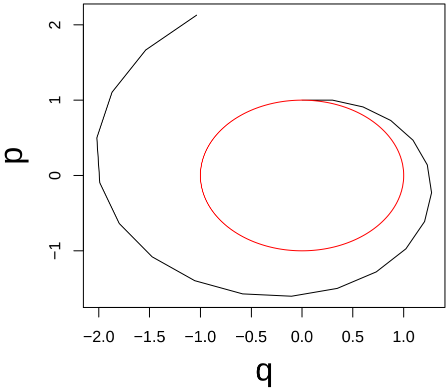
- Note: Derivatives of $U(\boldsymbol{q})$ are used.
- However, not very exact.

Hamiltonian dynamics - Eulers method

- Assume

$$
\begin{aligned}
p_i(t+\varepsilon) =& p_i(t) + \varepsilon \frac{dp_i}{dt}(t) \\
=& p_i(t) - \varepsilon \frac{\partial U}{\partial q_i}(q_i(t)) \\
q_i(t+\varepsilon) =& q_i(t) + \varepsilon \frac{dq_i}{dt}(t) \\
=& q_i(t) + \varepsilon p_i(t)
\end{aligned}
$$

- Note: Derivatives of $U(\boldsymbol{q})$ are used.
- However, not very exact.

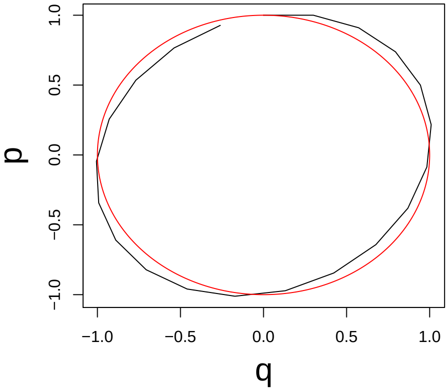


Hamiltonian dynamics - the modified Eulers method

- Assume

$$
\begin{aligned}
p_i(t+\varepsilon) =& p_i(t) - \varepsilon \frac{\partial U}{\partial q_i}(q(t)) \\
q_i(t+\varepsilon) =& q_i(t) + \varepsilon p_i(t+\varepsilon)
\end{aligned}
$$

- Better than Eulers method.

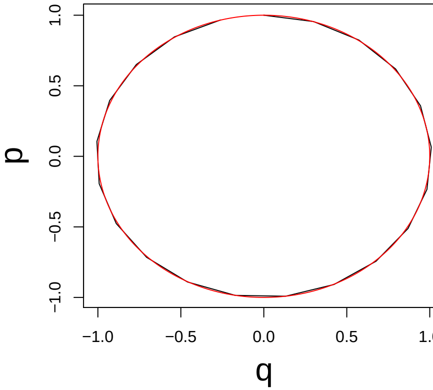


Hamiltonian dynamics - the Leapfrog method

- Assume

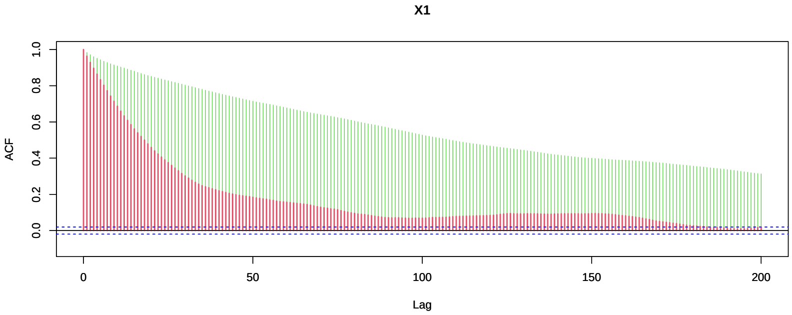
$$
\begin{aligned}
p_i(t+\tfrac{\varepsilon}{2}) =& p_i(t) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q(t))\\
q_i(t+\varepsilon) =& q_i(t) + \varepsilon p_i(t+\tfrac{\varepsilon}{2})\\
p_i(t+\varepsilon) =& p_i(t+\tfrac{\varepsilon}{2}) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q_(t+\varepsilon))
\end{aligned}
$$

- Quite exact!
- Idea: Use this L steps



Example - 2-dimensional Gaussian

- Assume $\boldsymbol{x} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.95 \\ 0.95 & 1 \end{pmatrix}$
- $H(\boldsymbol{x}, \boldsymbol{p}) = 0.5\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + 0.5\boldsymbol{p}^T\boldsymbol{p}$
- Use $L = 5$ leapfrog steps, with stepsize $\varepsilon = 0.1$
- `Leapfrog_Gauss2.R`



Example - mixture Gaussians

- Assume

$$\pi(x) = pN(x; \mu_1, \sigma_1^2) + (1-p)N(x; \mu_2, \sigma_2^2)$$

- $H(\boldsymbol{x}, \boldsymbol{p}) = -log(\pi(x) + 0.5\boldsymbol{p}^T\boldsymbol{p}$
- Use $L = 5$ leapfrog steps, with stepsize $\varepsilon = 0.1$
- `Leapfrog_mixture.R`

Pseudo-marginal MCMC

Pseudo-marginal MCMC - Andrieu and Roberts (2009)

- Assume a Hierarchical model

$$\begin{aligned} \boldsymbol{y} &\sim p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\theta}) && \text{Observations} \\ \boldsymbol{x} &\sim p(\boldsymbol{x}|\boldsymbol{\theta}) && \text{Latent process} \\ \boldsymbol{\theta} &\sim p(\boldsymbol{\theta}) && \text{Prior} \end{aligned}$$

- Main interest in $\boldsymbol{\theta}$,

$$\pi(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{y}) = \int_{\boldsymbol{x}} p(\boldsymbol{\theta},\boldsymbol{x}|\boldsymbol{y})d\boldsymbol{x}$$

- More general setting: Interest in only a small subsets of the unknowns
- Possible: Construct MCMC algorithm for $p(\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y})$
- More efficient: MCMC algorithm directly for $p(\boldsymbol{\theta}|\boldsymbol{y})$
  - M-H: Require calculation of $\frac{\pi(\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta})}$, difficult!
- Idea of pseudo-marginal MCMC: Replace $\pi(\boldsymbol{\theta}^*)$ by an (unbiased) estimate!

Pseudo-marginal MCMC - cont

- Algorithm, current values $\theta_t, \hat{\pi}(\theta_t)$:
  1. Sample $\theta^* \sim g(\theta^*|\theta)$
  2. Construct estimate $\hat{\pi}(\theta^*)$
  3. Accept θ^* with probability

$$\widehat{R}(\theta^*) = \min\left\{1, \frac{\hat{\pi}(\theta^*)g(\theta|\theta^*)}{\hat{\pi}(\theta)g(\theta^*|\theta)}\right\}$$

- Property: If
  - the M-H algorithm with exact $\pi(\theta)$ converges "properly"
  - the esimates $\hat{\pi}(\theta)$ are unbiased

  then the approximate M-H algorithm will also converge "properly"
  - but convergence is slower!
- See Andrieu and Roberts (2009) for more details

How to estimate $\pi(\theta)$?

- We have

$$\begin{aligned}\pi(\boldsymbol{\theta}) &= \int_{\boldsymbol{x}} \pi(\boldsymbol{x},\boldsymbol{\theta})d\boldsymbol{x}\\ &= \int_{\boldsymbol{x}} \frac{\pi(\boldsymbol{x},\boldsymbol{\theta})}{g_\theta(\boldsymbol{x})} g_\theta(\boldsymbol{x})d\boldsymbol{x}\\ &\approx \frac{1}{N}\frac{\pi(\boldsymbol{x}^{(i)},\boldsymbol{\theta})}{g_\theta(\boldsymbol{x}^{(i)})} \qquad\qquad \boldsymbol{x}^{(i)} \sim g_\theta(\boldsymbol{x})\end{aligned}$$

- Unknown normalization constant cancels out in ratio.
- Alternatives possible

Why do pseudo-marginal MCMC work?

- Consider extended distribution

$$\tilde{\pi}(\theta, \mathbf{x}_{1:N}) = \frac{1}{N}\sum_{k=1}^{N}\pi(\theta, \mathbf{x}_k)\prod_{i\neq k} g_\theta(\mathbf{x}_i)$$

- We have

$$\int_{\mathbf{x}_{1:N}} \tilde{\pi}(\theta, \mathbf{x}_{1:N}) d\mathbf{x}_{1:N} = \pi(\theta)$$

- M-H algorithm (given $(\theta, \mathbf{x}_{1:N})$):
  1. Generate $\theta^* \sim g(\theta^*|\theta)$
  2. Generate $\mathbf{x}_{1:N} \sim \prod_{k=1}^{N} g_{\theta^*}(\mathbf{x}_k^*)$
  3. Calculate acceptance ratio

$$r = \frac{\tilde{\pi}(\theta^*, \mathbf{x}_{1:N}^*)}{\tilde{\pi}(\theta, \mathbf{x}_{1:N})} \times \frac{g(\theta^*|\theta)\prod_{k=1}^{N} g_{\theta^*}(\mathbf{x}_k^*)}{g(\theta^*|\theta)\prod_{k=1}^{N} g_{\theta^*}(\mathbf{x}_k^*)}$$

Logist regression

- Assume

$$y_i \sim \text{Binom}\left(1, \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right)$$

$$\eta_t = = \beta_0 + \sum_{j=1}^{p} {\color{red}\gamma}\beta_j x_{ij}$$

- Different models depending on $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)$
- Marginal distributions:

$$\begin{aligned} p(\boldsymbol{y}\boldsymbol{\gamma}) &= \int_{\boldsymbol{\beta}_\gamma} p(\boldsymbol{y}|\boldsymbol{\beta}_\gamma)p(\boldsymbol{\beta}_\gamma|\boldsymbol{\gamma})d\boldsymbol{\beta}_\gamma \\ &= \int_{\boldsymbol{\beta}_\gamma} p(\boldsymbol{y}|\boldsymbol{\beta}_\gamma)\frac{p(\boldsymbol{\beta}_\gamma|\boldsymbol{\gamma})}{g(\boldsymbol{\beta}_\gamma|\boldsymbol{\gamma})}g(\boldsymbol{\beta}_\gamma|\boldsymbol{\gamma})d\boldsymbol{\beta}_\gamma \end{aligned}$$

- Main challenge: Choice of $g()$
- Script: `Logreg_Pseudo.R`

Reversible jump MCMC

Changing dimensions

- Assume several models $\mathcal{M}_1, \ldots, \mathcal{M}_K$
- Corresponding parameters $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ of different dimensions!
- Aim: Simulate $\boldsymbol{x} = (\mathcal{M}, \boldsymbol{\theta}_{\mathcal{M}})$
- RJMCMC: M-H method for moving between spaces of different dimensions
- Main challenges:
  - When changing dimensions, how to compare densities on different spaces?
  - When changing $\mathcal{M} \to \mathcal{M}^*$, how to propose $\boldsymbol{\theta}_{\mathcal{M}^*}$?

Reversible jump MCMC

- Green (1995): Include auxiliary variables to match dimensions.
- Consider change $(\mathcal{M}_1, \boldsymbol{\theta}_1)$ to $(\mathcal{M}_2, \boldsymbol{\theta}_2)$ with $|\boldsymbol{\theta}_1| < |\boldsymbol{\theta}_2|$
  - $j(1 \to 2)$ probability for moving from $\mathcal{M}_1$ to $\mathcal{M}_2$
- Algorithm
  1. Generate $\boldsymbol{u}_1$ such that $|\boldsymbol{\theta}_1| + |\boldsymbol{u}_1| = |\boldsymbol{\theta}_2|$
  2. Propose $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_2(\boldsymbol{\theta}_1, \boldsymbol{u}_1)$
  3. Calculate acceptance ratio

$$r = \frac{\pi(\mathcal{M}_2, \boldsymbol{\theta}_2) q(2 \to 1)}{\pi(\mathcal{M}_1, \boldsymbol{\theta}_1) q(1 \to 2) q(\boldsymbol{u}_1)} \left| \frac{\partial(\boldsymbol{\theta}_2)}{\partial(\boldsymbol{\theta}_1, \boldsymbol{u}_1)} \right|$$

  4. Accept with probability $\min\{1, r\}$.
- Use $1/r$ for opposite move
- More general settings possible

Logist regression

- Assume

$$y_i \sim \text{Binom}\left(1, \frac{\exp(\eta_i)}{1+\exp(\eta_i)}\right)$$

$$\eta_t == \beta_0 + \sum_{j=1}^{p} \gamma_j \beta_j x_{ij}$$

- Different models depending on $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_p)$
- RJMCMC with changing one γ_j at a time: $\gamma_j \rightarrow 1 - \gamma_j$

RJ - logistic regression

- Adding component: $\gamma_j = 0 \rightarrow \gamma_j = 1$
- Increase dimension by 1.
- u_1: Simulation of β_j
- Jacobian=1
- Script: `Logreg_RJ.R`

Additional topics

Additional topics in MCMC

- Mode jumping MCMC
- Reversible jump MCMC
- Non-reversible MCMC
- Subsampling MCMC
- Continuous-time Markov processes
- Adaptive MCMC: Automatic tuning of proposal distributions
  - Main challenge: Specifying proposal based on history of chain breaks down the Markov property
  - Solution: Reduce the amount of tuning as the number of iterations increases

Additional topics in MCMC

- Mode jumping MCMC
- Reversible jump MCMC
- Non-reversible MCMC
- Subsampling MCMC
- Continuous-time Markov processes
- Adaptive MCMC: Automatic tuning of proposal distributions
  - Main challenge: Specifying proposal based on history of chain breaks down the Markov property
  - Solution: Reduce the amount of tuning as the number of iterations increases
- Simulated tempering
  - Define $f^i(\boldsymbol{x}) \propto \pi(\boldsymbol{x})^{1/\tau_i}$, $1 = \tau_1 < \tau_2 < \cdots < \tau_m$
  - Simulate $(\boldsymbol{x}, I)$, where I changes distribution
  - Easier to move around when $\tau_i > 1$
  - Keep samples for which $I = 1$

Additional topics in MCMC

- Mode jumping MCMC
- Reversible jump MCMC
- Non-reversible MCMC
- Subsampling MCMC
- Continuous-time Markov processes
- Adaptive MCMC: Automatic tuning of proposal distributions
  - Main challenge: Specifying proposal based on history of chain breaks down the Markov property
  - Solution: Reduce the amount of tuning as the number of iterations increases
- Simulated tempering
  - Define $f^i(\boldsymbol{x}) \propto \pi(\boldsymbol{x})^{1/\tau_i}$, $1 = \tau_1 < \tau_2 < \cdots < \tau_m$
  - Simulate $(\boldsymbol{x}, I)$, where I changes distribution
  - Easier to move around when $\tau_i > 1$
  - Keep samples for which $I = 1$
- Multiple-Try M-H
  - Generate k proposals $\boldsymbol{x}_1^*, ..., \boldsymbol{x}_k^*$ from $g(\cdot|\boldsymbol{x}^{(t)})$
  - Select $\boldsymbol{x}_j^*$ with probability $w(\boldsymbol{x}^{(t)}, \boldsymbol{x}_j^*) = \pi(\boldsymbol{x}^{(t)})g(\boldsymbol{x}_j^*|\boldsymbol{x}^{(t)})\lambda(\boldsymbol{x}^{(t)}, \boldsymbol{x}_j^*)$, λ symmetric
  - Sample $\boldsymbol{x}_1^{**}, ..., \boldsymbol{x}_{k-1}^{**}$ from $g(\cdot|\boldsymbol{x}_j^*)$, put $\boldsymbol{x}_k^{**} = \boldsymbol{x}^{(t)}$
  - Use Generalized M-H ratio

$$R_g = \frac{\sum_{i=1}^k w(\boldsymbol{x}^{(t)}, \boldsymbol{x}_i^*)}{\sum_{i=1}^k w(\boldsymbol{x}_j^*, \boldsymbol{x}_i^{**})}$$

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

R. M. Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.