L8. Computational Bayesian Statistics (Principles)

- Bayes' Theorem is usually expressed very simply in the unscaled form:
  - ➢ *posterior* proportional to *prior* times *likelihood*:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{K} \approx p(y|\theta)p(\theta) = g(\theta|y)$$

(unscaled Posterior distribution)

▸ $K = \iiint p(y|\theta_1, \theta_2, \ldots, \theta_p)p(\theta_1, \theta_2, \ldots, \theta_p)d\theta_1 \ldots d\theta_p$

▸ The closed form for high-dimensional integral exists only for particular cases,
(e.g., likelihood is in exponential family and prior is conjugate prior to likelihood)

- The difficulty in computing K has restricted the easy implementation of Bayesian statistics

- Without knowing the value for K, we only know the shape of $p(\theta|D)$:

  - Can: find a mode, compute relative values at any two location
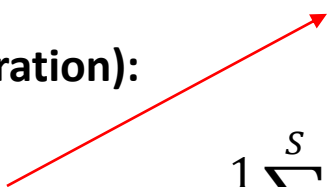 - Can't: compute a probability (density), compute moments, conduct inferences

To conduct meaningful inferences, we need to compute or approximate the posterior $p(\theta|D)$

1. Analytical computation

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)\, d\theta}$$

2. Numerical Approximation (integration):

$$\int p(y|\theta)p(\theta)\, d\theta \approx \frac{1}{S}\sum_{s=1}^{S} w_s\, p(y|\theta^s)p(\theta^s)$$

- with the weight w_s corresponding to the volume of space represented by the point θ^s

3. Direct Sampling (from unscaled posterior $g(\theta|y) = p(y|\theta)p(\theta)$)
- Rejection sampling
- Importance sampling

4. Markov Chain Monte Carlo (MCMC) sampling (from unscaled posterior $g(\theta|y) = p(y|\theta)p(\theta)$)
- Metropolis Sampling
- Metropolis Hasting sampling
- Gibbs sampling

5. Distributional Approximation
- Laplace approximation
- Variational inference

1. Analytical computation

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)\, d\theta}$$

2. Numerical Approximation (integration):

$$\int p(y|\theta)p(\theta)\, d\theta \approx \frac{1}{S}\sum_{s=1}^{S} w_s\, p(y|\theta^s)p(\theta^s)$$

- with the weight w_s corresponding to the volume of space represented by the point θ^s

3. Direct Sampling (Monte Carlo Sampling) (from unscaled posterior $g(\theta|y) = p(y|\theta)p(\theta)$)
- Rejection sampling
- Importance sampling

4. Markov Chain Monte Carlo (MCMC) sampling (from unscaled posterior $g(\theta|y) = p(y|\theta)p(\theta)$)
- Metropolis Sampling
- Metropolis Hasting sampling
- Gibbs sampling

5. Distributional Approximation
- Laplace approximation
- Variational inference

This lecture will focus on sampling methods

Replace very difficult numerical integration with the much easier process of drawing random samples

Direct Sampling Approaches

What can we do, when we can do sampling?

- Conventional inverse sampling require exact form of $p(\theta|y)$ but we can also samples from unscaled posteriors $g(\theta|y)$

- If we can draw samples $\theta_i \sim p(\theta|y)$ (using $g(\theta|y)$) for $i = 1, \dots, S$, we can

  ➢ Construct histogram (approximated PDF or PMF)

  ➢ Compute the probability for some event:

  $$p(\theta \in A|y) \approx \frac{1}{S}\sum_{i=1}^{S} I(\theta_i \in A), \text{ with } \theta_i \sim p(\theta|y)$$

  ➢ Compute Moments:

  $$E(\theta) = \int \theta p(\theta|y)d\theta \approx \frac{1}{S}\sum_{i=1}^{S} \theta_i, \text{ with } \theta_i \sim p(\theta|y)$$

  ➢ Predict future values (posterior predictive distribution):

  $$p(y_{new}|y) = \int_\theta p(y_{new}|\theta, y)p(\theta|y)d\theta \approx \frac{1}{S}\sum_{i=1}^{S} p(y_{new}|\theta_i, y), \text{ with } \theta_i \sim p(\theta|y)$$

- Allow applied statistician to use more realistic models because he or she is not constrained by analytic or numerical tractability

  ➢ Models that are based on the underlying situation can be used instead of models based on mathematical convenience

- They are not approximations: Estimates found from the Monte Carlo random sample from the posterior can achieve any required accuracy by setting the sample size large enough

- For high dimensional parameter space, sampling method is the only feasible method

- Existing exploratory data analysis (EDA) techniques can be used to explore the posterior

- They allow sensitivity analysis to be made on the model in a simple fashion

Direct sampling	Indirect sampling

Markov chain→ steady-state distribution

Sampling from unscaled version of $p(\theta|y)$

$$\mathbf{P}' = \begin{bmatrix} .400 & .150 & .300 & .150 \\ .100 & .467 & .300 & .133 \\ .150 & .225 & .575 & .050 \\ .300 & .400 & .200 & .100 \end{bmatrix} \Rightarrow \boldsymbol{\pi} = (.2, \ .3, \ .4, \ .1)$$

$$= p(\theta|y)$$

- **The acceptance-Rejection sampling (ARS)** algorithm allows us to draw a random sample directly from a target distribution when we only know its shape

- **ARS** works by reshaping a random sample drawn from an easily sampled candidate distribution (a.k.a., starting distribution) into a random sample from the target by only accepting some of the candidate values into the final sample

- Let the unscaled posterior $g(\theta) = p(\theta)f(y|\theta)$ be the unscaled target

- Let the easily sampled candidate density be $g_0(\theta)$

- It requires the candidate density dominates the unscaled target, which means that we can find a number M such that

$$Mg_0(\theta) \geq p(\theta)f(y|\theta) \text{ for all } \theta$$

Algorithm (for drawing a single sample)

1) Draw a random value $\theta^* \sim g_0(\theta)$

2) Calculate

$$w(\theta^*) = \frac{p(\theta^*)f(y|\theta^*)}{Mg_0(\theta^*)}, 0 \leq w(\theta^*) \leq 1$$

3) Draw (independently) u from uniform[0,1]

4) if $u > w(\theta^*)$: reject θ^*
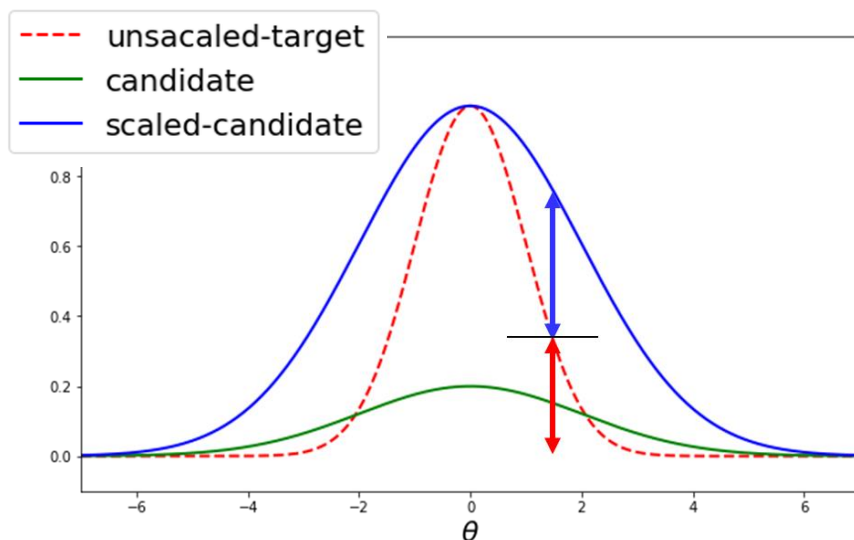
   if $u < w(\theta^*)$: accept θ^*

Example: Sample from the unscaled target $g(\theta) = p(\theta)f(y|\theta) = e^{-\frac{\theta^2}{2}}$

- Choose the normal candidate density, $g_0(\theta) = N(0,2^2)$
- Find the smallest value of M such that for all θ,
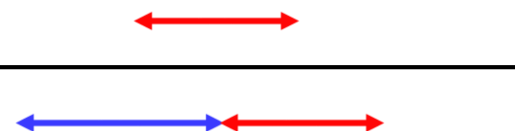
$$M \times g_0(\theta) \geq p(\theta) \times f(y|\theta)$$

$$M \geq \frac{p(\theta) \times f(y|\theta)}{g_0(\theta)} = \frac{e^{-\frac{\theta^2}{2}}}{\frac{1}{\sqrt{2\pi}2}e^{-\frac{\theta^2}{2}}}$$

➢ The maximum of RHS is $5.01325 \rightarrow M = 5.01325$

$$w(\theta^*) = \frac{p(\theta^*)f(y|\theta^*)}{M g_0(\theta^*)}$$
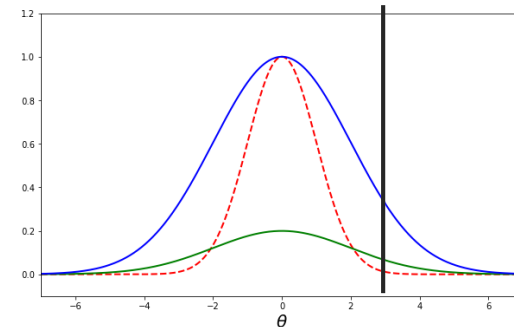
Acceptance rate =

Direction Sampling: Acceptance-Rejection Sampling

- Suppose that the first value drawn from the candidate density $g_0(\theta)$ is $\theta = 2.60$

- The value of the unscaled target will be

$$p(\theta = 2.60) \times f(y|\theta = 2.60) = \exp\left(-\frac{2.60^2}{2}\right) = 0.0340475$$

- The value of the candidate density will be

$$g_0(\theta = 2.60) = \frac{1}{\sqrt{2\pi}2} e^{-\frac{2.60^2}{2\times2^2}} = 0.0856843$$

- Compute the weight function at the value $\theta = 2.60$

$$w(\theta = 2.60) = \frac{\color{red}{0.0340475}}{\color{blue}{5.01326 \times 0.0856843}} = 0.0792617$$

- Suppose we draw $u = 0.435198$ from $u\sim U[0,1]$

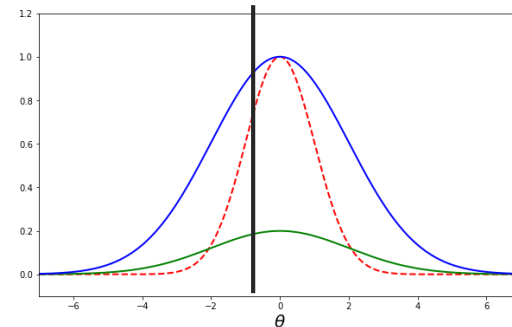- Since $u = 0.435198 > w(\theta = 2.60) = 0.0792617$ **Reject** $\theta = 2.60$

Direction Sampling: Acceptance-Rejection Sampling

- Suppose that the first value drawn from the candidate density $g_0(\theta)$ is $\theta = -0.94$

- The value of the unscaled target will be

$$p(\theta = -0.94) \times f(y|\theta = -0.94) = \exp\left(-\frac{-0.94^2}{2}\right) = 0.642878$$

- The value of the candidate density will be

$$g_0(\theta = -0.94) = \frac{1}{\sqrt{2\pi}2} e^{-\frac{0.94^2}{2\times 2^2}} = 0.178613$$

- Compute the weight function at the value $\theta = 2.60$

$$w(\theta = -0.94) = \frac{\color{red}{0.642878}}{\color{blue}{5.01326 \times 0.178613}} = 0.717953$$

- Suppose we draw $u = 0.577230$ from $u \sim U[0,1]$

- Since $u = 0.577230 < w(\theta = -0.94) = 0.717953$ **Accept** $\theta = -0.94$

Direction Sampling: Acceptance-Rejection Sampling

```python
p = lambda x: 0.7*norm.pdf(x,-0.3, 1)+0.3*norm.pdf(x,2.5, 1)
g = lambda x: norm.pdf(x,0, 3) # our proposal pdf : N(0,2^2)

N = 10000 # the total of samples we wish to generate
accepted = 0 # the number of accepted samples
samples = np.zeros(N)
count = 0 # the total count of proposals

M = 3
# generation loop
while (accepted < N):
    # Sample from g using inverse sampling
    xproposal = norm.rvs(0, 3, size=1)

    # pick a uniform number on [0, 1)
    y = np.random.uniform(0,1)

    # Do the accept/reject comparison
    if y < p(xproposal)/(M*g(xproposal)):
        samples[accepted] = xproposal
        accepted += 1
    count +=1
```
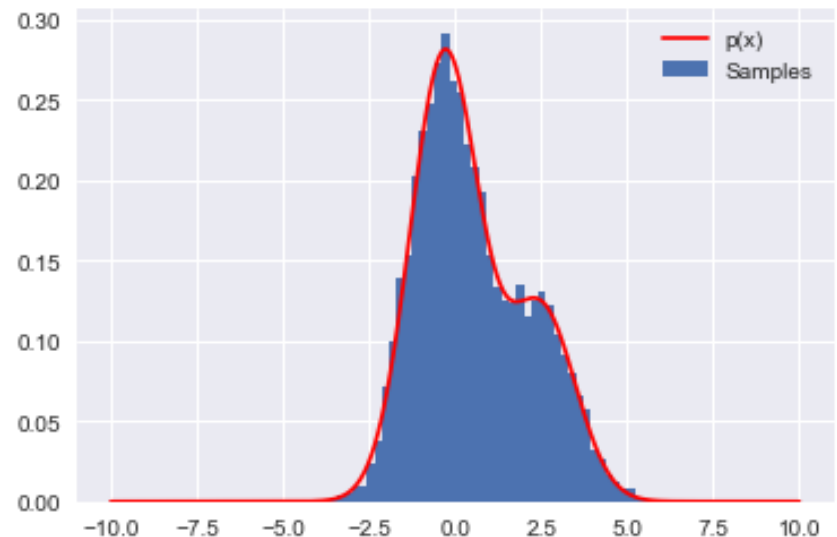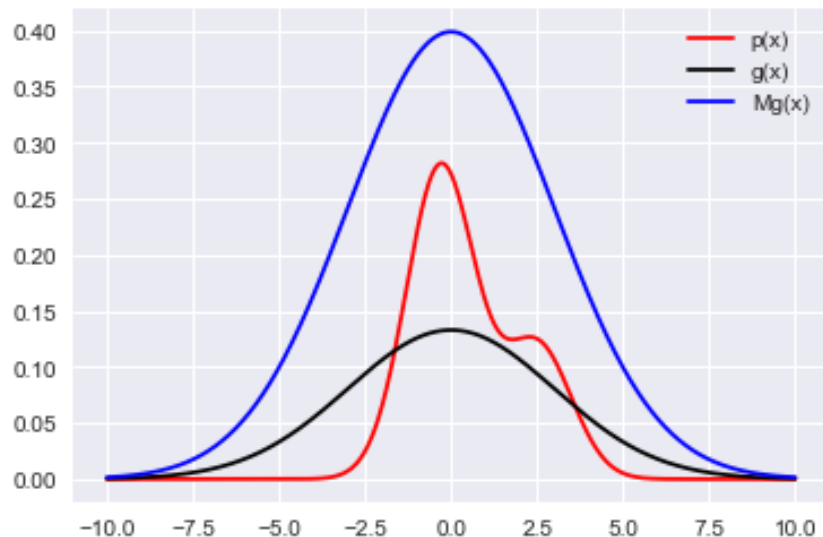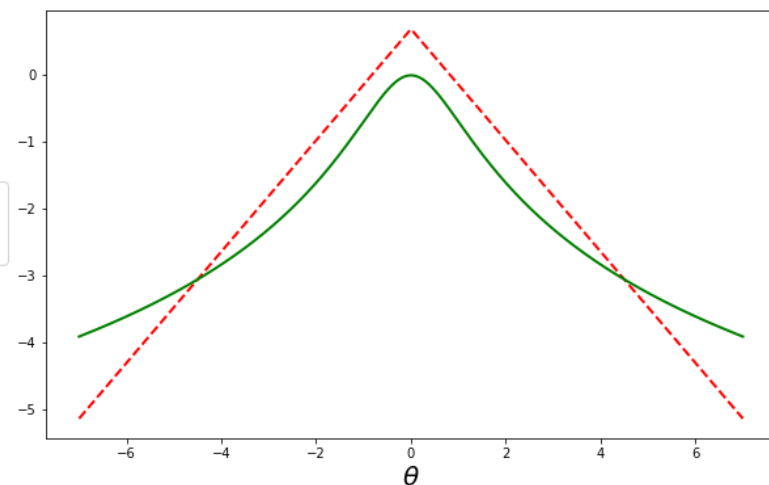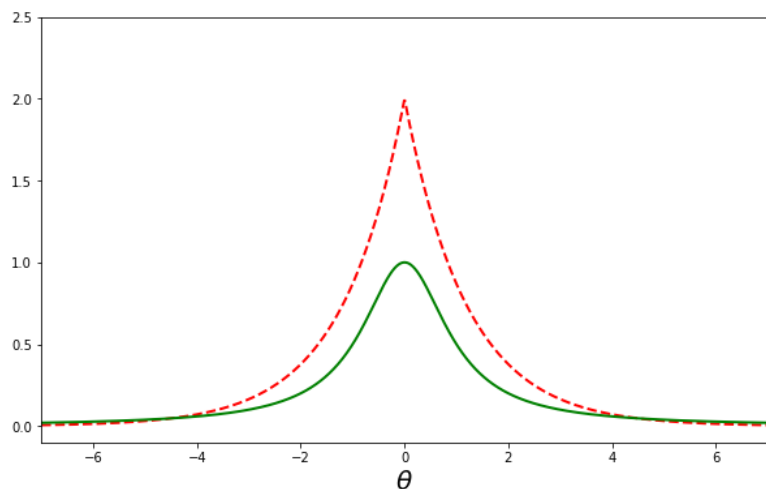
Important Aspects about Acceptance-Rejection Sampling

- Acceptance-rejection-sampling does not work if $M = \sup w(\theta)$ is not finite

$$w(\theta) = \frac{p(\theta)f(y|\theta)}{Mg_0(\theta)}$$

> ➢ This may happen if the candidate distribution $g_0(\theta)$ has lighter tails than the target
>
> ➢ **Take logarithms** of both the candidate density and the garget and graph them to see which has heavier tails

- Heavy-tailed candidate distributions such as *Student's t* with low degrees of freedom are recommended

- Acceptance-rejection-sampling is effective when
  - ➢ The candidate distribution has shape that is similar to the shape of the posterior

Direct Sampling: Sampling-Importance-Resampling algorithm

- Sampling-importance-resampling (SIR) is a two stage method for sampling from the posterior distribution $p(\theta|y)$ when all we know is its unscaled version $g(\theta|y) = p(\theta)f(\theta|y) \propto p(\theta|y)$

Select a large number of samples from the candidate density $g_0(\theta)$

Calculate the importance weight for each sample

Resample using the computed importance weight

Algorithm

1) Draw a large sample $\theta_1, \ldots, \theta_N \sim g_0(\theta)$ from the candidate density $g_0(\theta)$

2) Calculate the values of the unscaled posterior at each value

$$p(\theta_i)f(y|\theta_i) \text{ for } i = 1, \ldots, N$$

3) Calculate the values of the candidate density at each value of θ in the sample

$$g_0(\theta_i) \text{ for } i = 1, \ldots, N$$

4) Calculate the ratio of the unscaled posterior to the candidate density for each value of θ in the sample

$$r_i = \frac{p(\theta_i)f(y|\theta_i)}{g_0(\theta_i)}, 0 \leq w(\theta^*) \leq 1$$

5) Calculate the importance weights

$$w_i = \frac{r_i}{\sum r_i} = \frac{p(\theta_i)f(y|\theta_i)}{g_0(\theta_i)} \bigg/ \sum \frac{p(\theta_i)f(y|\theta_i)}{g_0(\theta_i)}$$

6) Resample from $\theta_1, \ldots, \theta_N$ using the weights $w_1, \ldots, w_N$ as the sampling probabilities

Direct Sampling: Sampling-Importance-Resampling algorithm

```python
p = lambda x: 0.7*norm.pdf(x,-0.3, 1)+0.3*norm.pdf(x,2.5, 1)
g = lambda x: 3*norm.pdf(x,0, 3) # our proposal pdf : N(0,2^2)

N = 100000 # the total of samples we wish to generate
n = 10000 # resample size
xproposal = norm.rvs(0, 3, size=N)
targetValue = p(xproposal)
candidateValue = g(xproposal)
ratio =  targetValue/candidateValue
importanceRatio = ratio/sum(ratio)
index = np.random.choice(N,size=n, replace=True, p=importanceRatio)
samples =xproposal[index]
```
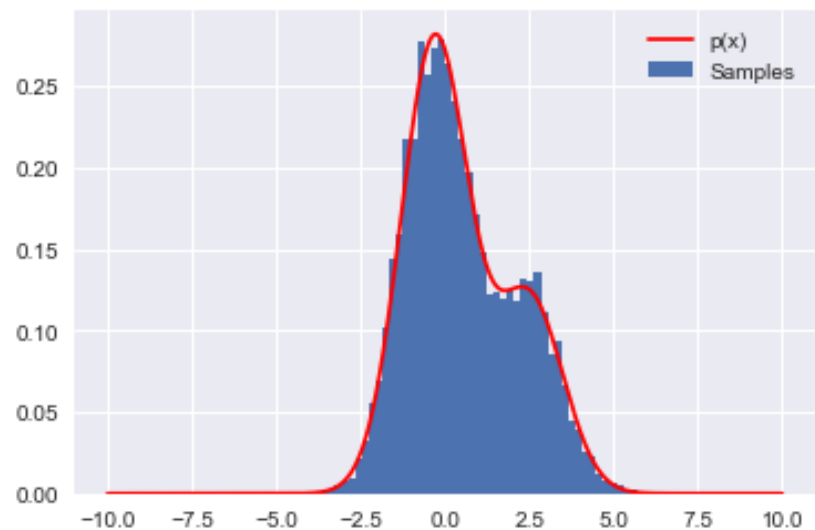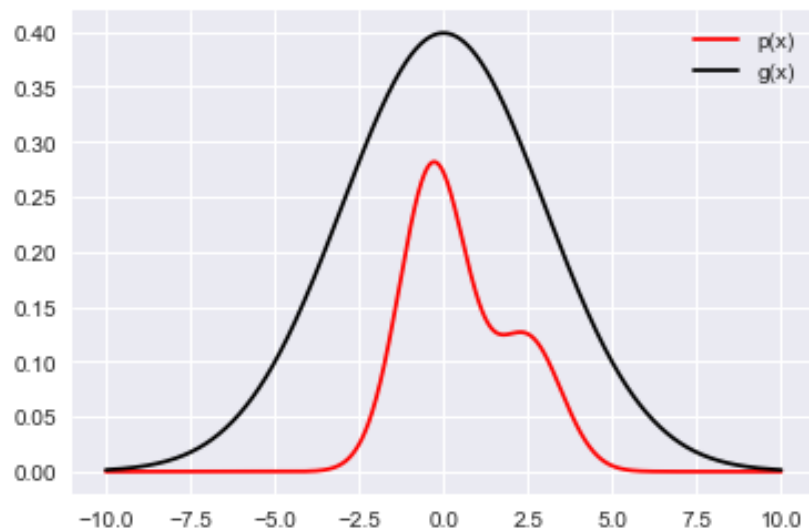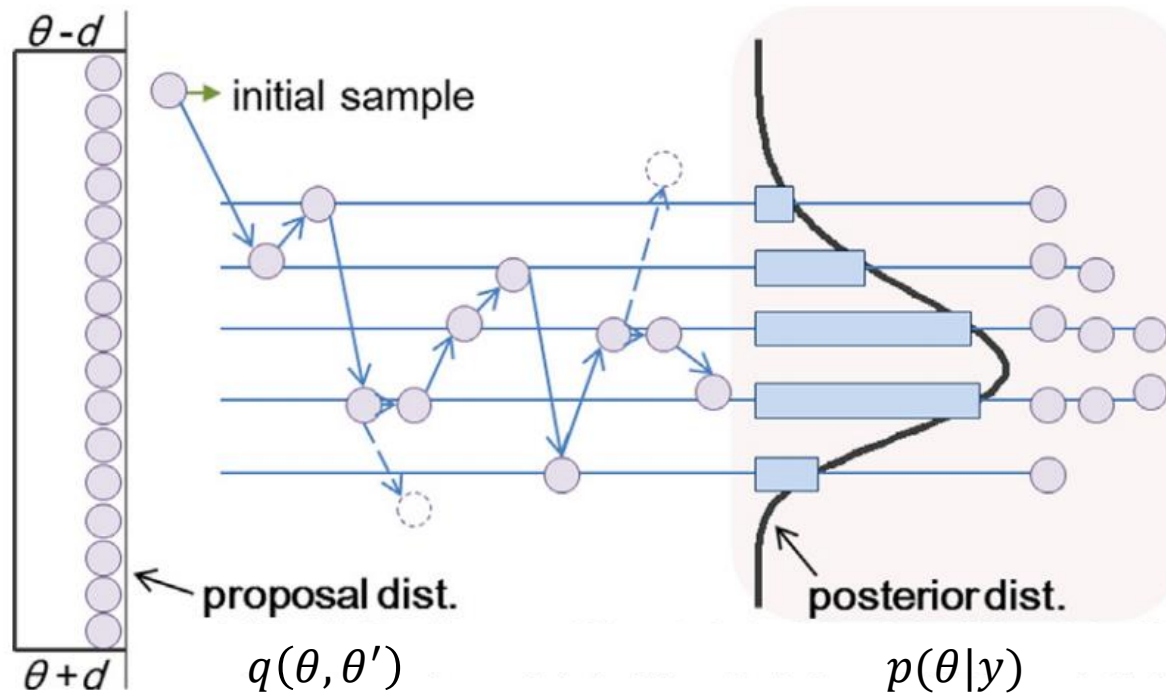
Fundamentals on Markov Chain

Markov Chain	Markov Chain Monte Carlo Sampling

- State-space θ
- Loong-run distribution $\pi(\theta) = P\pi(\theta)$

- Posterior density $p(\theta|y)$

$\theta - d$

initial sample

proposal dist.

$q(\theta, \theta')$

posterior dist.

$p(\theta|y)$

$\theta + d$

- Find the state transition matrix P such that Markov chain with P will have it's long-run stationary distribution $\pi(\theta)$ equals to posterior distribution $p(\theta|y)$

  ➢ Sampling from converged Markov chain is equal to sampling from posteriori distribution

Stochastic Process

- **A stochastic process** is a process that evolves through time according to **some probabilistic law**

- **The time points** when the process is allowed to change values
  - Discrete time stochastic process: the process can only change values at discrete values such as $t = 1, 2, \ldots$
  - Continuous time stochastic process: the process can change values at any time point $t \geq 0$

- The value that the process has at time t is called its **state** at time t

- The set of possible values that the stochastic process has at any particular time is called **the state space**

  - Finite discrete number of states: $S = \{1, 2, \ldots, m\}$

  - Infinite number of discrete states: $S = \{\ldots, -1, 0, 1, \ldots, \}$

  - Continuous state space of one dimension: $S = \{x: (-\infty < x < \infty)\}$

  - Continuous state space of p dimension: $S = \{(x_1, \ldots, x_p): (-\infty < x_i < \infty) \text{ for } i = 1, \ldots, p\}$

Markov Chain

- The mathematical model for a discrete stochastic process is a sequence of random variables $X^{(n)}$ for $n = 0,1,2,\dots$, where $X^{(n)}$ is the state at time n

- All knowledge about the probabilistic law for a stochastic process is contained in the joint probability distribution of every sequence of the random variables:

  - $P\big(X^{(0)} = x_0\big)$
  - $P\big(X^{(1)} = x_1, X^{(0)} = x_0\big)$
  - $P\big(X^{(n)} = x_n, \dots, X^{(0)} = x_0\big)$

- The joint probabilities of the process at times $0, 1, \dots, n$ can be built up sequentially from the previous joint probabilities and the conditional probabilities (transitional probabilities):

  - $P\big(X^{(1)} = x_1, X^{(0)} = x_0\big) = P\big(X^{(1)} = x_1 | X^{(0)} = x_0\big) \times P\big(X^{(0)} = x_0\big)$
  - $P\big(X^{(2)} = x_2, X^{(1)} = x_1, X^{(0)} = x_0\big) = P\big(X^{(2)} = x_2 | X^{(1)} = x_1, X^{(0)} = x_0\big) \times P\big(X^{(1)} = x_1, X^{(0)} = x_0\big)$
  - $P\big(X^{(n)} = x_n, \quad \dots, \quad X^{(0)} = x_0\big) = P\big(X^{(n)} = x_n | X^{(n-1)} = x_{n-1}, \dots, X^{(0)} = x_0\big) \times P\big(X^{(n-1)} = x_{n-1}, \dots, X^{(0)} = x_0\big)$

- The transition probabilities of a stochastic process generally depend on the entire past history of the process

$$P\big(X^{(n)} = x_n | \underline{X^{(n-1)} = x_{n-1}, \dots, X^{(0)} = x_0}\big)$$

Entire past history

- **A Markov chain** is a special kind of stochastic process: the conditional probabilities of the process at time n given the states at all previous times $n-1, \ldots, 0$ only depends on the single previous state at time $n-1$ as:

  ✓ $P\left(X^{(2)} = x_2 | X^{(1)} = x_1, X^{(0)} = x_0\right) = P\left(X^{(2)} = x_2 | X^{(1)} = x_1\right)$
  ✓ $P\left(X^{(3)} = x_3 | X^{(2)} = x_2, X^{(1)} = x_1, X^{(0)} = x_0\right) = P\left(X^{(3)} = x_3 | X^{(2)} = x_2\right)$
  ✓ $P\left(X^{(n)} = x_n | X^{(n-1)} = x_{n-1}, \ldots, X^{(0)} = x_0\right) = P\left(X^{(n)} = x_n | X^{(n-1)} = x_{n-1}\right)$

- The joint probability distributions of all the states up to time n can be built up sequentially by: (i.e., Factorization based on conditional independence)

✓ $P\left(X^{(1)} = x_1, X^{(0)} = x_0\right) = P\left(X^{(1)} = x_1 | X^{(0)} = x_0\right) \times P\left(X^{(0)} = x_0\right)$
✓ $P\left(X^{(2)} = x_2, X^{(1)} = x_1, X^{(0)} = x_0\right) = P\left(X^{(2)} = x_2 | X^{(1)} = x_1\right) \times P\left(X^{(1)} = x_1 | X^{(0)} = x_0\right) \times P\left(X^{(0)} = x_0\right)$

- If we are only interested in the process at time n:

$$P\left(X^{(n)} = x_n\right) = \sum_{x_{n-1}} P\left(X^{(n)} = x_n | X^{(n-1)} = x_{n-1}\right) \times P\left(X^{(n-1)} = x_{n-1}\right)$$

- The transition probabilities for a ***Markov Chains with Finite State Space*** can be represented:

$$\mathbf{P}^{[n-1,n]} = \begin{bmatrix} p_{1,1}^{[n-1,n]} & \cdots & p_{1,K}^{[n-1,n1]} \\ \vdots & \ddots & \vdots \\ p_{K,1}^{[n-1,n]} & \cdots & p_{K,K}^{[n-1,n]} \end{bmatrix}$$

✓ where $p_{i,j}^{[n-1,n]} = P\big(X^{(n)} = j | X^{(n-1)} = i\big)$ is the transition probability

✓ $\sum_j p_{i,j}^{[n-1,n]} = 1$

- ***Time-invariant (or homogeneous) Markov chain with Finite State Space***

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,K} \\ \vdots & \ddots & \vdots \\ p_{K,1} & \cdots & p_{K,K} \end{bmatrix}$$

✓ The transition probability only depend on the state, not the time n
✓ where $p_{i,j} = P(X = j | X = i)$ is the transition probability
✓ $\sum_j p_{i,j} = 1$

Higher Transition Probability Matrices

- The matrix of two-step transition probabilities is given by

$$\mathbf{P}^{[2]} = \begin{bmatrix} p_{1,1}^{[2]} & \cdots & p_{1,K}^{[2]} \\ \vdots & \ddots & \vdots \\ p_{K,1}^{[2]} & \cdots & p_{K,K}^{[2]} \end{bmatrix}$$

  ✓ where $p_{i,j}^{[2]} = P(X^{(n+2)} = j | X^{(n)} = i)$ is the probability of going from state i to state j in **two steps**, for all n, which can be computed as

$$P(X^{(n+2)} = j | X^{(n)} = i) = \sum_{k=1}^{K} P(X^{(n+2)} = j, X^{(n+1)} = k | X^{(n)} = i) \quad \text{(Marginalization)}$$

$$= \sum_{k=1}^{K} P(X^{(n+2)} = j | X^{(n+1)} = k, X^{(n)} = i) \times P(X^{(n+1)} = k | X^{(n)} = i)$$

$$= \sum_{k=1}^{K} P(X^{(n+2)} = j | X^{(n+1)} = k) \times P(X^{(n+1)} = k | X^{(n)} = i)$$

- Hence, the two-step transition probability is given by

$$p_{i,j}^{[2]} = \sum_{k=1}^{K} p_{i,k} \times p_{k,j} \Rightarrow \mathbf{P}^{[2]} = \mathbf{P} \times \mathbf{P} = \mathbf{P}^2$$

- $\mathbf{P}^{[m+n]} = \mathbf{P}^{[m]} \times \mathbf{P}^{[n]}, \ \mathbf{P}^{[n]} = \mathbf{P}^n$ (Generalization)

The occupation probability distribution

- The occupation probability distribution is given by

$$\boldsymbol{\alpha}^{(0)} = \left(\alpha_1^{(0)}, \dots, \alpha_m^{(0)}\right)$$

  ✓ where $\alpha_i^{(0)} = P\left(X^{(n)} = i\right)$ is the probability that the chain is in state i at time 0.

  ✓ For example,

$$P\left(X^{(1)} = j\right) = \sum_{i=1}^{K} P\left(X^{(1)} = j, X^{(0)} = i\right) = \sum_{i=1}^{K} P\left(X^{(0)} = i\right) \times P\left(X^{(1)} = j | X^{(0)} = i\right)$$

  ✓ In other words

$$\alpha_j^{(1)} = \sum_{i=1}^{K} \alpha_i^{(0)} \times p_{i,j} \quad \Rightarrow (\text{matrix form}) \; \boldsymbol{\alpha}^{(1)} = \boldsymbol{\alpha}^{(0)} \times \mathbf{P}$$

- Repeating this process back to the initial state we get

$$\boldsymbol{\alpha}^{(n)} = \boldsymbol{\alpha}^{(0)} \times \mathbf{P}^n$$

Long-run distribution of the chain and the steady state equation

- If a Markov chain processes a limiting distribution that is independent of the initial state (or initial distribution of states), it is called ***long-run distribution of the chain:***

$$\boldsymbol{\pi} = \lim_{n \to \infty} \boldsymbol{\alpha}^{(n)}$$

- When a chain does possess a limiting distribution,

$$\lim_{n \to \infty} \mathbf{I} \times \mathbf{P}^n = \lim_{n \to \infty} \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & 1 & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \times \mathbf{P}^n = \begin{bmatrix} \lim_{n \to \infty} \boldsymbol{\alpha}^{(0)} \times \mathbf{P}^n \\ \vdots \\ \lim_{n \to \infty} \boldsymbol{\alpha}'^{(0)} \times \mathbf{P}^n \end{bmatrix} = \begin{bmatrix} \lim_{n \to \infty} \boldsymbol{\alpha}^{(n)} \\ \vdots \\ \lim_{n \to \infty} \boldsymbol{\alpha}'^{(n)} \end{bmatrix} = \begin{bmatrix} \pi_1 & \cdots & \pi_K \\ \vdots & \ddots & \vdots \\ \pi_1 & \cdots & \pi_K \end{bmatrix} = \lim_{n \to \infty} \mathbf{P}^n$$

- From one step transition relationships, $\mathbf{P}^{[n+1]} = \mathbf{P}^{[n]} \times \mathbf{P}$, If we take $n \to \infty$, both $\mathbf{P}^{[n]}$ and $\mathbf{P}^{[n+1]}$ will approach the same limit. Hence

$$\begin{bmatrix} \pi_1 & \cdots & \pi_K \\ \vdots & \ddots & \vdots \\ \pi_1 & \cdots & \pi_K \end{bmatrix} = \begin{bmatrix} \pi_1 & \cdots & \pi_K \\ \vdots & \ddots & \vdots \\ \pi_1 & \cdots & \pi_K \end{bmatrix} \times \mathbf{P}$$

✓ Each row gives the same equation (**steady-state equation**)

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

✓ The steady state probability of entering state j from state i equals $\pi_i p_{i,j}$
✓ $\pi_j = \sum_{i=1}^{K} \pi_i p_{i,j}$: the probability flowing into each state at the steady state equals its steady state probability

Long-run distribution of the chain and the steady state equation

- It is possible to have a solution of the steady state equation, but not have a long-run distribution

$$\mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$(.5, .5) \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = (.5, .5)$$

- ✓ $(.5, .5)$ is a solution of the steady state equation for the Markov chain having the transition matrix $\mathbf{P}$

- ✓ However,

$$\mathbf{P}^{(n)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ when } n \text{ is odd} \quad \mathbf{P}^{(n)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ when } n \text{ is even}$$

- ➤ Thus, this chain does not have a long-run distribution as it just cycles back and forth between the two states

- A chain that only visits states on some multiple are called ***periodic*** Markov chains

- We will restrict ourselves to ***aperiodic*** Markov chins
    - ➤ after the chain has run long enough, it could be in any of the states at any time after

- The pair of states are said to ***communicate*** with each other when
  - ✓ state j is reachable from state i, i.e., $p_{i,j}^{[n]} > 0$ for some n,

- A set of states is called a ***communicating class of states*** when
  - ✓ Every pair of states inside the class communicate with each other
  - ✓ No state outside the set is reachable from inside the set

- A Markov chain is called ***irreducible*** when
  - ➤ The chain consists of a single communicating class

- The probability that the first passage is in step m is expressed as

$$f_{i,j}^{[m]} = P\left(X^{(m)} = j, X^{(m-1)} \neq j, \ldots, X^{(1)} \neq j \mid X^{(0)} = i\right)$$

- The first return probability is calculated recursively as

$$f_{i,i}^{[n]} = p_{i,i}^{[n]} - \sum_{m=1}^{n-1} f_{i,i}^{[m]} p_{i,i}^{[n-m]}$$

- A state is called ***positive recurrent*** when

  ✓ $\sum_n f_{i,i}^{[n]} = 1$

  ✓ The mean first return time $\sum_n n f_{i,i}^{[n]} < \infty$

  ➤ $f_{i,i}^{[n]}$ must go to zero at a faster rate than n^{-2}

  ✓ the chain is certain to return to state i

  ➤ the chain is certain to return to state i infinitely often

- If a Markov chain is

  ➤ irreducible,
  ➤ Aperiodic
  then, all states are positive recurrent

- A Markov chain is called *ergodic* when the chain is
  - ✓ Irreducible
  - ✓ Aperiodic

- If a chain is *ergodic*,
  - ✓ The chain has states that are all positive recurrent
  - ✓ The time average of a single realization approach the average of all possible realization of the same Markov chain at some particular time point
    - ➢ **We can estimate long-run probabilities for this type chain by taking the time average of a single realization of the chain**

Theorem

In an ***ergodic Markov chain*** the limits

$$u_j = \lim_{n \to \infty} p_{i,j}^{[n]} > 0$$

exists and are independent of the initial state i. Also

$$\sum_j u_j = 1$$

and

$$u_j = \sum_i u_i p_{i,j}$$

Conversely, if the chain has $u_j > 0$ satisfying the conditions above, then the chain is ergodic and $u_j = 1/m_j$, the reciprocal of the mean return time

- Irreducible
- Aperiodic

→ **MC is ergodic** →

A unique non-zero solution of the steady state equation

Time averages from an ergodic Markov chain will approach the steady state probabilities of the chain

- The occupation probability distribution for a Markov chain at time n actually represents the average over the ensemble, which is the set of all possible Markov chains with those transition probabilities.

- The long-run distribution of a Markov chain describes the probability that the chain is in each particular state after the chain has been running a long time
  - ➢ The long-run distribution for an ergodic chain can be thought of the average over the ensemble at some time n a long time in the future
  - ➢ We are going to take a sample from a single realization of the Markov chain after we have let it run a period of time called the burn-in time

Time-Reversible Markov Chains and Detailed Balance

- So far we found the long-run distribution from the given transition probabilities:

Transition probabilities $p_{i,j}$ for all values of i and j	⟶	Long-run distribution π for all values of i

- Starting with the long-run distribution, Markov Chain Monte Carlo Simulation finds a Markov chain with that distribution:
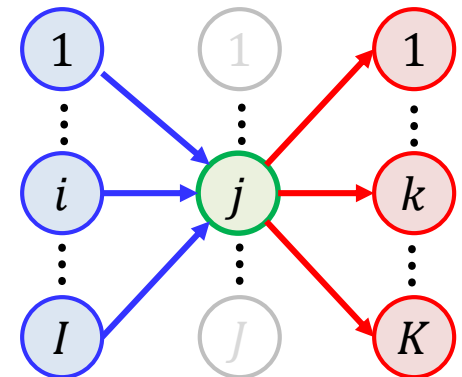
Long-run distribution π for all values of i	⟶	Transition probabilities $p_{i,j}$ for all values of i and j

✓ First transition matrix satisfies:

$$\sum_j p_{i,j} = 1$$

✓ When a chain is at steady state, the total amount of probability flowing out of state j must balance the total flow of probability going into state j:

$$\sum_i \pi_i p_{i,j} = \sum_k \pi_j p_{j,k}$$

➤ This **total balance** must be true for all states j

Time-Reversible Markov Chains and Detailed Balance

- The transition probabilities for the backwards chain be

$$q_{i,j} = P\big(X^{(n)} = j | X^{(n+1)} = i\big) = \frac{P\big(X^{(n)} = j, X^{(n+1)} = i\big)}{P(X^{(n+1)} = i)} = \frac{P\big(X^{(n)} = j\big)P\big(X^{(n+1)} = i | X^{(n)} = j\big)}{P(X^{(n+1)} = i)}$$

- When the chain is at steady state:

$$q_{i,j} = \frac{\pi_j p_{j,i}}{\pi_i}$$

- Markov chain is said to be ***time reversible*** when the backwards Markov chain and the forward Markov chain have the same transition probabilities,

$$q_{i,j} = p_{i,j}$$

- Then, it satisfy

$$\pi_i p_{i,j} = \pi_j p_{j,i} \text{ for all } i \text{ and } j$$

- ✓ This is called "***detailed balance***"
- ✓ For every pair of sates the flows between the two states balance each other
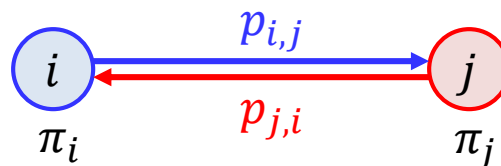
Theorem

A set of transition probabilities satisfying the detailed balance condition will have steady state distribution $\boldsymbol{\pi}$

Proof:

$$\sum_i \pi_i p_{i,j} = \sum_i \pi_j p_{j,i} \qquad \because \pi_i p_{i,j} = \pi_j p_{j,i}$$

$$= \pi_j \sum_i p_{j,i}$$

$$= \pi_j \sum_j p_{i,j} \qquad \because p_{j,i} = p_{i,j} \text{ due to the reversibility}$$

$$= \pi_j$$

✓ This hold for all j
✓ Thus, the Markov chain with transition matrix that satisfies the detailed balance condition has the steady state distribution $\boldsymbol{\pi}$

- Thus, to find a Markov chain that has the desired steady state probabilities, we have to find the transition probabilities of a Markov chain that satisfies the detailed balance condition.

| Given: $\boldsymbol{\pi}$ | → | *Find $\mathbf{P}$ satisfying the detailed balance condition with $\boldsymbol{\pi}$* | → | Markov Chain with $\mathbf{P}$ will have steady state $\boldsymbol{\pi}$ |
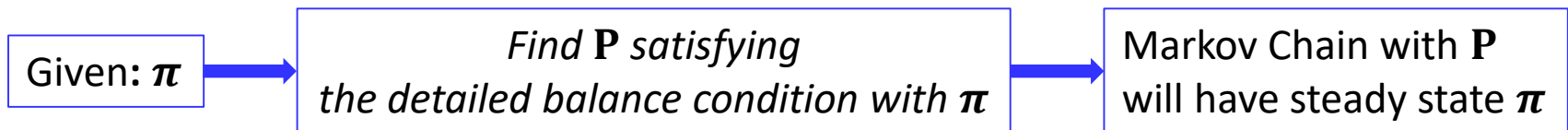
The Metropolis Algorithm

- Metropolis et al. (1953) discovered an algorithm that **finds transition probabilities** for a Markov chain that will yield the **desired steady state distribution**

- Starting with **an arbitrary set of transition probabilities**, the algorithm show how to construct a new set of transition probabilities that satisfies *the detailed balance condition*
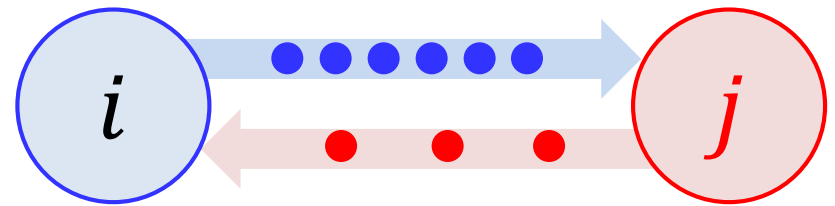   - ✓ This is done by only accepting some of the transitions

$$\pi_i p_{i,j} < \pi_j p_{j,i} \qquad\qquad \pi_i p_{i,j} > \pi_j p_{j,i}$$
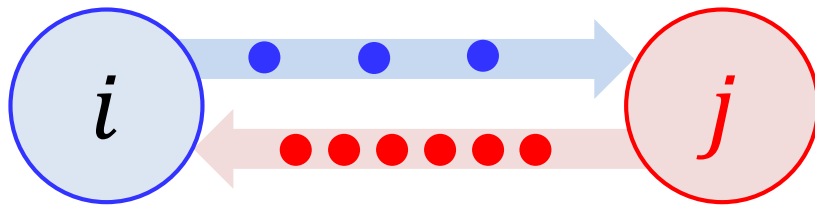
Too much flow from j to i
- ✓ Accept all the transitions from i to j
- ✓ Accept only some of the transitions from j to i

Too much flow from i to j
- ✓ Accept all the transitions from j to i
- ✓ Accept only some of the transitions from i to j

$$\pi_i p_{i,j} = \pi_j p_{j,i}$$
Balance!

The Metropolis Algorithm

Algorithm

1) Start from the transition probabilities $p_{i,j}$ and the desired steady state probabilities π_i

2) For each pair of states i and j, define the acceptance probability

$$\alpha_{i,j} = \min\left[\frac{\pi_j p_{j,i}}{\pi_i p_{i,j}}, 1\right]$$

3) Then for each i and j let

$$p'_{i,j} = \alpha_{i,j} p_{i,j}$$

4) Adjust self transition probability

$$p'_{i,i} = 1 - \sum_{j \neq i} p'_{i,j}$$

- Then $\boldsymbol{\pi}$ is the steady state distribution for the Markov chain with transition probabilities given by $p'_{i,j}$

Theorem

The Markov chain having transition probabilities given by $p'_{i,j}$ satisfies the detailed balance condition, and thus it has the desired steady state distribution

Proof:

- First, we note that these are transition probabilities since $\sum_j p'_{i,j} = 1$ for all i
- Then, we note that

$$\pi_i p'_{i,j} = \pi_i \alpha_{i,j} p_{i,j} \qquad\qquad \pi_j p'_{j,i} = \pi_j \alpha_{j,i} p_{j,i}$$

$$= \pi_i \min\left[\frac{\pi_j p_{j,i}}{\pi_i p_{i,j}}, 1\right] p_{i,j} \qquad\qquad = \pi_j \min\left[\frac{\pi_i p_{i,j}}{\pi_j p_{j,i}}, 1\right] p_{j,i}$$

$$= \min\left[\pi_j p_{j,i}, \pi_i p_{i,j}\right] \qquad\qquad = \min\left[\pi_i p_{i,j}, \pi_j p_{j,i}\right]$$

- This satisfies the detailed balance condition, so the Markov chain with transition probabilities given by $p'_{i,j}$ has the desired steady state distribution $\boldsymbol{\pi}$

The Metropolis Algorithm: Example

- Suppose we have a transition matrix given by:

$$\mathbf{P} = \begin{bmatrix} .1 & .2 & .3 & .4 \\ .1 & .2 & .3 & .4 \\ .3 & .4 & .2 & .1 \\ .3 & .4 & .2 & .1 \end{bmatrix}$$

- We want to achieve the steady state distribution $\boldsymbol{\pi} = (.2, \; .3, \; .4, \; .1)$

- Acceptance probability can be computed as $\alpha_{i,j} = \min\left[\dfrac{\pi_j p_{j,i}}{\pi_i p_{i,j}}, 1\right]$

The Metropolis Algorithm: Example

- Suppose we have a transition matrix given by:

$$\mathbf{P} = \begin{bmatrix} .1 & .2 & .3 & .4 \\ .1 & .2 & .3 & .4 \\ .3 & .4 & .2 & .1 \\ .3 & .4 & .2 & .1 \end{bmatrix}$$

- We want to achieve the steady state distribution $\boldsymbol{\pi} = (.2,\ .3,\ .4,\ .1)$

- Acceptance probability can be computed as $\alpha_{i,j} = \min\left[\dfrac{\pi_j p_{j,i}}{\pi_i p_{i,j}}, 1\right]$

$$\alpha_{1,2} = \min\left[\frac{.3\times.1}{.2\times.2}, 1\right] = \frac{3}{4} \qquad \alpha_{1,3} = \min\left[\frac{.4\times.3}{.2\times.3}, 1\right] = 1 \qquad \alpha_{1,4} = \min\left[\frac{.1\times.3}{.2\times.4}, 1\right] = \frac{3}{8}$$

$$\alpha_{2,1} = \min\left[\frac{.2\times.2}{.3\times.1}, 1\right] = 1 \qquad \alpha_{2,3} = \min\left[\frac{.4\times.4}{.3\times.2}, 1\right] = 1 \qquad \alpha_{2,4} = \min\left[\frac{.1\times.4}{.3\times.4}, 1\right] = \frac{1}{3}$$

$$\alpha_{3,1} = \min\left[\frac{.2\times.3}{.4\times.3}, 1\right] = \frac{1}{2} \qquad \alpha_{3,2} = \min\left[\frac{.3\times.3}{.4\times.4}, 1\right] = \frac{9}{16} \qquad \alpha_{3,4} = \min\left[\frac{.1\times.2}{.4\times.1}, 1\right] = \frac{1}{2}$$

$$\alpha_{4,1} = \min\left[\frac{.2\times.4}{.1\times.3}, 1\right] = 1 \qquad \alpha_{4,2} = \min\left[\frac{.3\times.4}{.1\times.4}, 1\right] = 1 \qquad \alpha_{4,3} = \min\left[\frac{.4\times.1}{.1\times.2}, 1\right] = 1$$

- Then for each $j \neq i$ let $p'_{i,j} = \alpha_{4,1}\times p_{i,j}$, and let $p'_{i,i} = 1 - \sum_{j\neq i} p'_{i,j}$

$$\mathbf{P'} = \begin{bmatrix} .400 & .150 & .300 & .150 \\ .100 & .467 & .300 & .133 \\ .150 & .225 & .575 & .050 \\ .300 & .400 & .200 & .100 \end{bmatrix} \Rightarrow \boldsymbol{\pi} = (.2,\ .3,\ .4,\ .1)$$

Markov Chains with Continuous State Space

- Sometimes the state of a Markov chain are continuous

- The Markov chain has a one-dimensional continuous state space
  - ➢ The state space consists of all possible values in an interval

- The Markov chain has a p-dimensional continuous state space:
  - ➢ The state space consists of all possible values in a rectangular region of dimension p

- The probability of a transition between all but a countable number of pairs of possible value must be zero

- Thus, we define the transition probabilities from each possible state x to each possible measurable set of sates A:

$$P(x_0, A) = P\big(X^{(n+1)} \in A | X^{(n)} = x_0\big)$$

  - ✓ Called the transition kernel of the Markov chain for all values of x and all possible measureable sets A

- In the single dimensional case, the probability of a measurable set A can be found from the probability of the transition CDF which is

$$F(v|x) = P\big(X^{(n+1)} < v | X^{(n)} = x\big)$$

- When it is absolutely continuous we can take the derivative

$$f(v|x) = \frac{\partial P\big(X^{(n+1)} < v | X^{(n)} = x\big)}{\partial v}$$

  ✓ Which is the probability density function of the one-step transition

- The probability density of two-step transitions would be given by

$$f^{[2]}(v|x) = \int_{-\infty}^{\infty} f(v|w)\, f(w|x)dw$$

- If the Markov chain possesses a limiting transition density independent of the initial state,

$$\lim_{n\to\infty} f^{[n]}(v|x) = g(v)$$

  ✓ $g(v)$, which does not depends on x, is called ***the steady state density*** of the Markov chain and is the solution of the steady state equation

$$g(v) = \int_{-\infty}^{\infty} g(w)f(v|w)\, dw \qquad\qquad : \boldsymbol{\pi} = \boldsymbol{\pi} \times \mathbf{P}$$

Markov Chains with Continuous State Space

- The main results we found for discrete Markov chains continue to hold for Markov chains with continuous state space:

  ✓ All states in an ergodic (irreducible and aperiodic) Markov chain are positive recurrent if and only if there is a unique non-zero solution to the steady state equation

$$g(v) = \int_{-\infty}^{\infty} g(w)f(v|w)\,dw$$

  ✓ The time average of a single realization of an ergodic Markov chain approaches the steady state distribution that is the average of the ensemble (all possible realizations of the chain) at some fixed time point far in the future

  ✓ *Time reversible Markov chains* satisfy *the detailed balance condition*. Let $g(x)$ be the steady state density and $f(v|x)$ be the density function of the one-step transitions

$$g(x)f(v|x) = g(v)f(x|v) \qquad\qquad : \pi_i p_{i,j} = \pi_j p_{j,i}$$

  for all states x and v

Markov Chain Monte Carlo Sampling form Posterior

- Direct Monte Carlo sampling from the posterior using the methods (acceptance-rejection, sampling-importance resampling) is inefficient (require a huge number of samples) when

  - ✓ There are many parameters
    - ➢ high dimensional parameter space
    - ➢ a large number of parameters from hierarchical model

  - ✓ The prior is very diffuse compared to the likelihood

- Markov Chain Monte Carlo (MCMC) is methods for generating samples using an unscaled posterior distribution:
  - ✓ It does not draw samples from the posterior distribution directly
  - ✓ It setup a **Marko chain that has the posterior distribution as its limiting distribution**
  - ✓ Examples includes the Metropolis-Hasting algorithm, Gibbs sampler, substation sampler

- An approximately random sample from the posterior distribution can be found by taking values from a single run of the Markov chain at widely spaced time points after the initial burn-in time

$p(\theta|y)$: The posterior distribution on the parameter θ given the observation y

$\boldsymbol{\pi}$: The steady state distribution of the Markov chain with the transition probabilities $\mathbf{P}$

- The parameter space will be the state space of the Markov chain

- How to find $\mathbf{P}$?
  - ➤ The long-run distribution $\boldsymbol{\pi}$ of a finite ergodic Markov chain is a solution of the steady state equation:

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

  - ➤ The long−run distribution $\pi(\theta)$ of a continous state space Markov chain satisfies :

$$\int_A \pi(\theta)\, d\theta \quad = \quad \int \pi(\theta)\, P(\theta, A) d\theta \quad \text{for all} \quad A$$

The total steady state probability of set A The steady state flow of probability into set A

- We will set the long-run distribution $\pi(\theta)$ for the Markov chain equal to the posterior density $p(\theta|y)$

- We need to find a probability transition kernel $P(\theta, A)$ that satisfies

$$\int_A \pi(\theta)\, d\theta \quad = \quad \int \pi(\theta)\, P(\theta, A) d\theta \quad \text{for all} \quad A$$

$$\int_A p(\theta|y) d\theta = \int p(\theta|y) P(\theta, A) d\theta \quad \text{for all} \quad A$$

✓ Generally we will use an unscaled posterior density $g(\theta|y) = p(\theta)f(y|\theta) \propto p(\theta|y)$

- **A finite ergodic Markov chain:**
  - ➤ *A set of transition probabilities **P** satisfying the detailed balance condition will have steady state distribution **π***

- **A continuous state-space ergodic Markov chain :**
  - ➤ *If a transition kernel $P(\theta, A)$ balance the steady state flow between every possible pair of states, then the chain will have the long-run distribution $p(\theta|y)$*

Theorem

*Let $q(\theta, \theta')$ be a candidate distribution that generates a candidate θ' given starting value θ. If for all θ and θ', the candidate distribution $q(\theta, \theta')$ satisfies **the reversibility condition (detailed balance)**,*

$$p(\theta|y) \times q(\theta, \theta') = p(\theta'|y) \times q(\theta', \theta) \text{ for all } \theta, \theta'$$

Then, $p(\theta|y)$ is the long-run distribution for the Markov Chain with probability kernel

$$P(\theta, A) = \int_A q(\theta, \theta')d\theta' + r(\theta)\delta_A(\theta)$$

where $r(\theta) = 1 - \int_A q(\theta, \theta')d\theta'$ is the probability the chain remains at θ, and where $\delta_A(\theta)$ is the indicator function of set A

$$\delta_A(\theta) = \begin{cases} 1 & \text{if } \theta \in A \\ 0 & \text{if } \theta \notin A \end{cases}$$
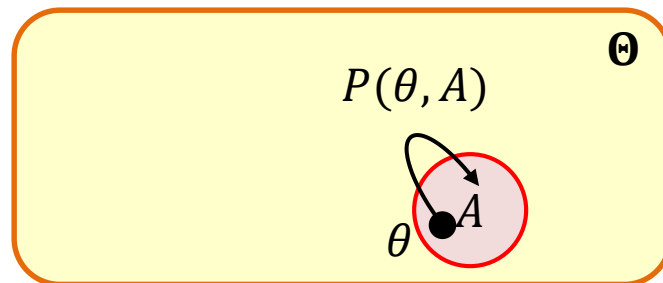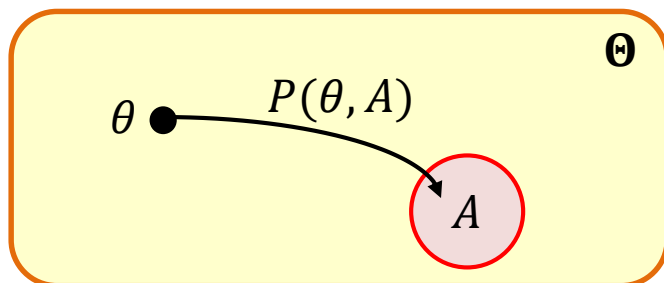
Proof:

$$\int p(\theta|y)P(\theta,A)d\theta = \int p(\theta|y)\left\{\int_A q(\theta,\theta')d\theta' + r(\theta)\delta_A(\theta)\right\}d\theta$$

$$= \int\int_A p(\theta|y)q(\theta,\theta')d\theta'\,d\theta + \int p(\theta|y)\,r(\theta)\delta_A(\theta)d\theta$$

$$= \int_A\int p(\theta|y)q(\theta,\theta')d\theta'\,d\theta + \int_A p(\theta|y)r(\theta)d\theta$$

$$= \int_A\int p(\theta'|y)q(\theta',\theta)d\theta d\theta' + \int_A p(\theta|y)r(\theta)d\theta$$

$$= \int_A p(\theta'|y)\big(1 - r(\theta')\big)d\theta' + \int_A p(\theta|y)r(\theta)d\theta$$

$$= \int_A p(\theta'|y)\,d\theta'$$

$$\because p(\theta|y) \times q(\theta,\theta') = p(\theta'|y) \times q(\theta',\theta) \text{ for all } \theta,\theta' \text{ (detailed balance)}$$

Metropolis-Hasting Algorithm

- Most candidate distributions don't satisfy *the detailed balance condition*:

$$p(\theta|y) \times q(\theta, \theta') \neq p(\theta'|y) \times q(\theta', \theta) \text{ for some } \theta \text{ and } \theta'$$

  ✓ The probability of moving from θ to θ' *is not the same as the probability of moving in the reverse direction*

- Metropolis et al. (1953) supplied the solution. They resorted the balance by introducing a probability of moving (acceptance probability):

$$\alpha_{i,j} = \min\left[\frac{\pi_j p_{j,i}}{\pi_i p_{i,j}}, 1\right] \quad \Rightarrow \quad \alpha(\theta, \theta') = \min\left[\frac{p(\theta'|y)q(\theta', \theta)}{p(\theta|y)q(\theta, \theta')}, 1\right]$$

  A finite state MC   A continuous state-space MC

- We do not need to know the exact posterior $p(\theta|y)$ but **can use unscaled posterior** $g(\theta|y)$

$$\alpha(\theta, \theta') = \min\left[\frac{\cancel{k}g(\theta'|y)q(\theta', \theta)}{\cancel{k}g(\theta|y)q(\theta, \theta')}, 1\right]$$

Metropolis-Hasting Algorithm

Algorithm

1) Start at an initial $\theta^{(0)}$

2) Do for $n = 1, \dots, N$

    (a) Draw θ' from $q\left(\theta^{(n-1)}, \theta'\right)$

    (b) Calculate the accept probability

$$\alpha\left(\theta^{(n-1)}, \theta'\right) = \min\left[\frac{g(\theta'|y)q\left(\theta', \theta^{(n-1)}\right)}{g\left(\theta^{(n-1)}|y\right)q\left(\theta^{(n-1)}, \theta'\right)}, 1\right]$$

    (c) Draw from $U(0,1)$

    (d) If $u < \alpha\left(\theta^{(n-1)}, \theta'\right)$ then let $\theta^{(n)} = \theta'$, else let $\theta^{(n)} = \theta^{(n-1)}$

Note:

- Having the candidate density $q(\theta, \theta')$ close to the target $g(\theta'|y)$ leads to more candidates being accepted
- In fact, when the candidate density is exactly the same shape as the target

$$q(\theta, \theta') = k \times g(\theta'|y)$$

- Then the acceptance probability is

$$\alpha\left(\theta^{(n-1)}, \theta'\right) = \min\left[\frac{g(\theta'|y)q(\theta', \theta)}{g(\theta|y)q(\theta, \theta')}, 1\right] = \min\left[\frac{g(\theta'|y)kg(\theta|y)}{g(\theta|y)kg(\theta'|y)}, 1\right] = 1$$

Single parameter with a random-walk candidate density

- For a random-walk candidate generating distribution the candidate is drawn from a symmetric distribution centered at the current value as:

$$q(\theta, \theta') = q_1(\theta' - \theta)$$

✓ where q_1 is a function symmetric about 0.

- For a random-walk candidate generating distribution the candidate is drawn from a symmetric distribution centered at the current value as:

$$\alpha(\theta, \theta') = \min\left[\frac{p(\theta'|y)q(\theta', \theta)}{p(\theta|y)q(\theta, \theta')}, 1\right] = \min\left[\frac{p(\theta'|y)}{p(\theta|y)}, 1\right]$$

- If $p(\theta'|y) > p(\theta|y)$, a candidate θ' will be always accepted
  - ➤ The chain will always move "uphill"

- If $p(\theta'|y) < p(\theta|y)$, a candidate θ' will be accepted with a probability less than 1
  - ➤ There is a certain chance that the chain will move "downhill"
  - ➤ This allows a chain to move around the whole parameter space over time

- Will generally have many accepted candidates, but most of the moves will be a short distance
  - ➤ It might take a long time to move around the whole parameter space

Single parameter with a random-walk candidate density : Example

Target: $p(\theta|y)$

Proposal
$q(\theta^{(n-1)}, \theta') = q_1(\theta' - \theta^{(n-1)})$

$\theta^{(n-1)}$

Single parameter with a random-walk candidate density : Example

Target: $p(\theta|y)$

Proposal
$q(\theta^{(n-1)}, \theta') = q_1(\theta' - \theta^{(n-1)})$

θ' $\theta^{(n-1)}$

1. Draw θ' from $q(\theta^{(n-1)}, \theta')$

Single parameter with a random-walk candidate density : Example

Target: $p(\theta|y)$

θ' $\theta^{(n-1)}$

2. Calculate the acceptance probability

$$\alpha\big(\theta^{(n-1)}, \theta'\big) = \min\left[\frac{p(\theta'|y)}{p\big(\theta^{(n-1)}|y\big)}, 1\right]$$

Single parameter with a random-walk candidate density : Example

Target: $p(\theta|y)$

θ' $\quad\theta^{(n-1)}$
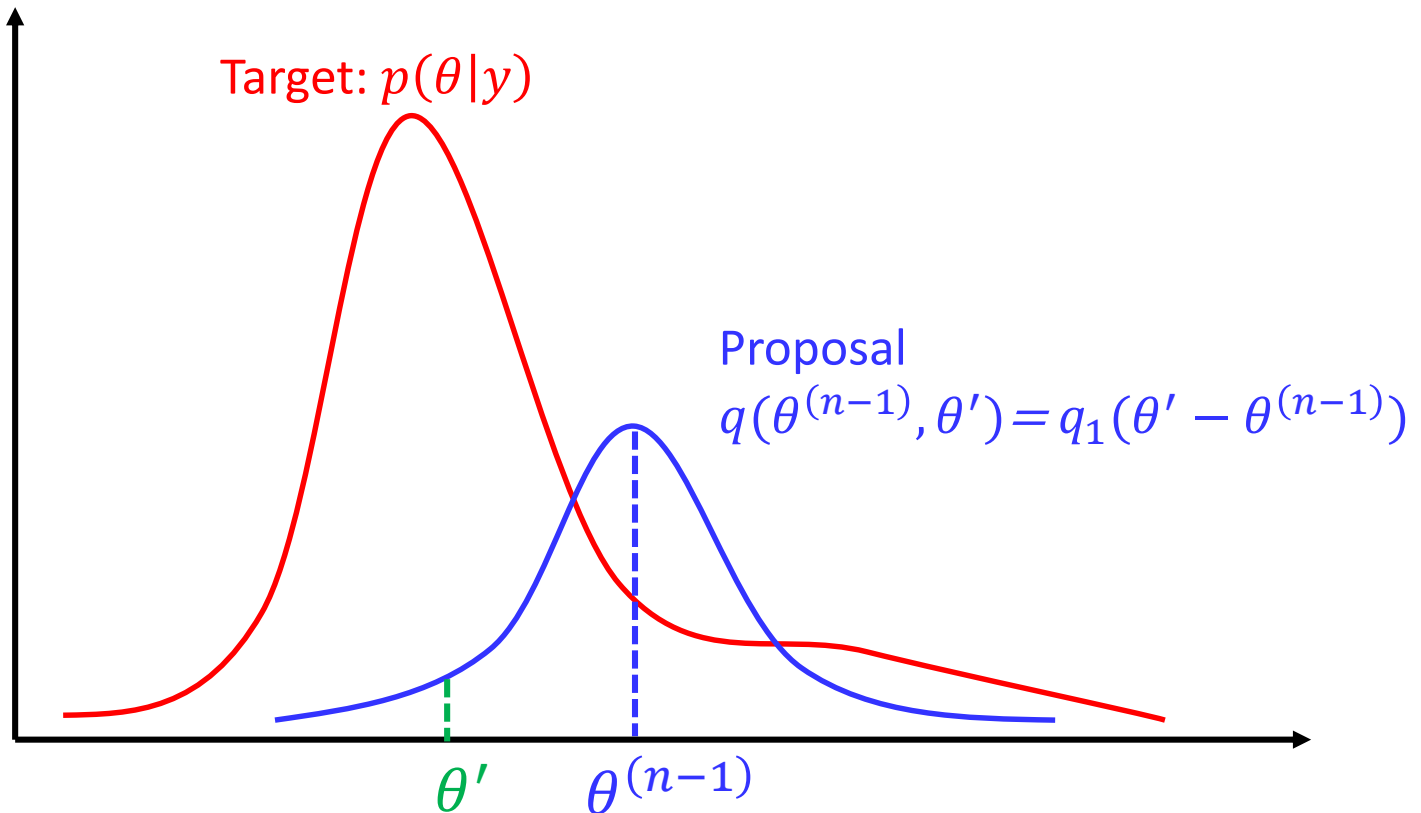
3. Accept θ' based on $\alpha\big(\theta^{(n-1)},\theta'\big)$

- because $p(\theta'|y) > p\big(\theta^{(n-1)}|y\big)$, a candidate θ' will be always accepted
  - ➤ The chain will always move "uphill"

Single parameter with a random-walk candidate density : Example

Target: $p(\theta|y)$

Proposal
$q(\theta^{(n-1)}, \theta') = q_1(\theta' - \theta^{(n-1)})$

$\theta^{(n)}$

4. Accept $\theta' = \theta^{(n)}$

Single parameter with a random-walk candidate density : Example

$p(\theta|y)$

$q_2(\theta' - \theta)$

$p(\theta'|y) > p(\theta|y)$

$p(\theta'|y) < p(\theta|y)$

θ' θ

θ θ'

θ

t
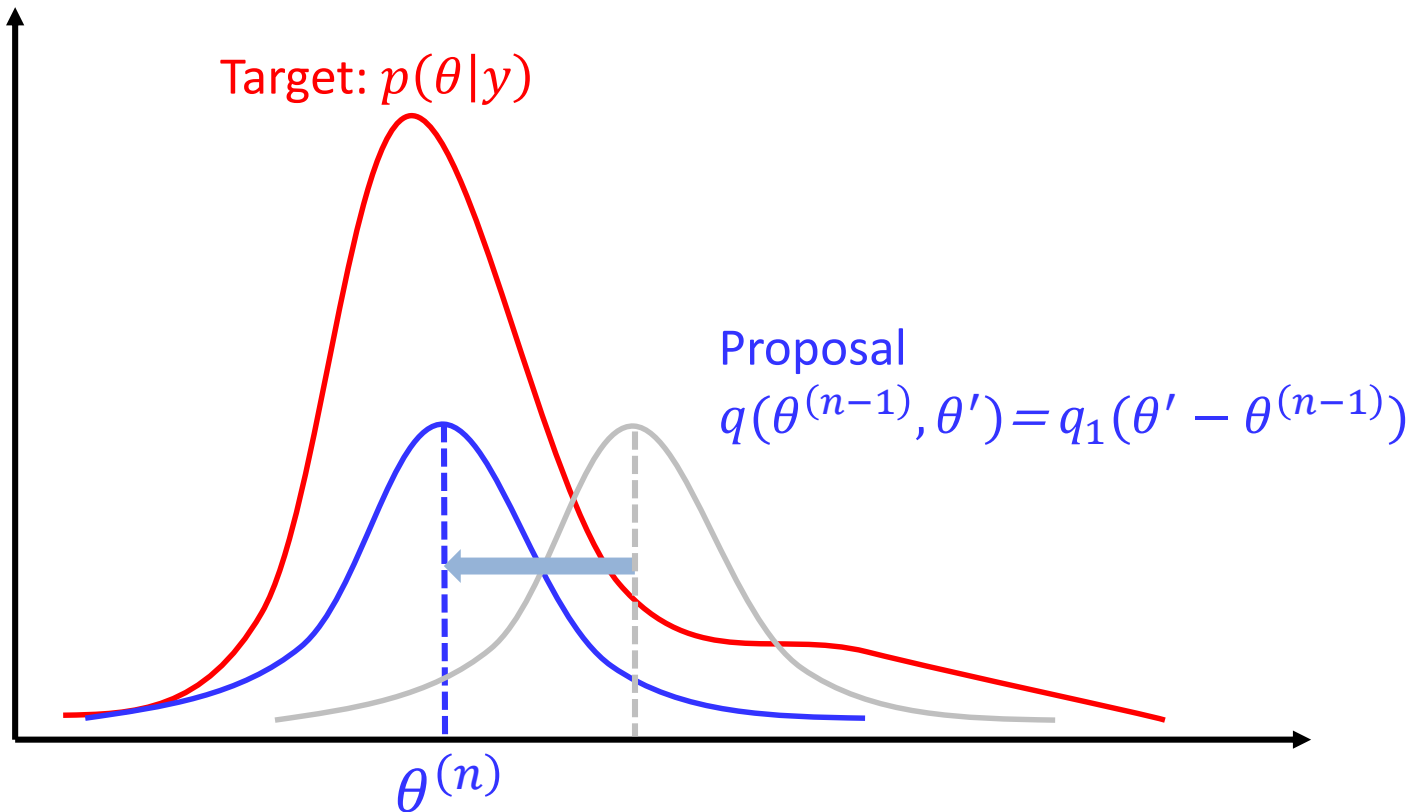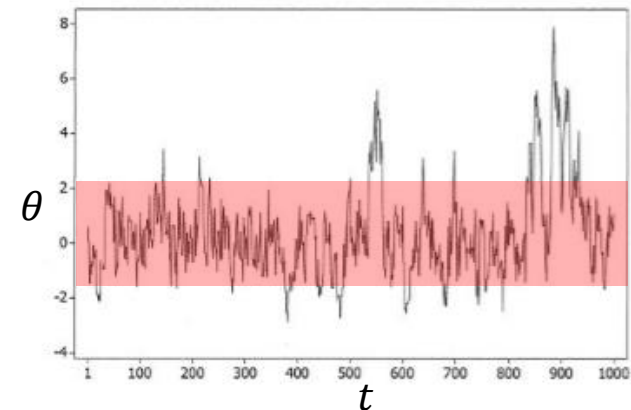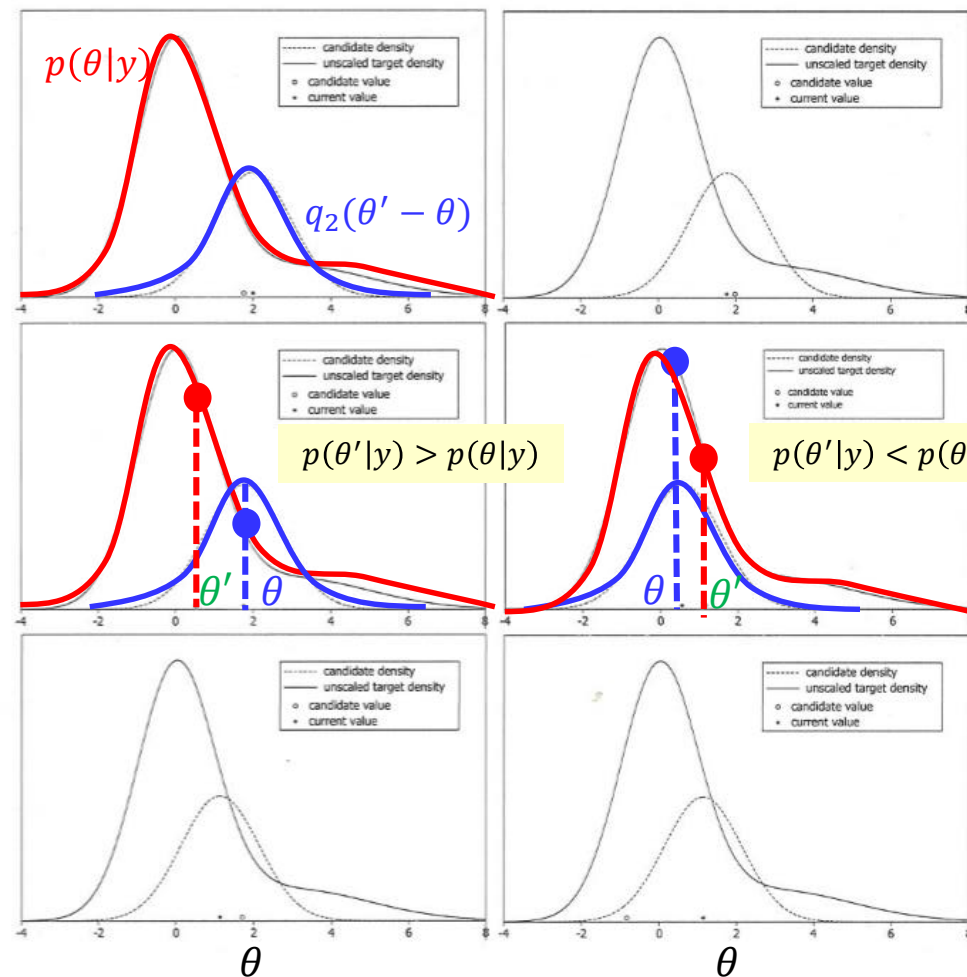
Small jump (slow mixing and highly correlated)
- More accepted
- If the long-run distribution has multiple model, it will take long time to reach
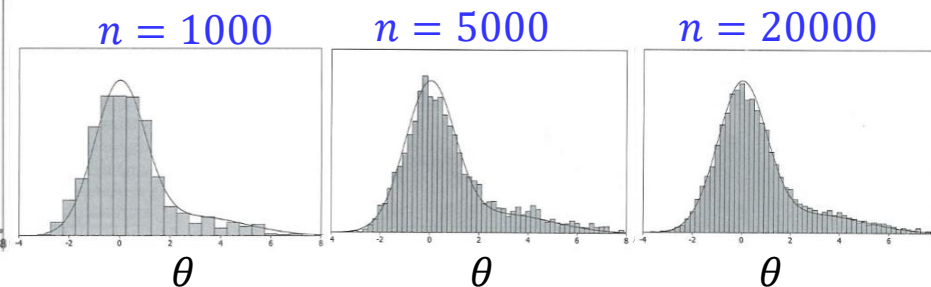
$n = 1000$ $n = 5000$ $n = 20000$

θ θ θ

Single parameter with an independent candidate density

- Hasting (1970) introduced Markov chains with candidate generating density that did not depend on the current value of the chain:

$$q(\theta, \theta') = q_2(\theta')$$

  - ✓ $q_2(\theta)$ must dominate the target in the tails
  - ✓ It is preferable that the candidate density has the same mode as the target as this leads to a higher proportion of accepted candidates

- For an independent candidate density, the acceptance probability simplifies to be

$$\alpha(\theta, \theta') = \min\left[\frac{p(\theta'|y)q(\theta',\theta)}{p(\theta|y)q(\theta,\theta')}, 1\right] = \min\left[\frac{p(\theta'|y)}{p(\theta|y)} \times \frac{q_2(\theta)}{q_2(\theta')}, 1\right]$$

- $\alpha(\theta, \theta')$ is a product of two ratios that give the chain two opposite tendencies:

  - ✓ The first ratio $p(\theta'|y)/p(\theta|y)$ gives the chain a tendency to move "uphill" with respect to the target (exploitation)

  - ✓ The second ratio $q_2(\theta)/q_2(\theta')$ gives the chain a tendency to move "downhill" with respect to the candidate density (exploration)
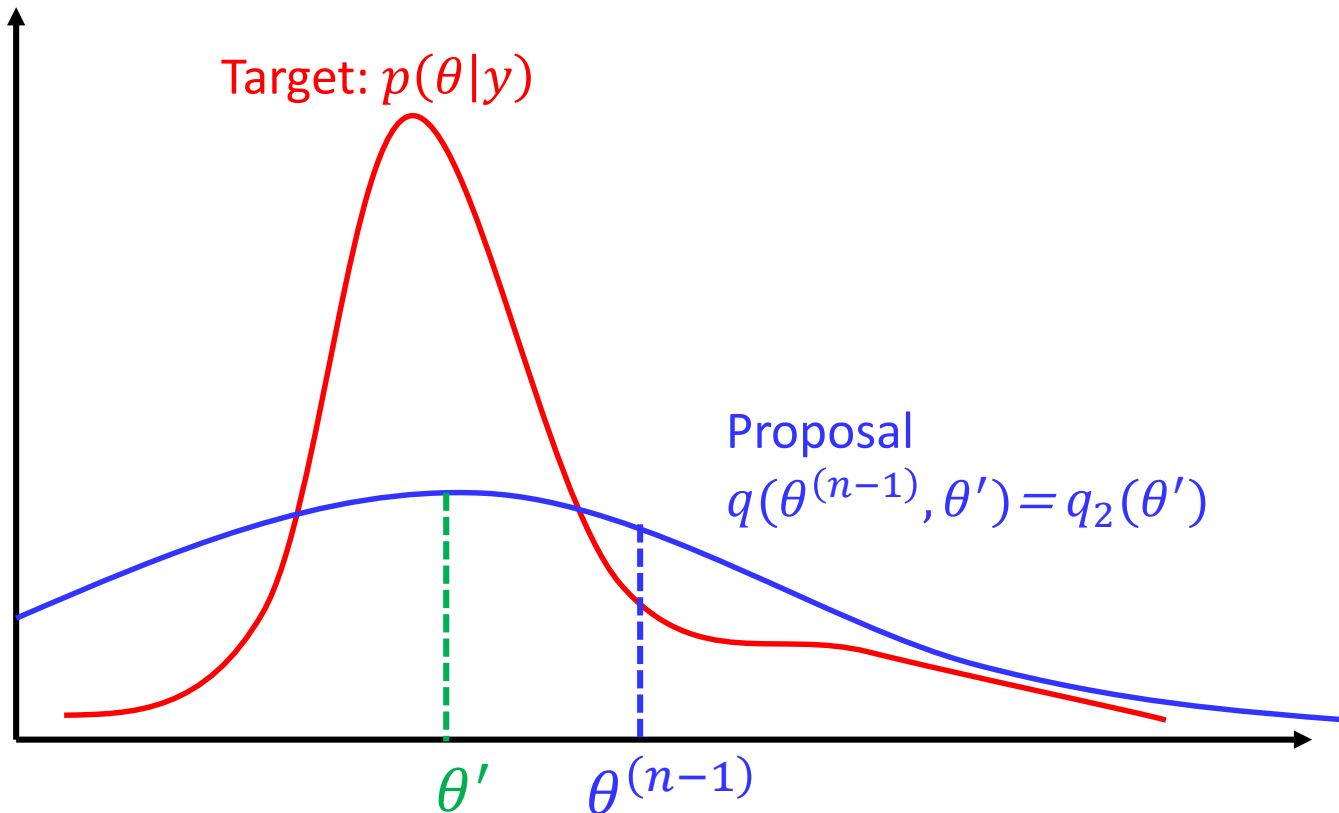
Single parameter with an independent candidate density

Target: $p(\theta|y)$
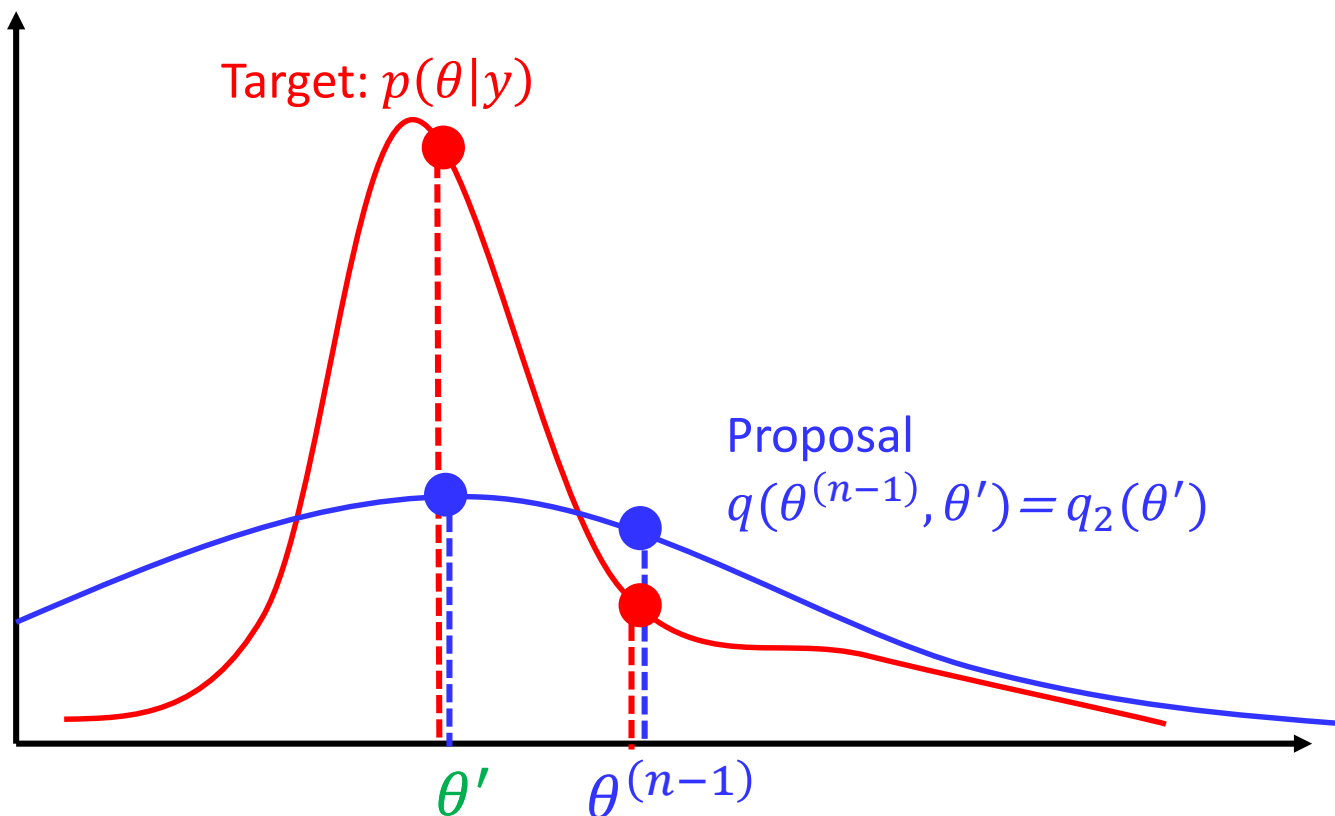
Proposal
$q(\theta^{(n-1)}, \theta') = q_2(\theta')$

Single parameter with an independent candidate density

1. Draw θ' from $q\left(\theta^{(n-1)}, \theta'\right) = q_2(\theta')$

Single parameter with an independent candidate density

Target: $p(\theta|y)$

Proposal
$q(\theta^{(n-1)}, \theta') = q_2(\theta')$

θ' $\theta^{(n-1)}$

2. Calculate the acceptance probability

$$\alpha(\theta^{(n-1)}, \theta') = \min\left[\frac{p(\theta'|y)}{p(\theta^{(n-1)}|y)} \times \frac{q_2(\theta^{(n-1)})}{q_2(\theta')}, 1\right]$$

Single parameter **with an independent candidate density**

Target: $p(\theta|y)$

Proposal
$q(\theta^{(n-1)}, \theta') = q_2(\theta')$

θ' $\qquad$ $\theta^{(n-1)}$

2. Calculate the acceptance probability

$$\alpha(\theta^{(n-1)}, \theta') = \min\left[\frac{-\text{---}\text{---}}{-\text{---}} \times \frac{-\text{---}}{-\text{---}} \quad 1\right]$$

Single parameter with an independent candidate density

Target: $p(\theta|y)$

$\theta^{(n)} \quad \theta^{(n-1)}$
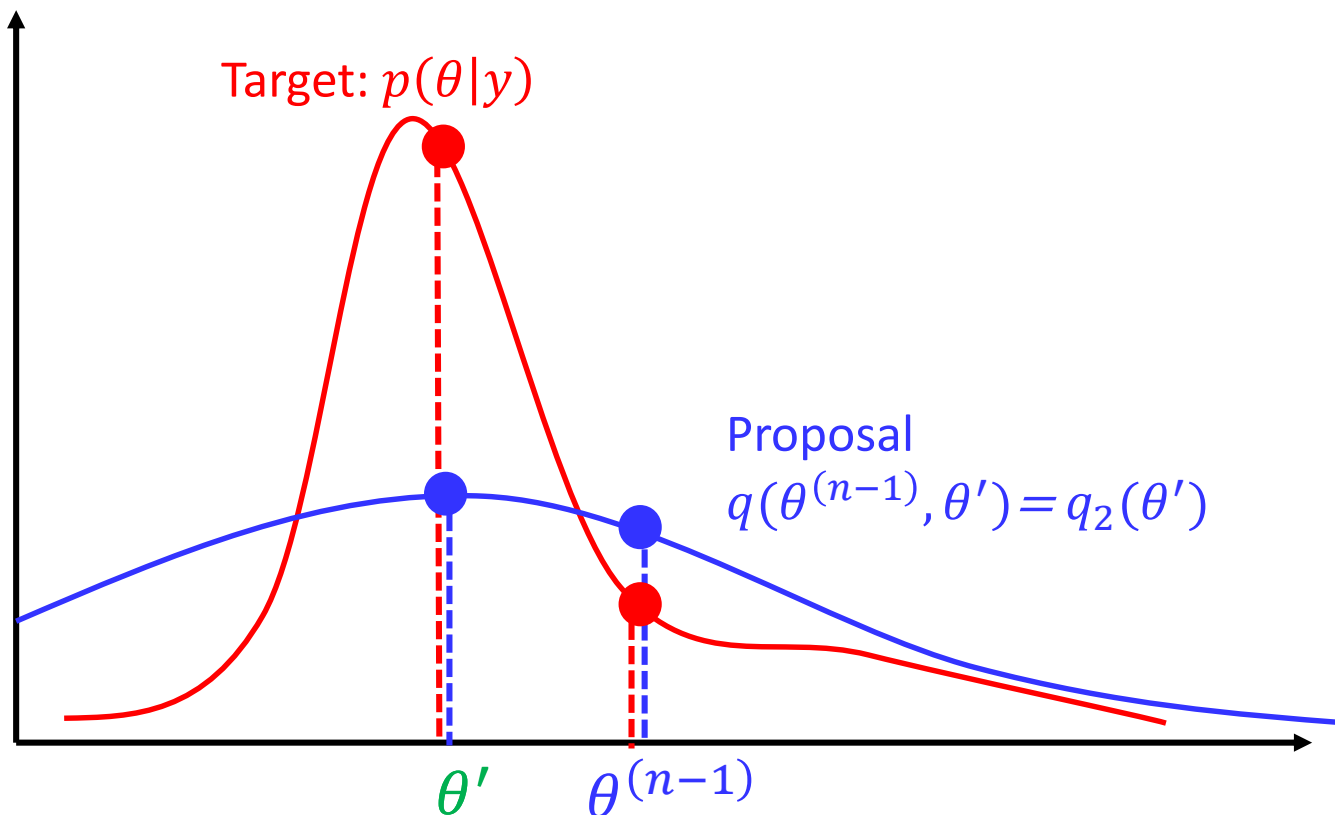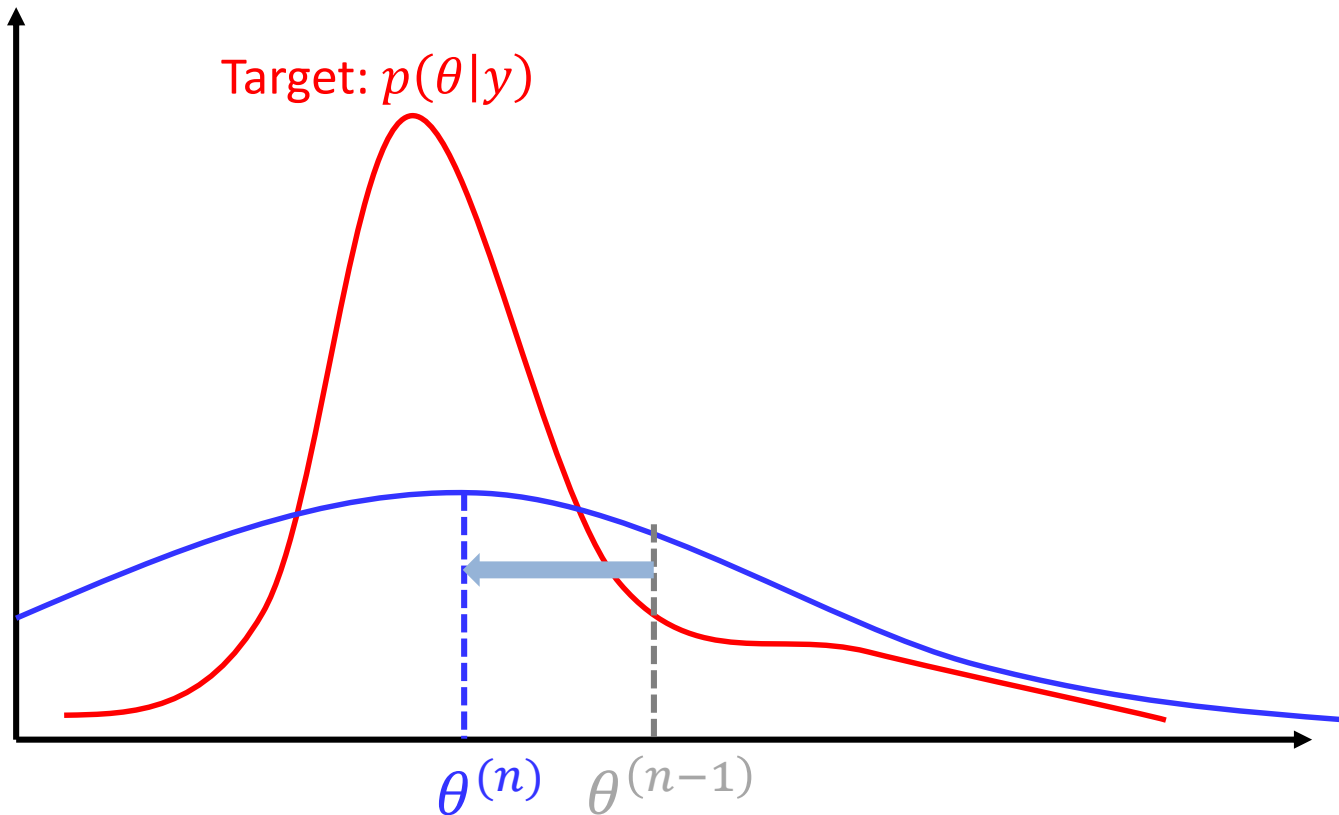
3. Accept θ' based on $\alpha\left(\theta^{(n-1)}, \theta'\right)$

4. Accept $\theta' = \theta^{(n)}$

Single parameter with an independent candidate density : Example

- Big jump (fast and better mixing and less correlated)
- Less accepted
- If candidate density have the same shape covariance structure as the target, then the independent candidate chain will move through the parameter space fairly quickly.
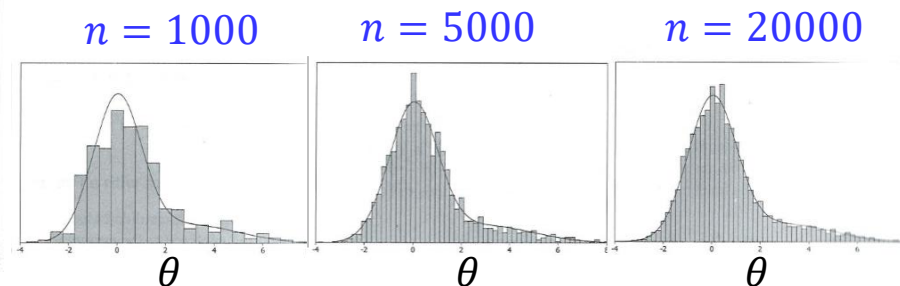
$n = 1000$ $n = 5000$ $n = 20000$

Code demonstration

Multiple parameters

- Suppose we have p parameters $\theta_1, \ldots, \theta_p$. Let the parameter vector be

$$\boldsymbol{\theta} = \left(\theta_1, \ldots, \theta_p \right)$$

- Let $q(\boldsymbol{\theta}, \boldsymbol{\theta}')$ be the candidate density when the chain is at $\boldsymbol{\theta}$ and $p(\boldsymbol{\theta}|y)$ be the posterior density. The detailed balance condition (reversibility condition) will be

$$p(\boldsymbol{\theta}|y)q(\boldsymbol{\theta}, \boldsymbol{\theta}') = p(\boldsymbol{\theta}'|y)q(\boldsymbol{\theta}', \boldsymbol{\theta}) \text{ for all } \boldsymbol{\theta}, \boldsymbol{\theta}'$$

- To satisfy always the reversibility condition, Introduce an acceptance probability

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left[1, \frac{p(\boldsymbol{\theta}'|y)q(\boldsymbol{\theta}', \boldsymbol{\theta})}{p(\boldsymbol{\theta}|y)q(\boldsymbol{\theta}, \boldsymbol{\theta}')} \right]$$

Multiple parameters with a random-walk candidate density

- For a random-walk candidate distribution, the candidate is drawn from a symmetric distribution centered at the current value:

$$q(\boldsymbol{\theta}, \boldsymbol{\theta}') = q_1(\boldsymbol{\theta}' - \boldsymbol{\theta})$$

  ✓ where $q_1(\,,\dots,)$ is symmetric about **0** for each of its arguments.

- Because of symmetry, $q(\boldsymbol{\theta}, \boldsymbol{\theta}') = q_1(\boldsymbol{\theta}' - \boldsymbol{\theta}) = q_1(\boldsymbol{\theta} - \boldsymbol{\theta}') = q(\boldsymbol{\theta}', \boldsymbol{\theta})$, the acceptance probability is simplifies to
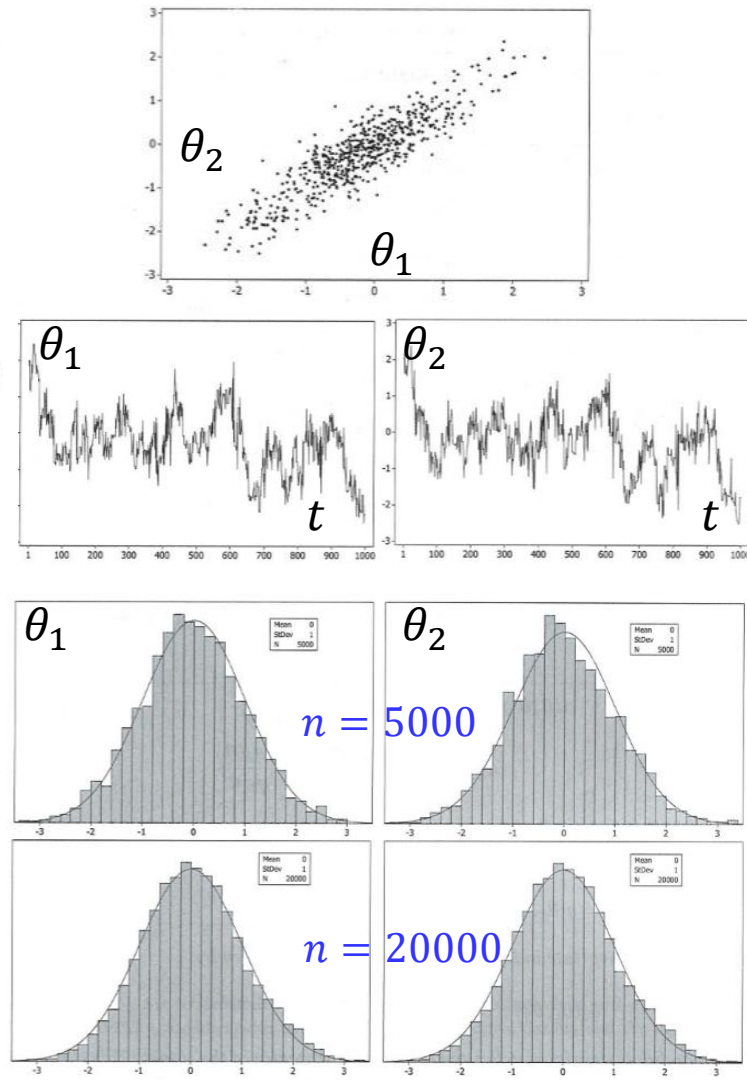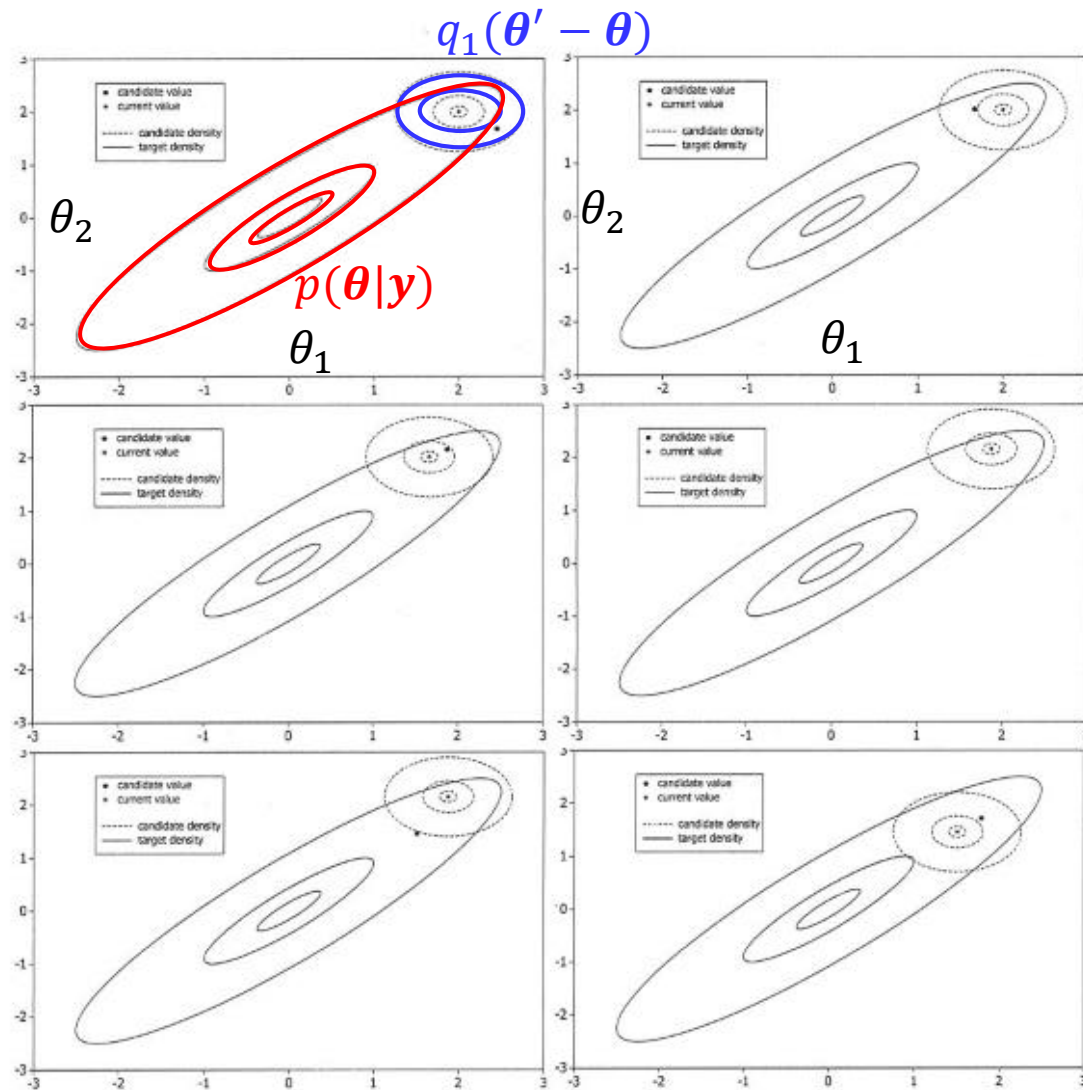
$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left[1, \frac{p(\boldsymbol{\theta}'|y)q(\boldsymbol{\theta}', \boldsymbol{\theta})}{p(\boldsymbol{\theta}|y)q(\boldsymbol{\theta}, \boldsymbol{\theta}')}\right] = \min\left[1, \frac{p(\boldsymbol{\theta}'|y)}{p(\boldsymbol{\theta}|y)}\right]$$

  ✓ If $p(\boldsymbol{\theta}'|y) > p(\boldsymbol{\theta}|y)$, a candidate $\boldsymbol{\theta}'$ will be always accepted
    ➤ The chain will always move "uphill"

  ✓ If $p(\boldsymbol{\theta}'|y) < p(\boldsymbol{\theta}|y)$, a candidate θ' will be accepted with a probability less than 1
    ➤ There is a certain chance that the chain will move "downhill"
    ➤ This allows a chain to move around the whole parameter space over time

  ✓ Will generally have many accepted candidates, but most moves will be a short distance
    ➤ It might take a long time to move around the whole parameter space

Multiple parameters with a random-walk candidate density: Example

$$p(\theta_1, \theta_2|\boldsymbol{y}) = \exp\left\{-\frac{1}{2(1-.9^2)}(\theta_1^2 + 2 \times .9 \times \theta_1\theta_2 + \theta_2^2)\right\} \qquad q(\theta_1' - \theta_1, \theta_2' - \theta_2) = \exp\left\{-\frac{1}{2 \times .4^2}\left((\theta_1' - \theta_1)^2 + (\theta_2' - \theta_2)^2\right)\right\}$$

Multiple parameters with an independent candidate density

- When an independent candidate density is used, the candidate is drawn regardless of the current value

$$q(\boldsymbol{\theta}, \boldsymbol{\theta}') = q_2(\boldsymbol{\theta}')$$
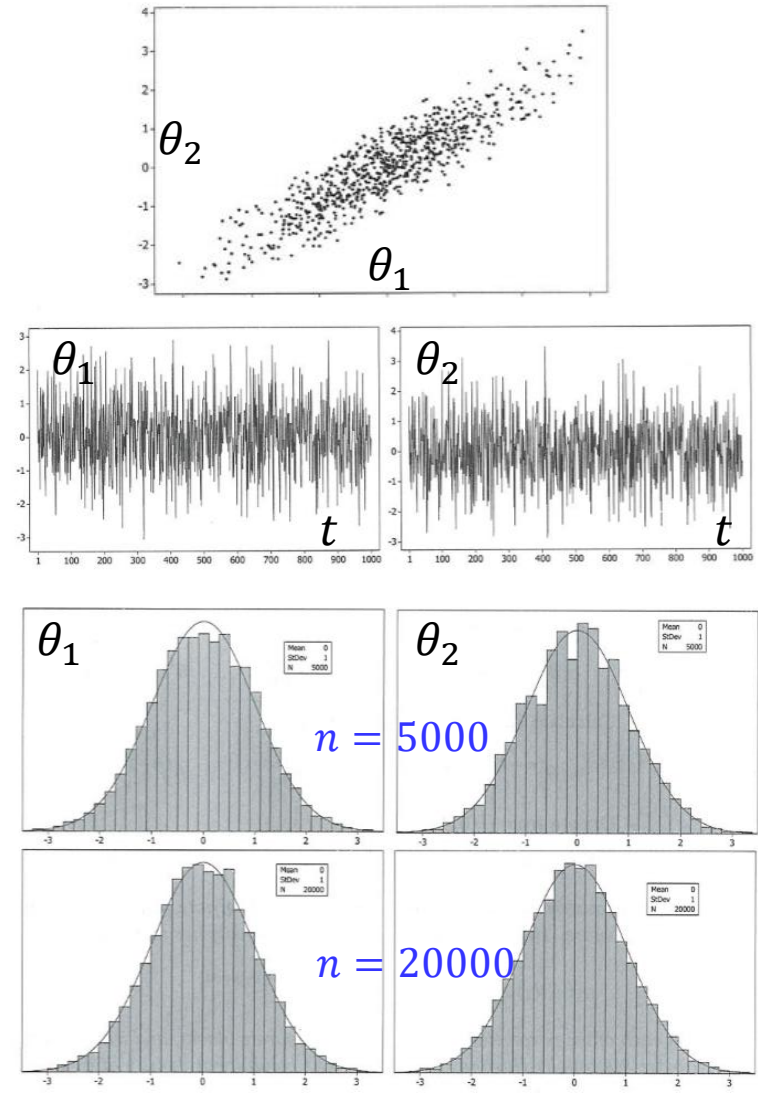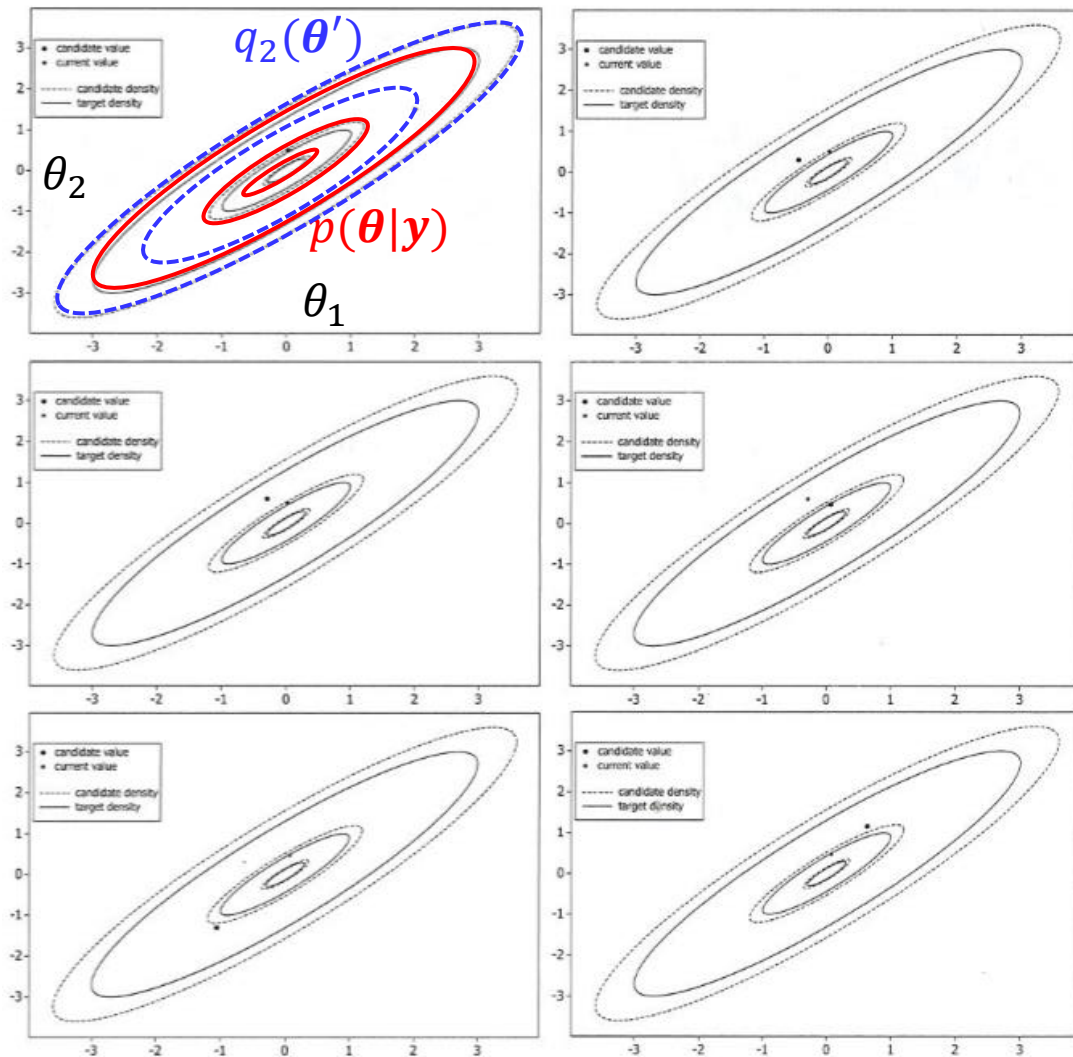
- Then, the acceptance probability is simplifies to

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}') = \min\left[1, \frac{p(\boldsymbol{\theta}'|y)q(\boldsymbol{\theta}', \boldsymbol{\theta})}{p(\boldsymbol{\theta}|y)q(\boldsymbol{\theta}, \boldsymbol{\theta}')}\right] = \min\left[1, \frac{p(\boldsymbol{\theta}'|y)}{p(\boldsymbol{\theta}|y)} \times \frac{q_2(\boldsymbol{\theta})}{q_2(\boldsymbol{\theta}')}\right]$$

✓ $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is a product of two ratios that give the chain two opposite tendencies:

➢ The first ratio $p(\boldsymbol{\theta}'|y)/p(\boldsymbol{\theta}|y)$ gives the chain a tendency to move "uphill" with respect to the target (exploitation)

➢ The second ratio $q_2(\boldsymbol{\theta})/q_2(\boldsymbol{\theta}')$ gives the chain a tendency to move "downhill" with respect to the candidate density (exploration)

Multiple parameters with an independent candidate density: Example

$$p(\theta_1, \theta_2|\boldsymbol{y}) = \exp\left\{-\frac{1}{2(1-.9^2)}(\theta_1^2 + 2 \times .9 \times \theta_1\theta_2 + \theta_2^2)\right\}: \quad q(\theta_1' - \theta_1, \theta_2' - \theta_2) = \exp\left\{-\frac{1}{2 \times 1.2^2}\left((\theta_1' + 2 \times .9 \times \theta_1'\theta_2' + (\theta_2')^2)\right)\right\}$$
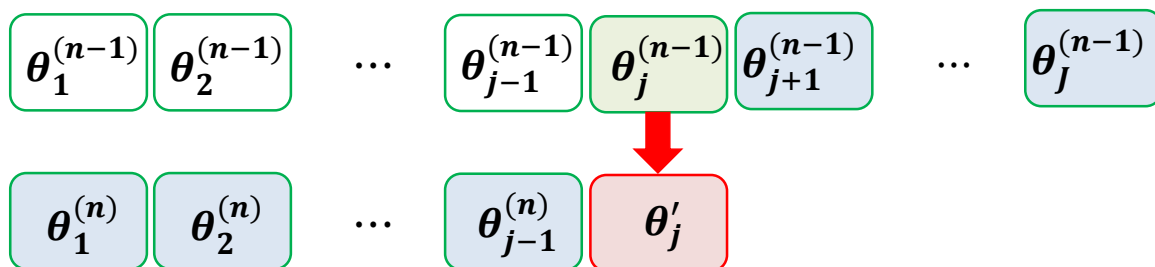
Block-wise Metropolis-Hasting Algorithm

- Let the parameter vector be portioned into blocks

$$\boldsymbol{\theta} = \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_J$$

  ✓ $\boldsymbol{\theta}_j$ is a block of parameters and $\boldsymbol{\theta}_{-j}$ be all the other parameters not in bock

- Instead of applying the Metropolis-Hastings algorithm to the whole parameter vector $\boldsymbol{\theta}$ all at once, the algorithm is applied sequentially to each block of parameters $\boldsymbol{\theta}_j$ in turn conditional on knowing the values of all other parameters

Proposal density: $q\left(\boldsymbol{\theta}_j^{(n-1)}, \boldsymbol{\theta}_j' | \boldsymbol{\theta}_1^{(n)}, \dots, \boldsymbol{\theta}_{j-1}^{(n)}, \boldsymbol{\theta}_{j+1}^{(n-1)}, \dots, \boldsymbol{\theta}_J^{(n-1)}\right)$
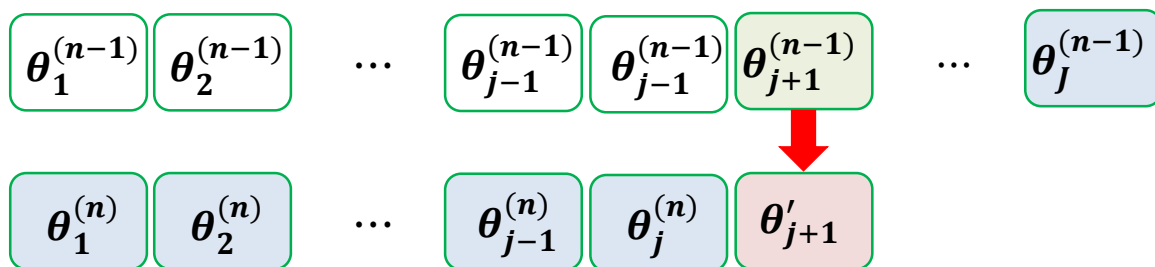
Block-wise Metropolis-Hasting Algorithm

- Let the parameter vector be portioned into blocks

$$\boldsymbol{\theta} = \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_J$$

  ✓ $\boldsymbol{\theta}_j$ is a block of parameters and $\boldsymbol{\theta}_{-j}$ be all the other parameters not in bock

- Instead of applying the Metropolis-Hastings algorithm to the whole parameter vector $\boldsymbol{\theta}$ all at once, the algorithm is applied sequentially to each block of parameters $\boldsymbol{\theta}_j$ in turn conditional on knowing the values of all other parameters

$$\boxed{\boldsymbol{\theta}_1^{(n-1)}}\boxed{\boldsymbol{\theta}_2^{(n-1)}} \quad \cdots \quad \boxed{\boldsymbol{\theta}_{j-1}^{(n-1)}}\boxed{\boldsymbol{\theta}_{j-1}^{(n-1)}}\boxed{\boldsymbol{\theta}_{j+1}^{(n-1)}} \quad \cdots \quad \boxed{\boldsymbol{\theta}_J^{(n-1)}}$$

$$\boxed{\boldsymbol{\theta}_1^{(n)}}\boxed{\boldsymbol{\theta}_2^{(n)}} \quad \cdots \quad \boxed{\boldsymbol{\theta}_{j-1}^{(n)}}\boxed{\boldsymbol{\theta}_j^{(n)}}\boxed{\boldsymbol{\theta}_{j+1}'}$$

Proposal density: $\quad q\left(\boldsymbol{\theta}_{j+1}^{(n-1)}, \boldsymbol{\theta}_{j+1}' \mid \boldsymbol{\theta}_1^{(n)}, \dots, \boldsymbol{\theta}_j^{(n)}, \boldsymbol{\theta}_{j+2}^{(n-1)}, \dots, \boldsymbol{\theta}_J^{(n-1)}\right)$
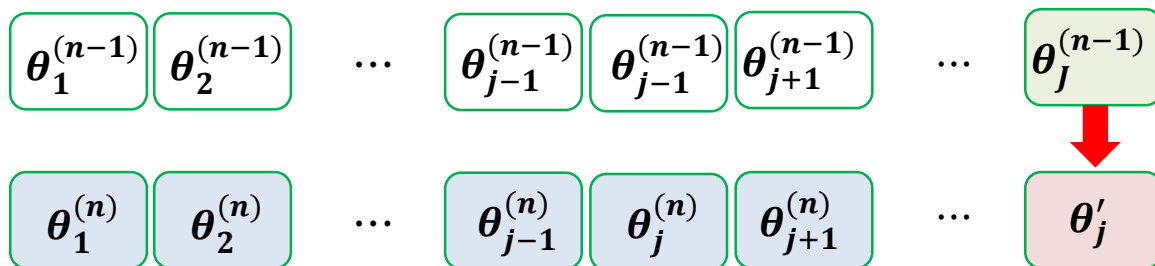
Block-wise Metropolis-Hasting Algorithm

- Let the parameter vector be portioned into blocks

$$\boldsymbol{\theta} = \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_J$$

  ✓ $\boldsymbol{\theta}_j$ is a block of parameters and $\boldsymbol{\theta}_{-j}$ be all the other parameters not in bock

- Instead of applying the Metropolis-Hastings algorithm to the whole parameter vector $\boldsymbol{\theta}$ all at once, the algorithm is applied sequentially to each block of parameters $\boldsymbol{\theta}_j$ in turn conditional on knowing the values of all other parameters

Proposal density: $\quad q\left(\boldsymbol{\theta}_J^{(n-1)}, \boldsymbol{\theta}_J' | \boldsymbol{\theta}_1^{(n)}, \dots, \boldsymbol{\theta}_{J-1}^{(n)}\right)$

Theorem

Let $P_j(\theta_j, A_j | \theta_{-j})$ be the transition kernel for the M-H algorithm applied to parameter bock θ_j holding all the other parameter bock fixed. Then, the transitional kernel

$$P(\theta, A) = \prod_{j=1}^{J} P_j(\theta_j, A_j | \theta_{-j})$$

Has the posterior density $p(\theta | y)$ as its long-run distribution when we cycle through the parameters, holding the rest at their most recent values/

- The practical significance of this product of kernels principle is **that we do not have to run each sub chain to convergence before we move on the next**.

- If we draw a bock of parameters from each sub chain in turn, given the most recent values of the other parameters, this process **will converge to the posterior distribution of the whole parameter vector**.

Block-wise Metropolis-Hasting Algorithm

Algorithm

1) Start at point in parameter space $\boldsymbol{\theta}_1^{(0)}, \dots, \boldsymbol{\theta}_J^{(0)}$

2) for $n = 1, \dots, N$ (iterate over samples)

    (a) for $j = 1, \dots, J$ (iterate over blocks)

        - draw candidate from

$$q\left(\boldsymbol{\theta}_j^{(n-1)}, \boldsymbol{\theta}_j' | \boldsymbol{\theta}_1^{(n)}, \dots, \boldsymbol{\theta}_{j-1}^{(n)}, \boldsymbol{\theta}_{j+1}^{(n-1)}, \boldsymbol{\theta}_J^{(n-1)}\right)$$

        - calculate the acceptance probability

$$a\left(\boldsymbol{\theta}_j^{(n-1)}, \boldsymbol{\theta}_j' | \boldsymbol{\theta}_1^{(n)}, \dots, \boldsymbol{\theta}_{j-1}^{(n)}, \boldsymbol{\theta}_{j+1}^{(n-1)}, \boldsymbol{\theta}_J^{(n-1)}\right)$$

        - draw u from $U(0,1)$

        - If $u < \alpha\left(\boldsymbol{\theta}_j^{(n-1)}, \boldsymbol{\theta}_j'\right)$ then let $\boldsymbol{\theta}_j^{(n)} = \boldsymbol{\theta}_j'$, else let $\boldsymbol{\theta}_j^{(n)} = \boldsymbol{\theta}_j^{(n-1)}$

Gibbs Sampling Algorithm

- We decide to **use the true conditional density as the candidate density** at each step for every block of parameters given the others

$$q\big(\boldsymbol{\theta}_j, \boldsymbol{\theta}_j'|\boldsymbol{\theta}_{-j}\big) = p\big(\boldsymbol{\theta}_j'|\boldsymbol{\theta}_{-j}, \boldsymbol{y}\big)$$

- The acceptance probability becomes:

$$a\left(\boldsymbol{\theta}_j^{(n-1)}, \boldsymbol{\theta}_j'|\boldsymbol{\theta}_1^{(n)}, \ldots, \boldsymbol{\theta}_{j-1}^{(n)}, \boldsymbol{\theta}_{j+1}^{(n-1)}, \boldsymbol{\theta}_J^{(n-1)}\right) = \min\left[1, \frac{p(\boldsymbol{\theta}_j'|\boldsymbol{\theta}_{-j}, y)q(\boldsymbol{\theta}_j', \boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j})}{p(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j}, y)q(\boldsymbol{\theta}_j, \boldsymbol{\theta}_j'|\boldsymbol{\theta}_{-j})}\right]$$

$$= \min\left[1, \frac{p(\boldsymbol{\theta}_j'|\boldsymbol{\theta}_{-j}, y)p(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j}, \boldsymbol{y})}{p(\boldsymbol{\theta}_j|\boldsymbol{\theta}_{-j}, y)p(\boldsymbol{\theta}_j'|\boldsymbol{\theta}_{-j}, \boldsymbol{y})}\right] = 1$$

- ✓ So the **candidate will be always accepted** at each step.

- ✓ Gibbs sampling algorithm is just **a special case of the block wise Metropolis-hasting** algorithm

- ✓ It is well suited for **the class of hierarchical methods**

Algorithm

Initialize $\boldsymbol{\theta}^{(0)}$

for iteration $i = 1, \dots do$

$$\boldsymbol{\theta}_1^{(i)} \sim P\left(\boldsymbol{\theta}_1 \middle| \boldsymbol{\theta}_2^{(i-1)}, \boldsymbol{\theta}_3^{(i-1)}, \dots, \boldsymbol{\theta}_J^{(i-1)}\right)$$

$$\boldsymbol{\theta}_2^{(i)} \sim P\left(\boldsymbol{\theta}_2 \middle| \boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_3^{(i-1)}, \dots, \boldsymbol{\theta}_J^{(i-1)}\right)$$
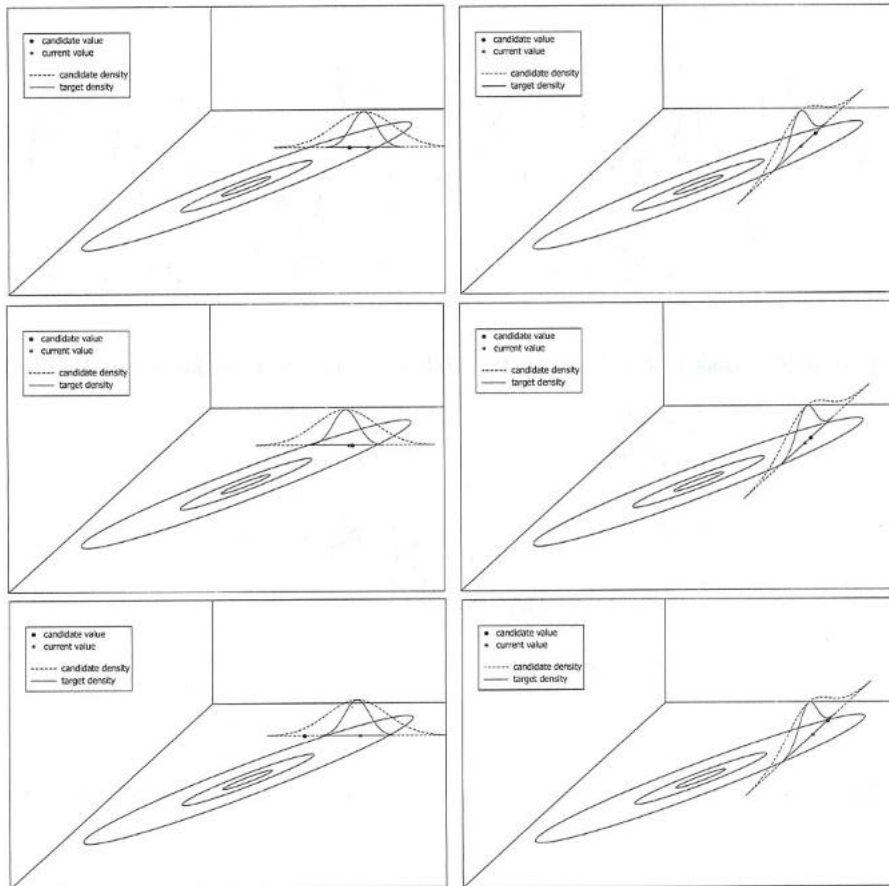
$$\boldsymbol{\theta}_3^{(i)} \sim P\left(\boldsymbol{\theta}_3 \middle| \boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_2^{(i)}, \dots, \boldsymbol{\theta}_J^{(i-1)}\right)$$

$$\vdots$$

$$\boldsymbol{\theta}_J^{(i)} \sim P\left(\boldsymbol{\theta}_J \middle| \boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_2^{(i)}, \dots, \boldsymbol{\theta}_{J-1}^{(i)}\right)$$
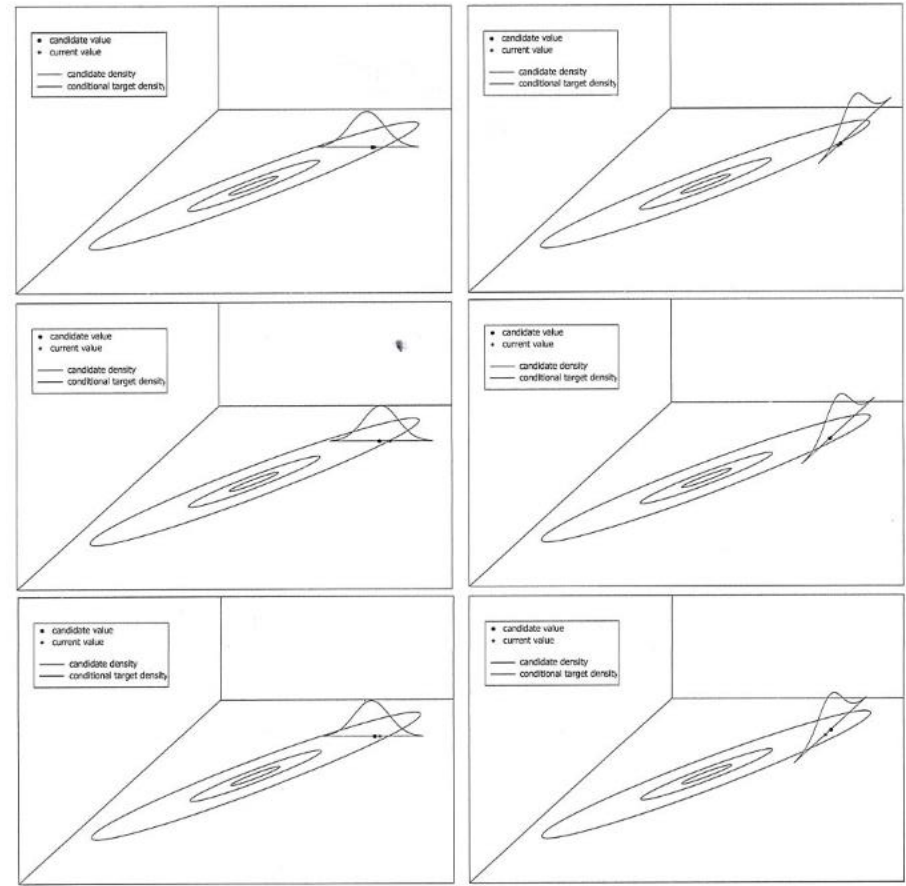
end for

Blok-wise Metropolis-Hastings and Gibbs Sampling Algorithm
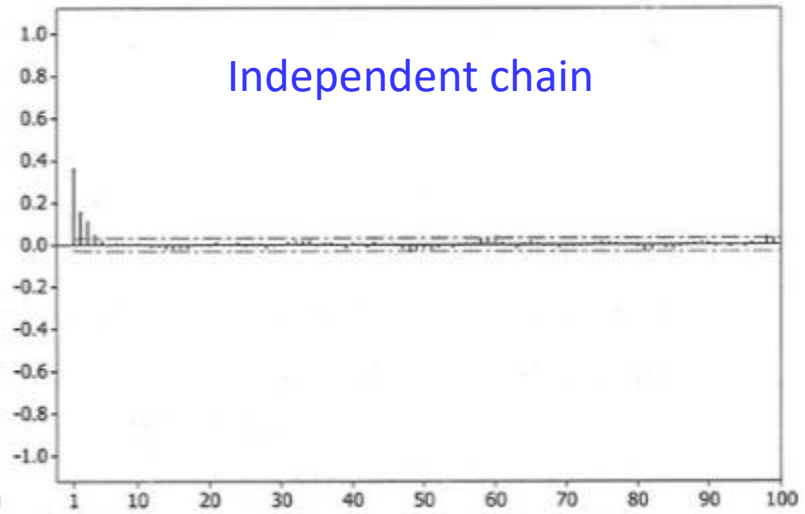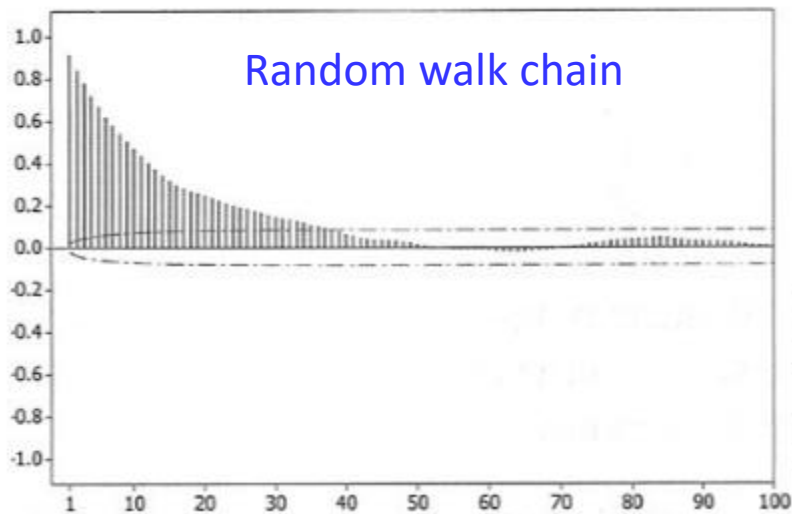
Blok-wise Metropolis-Hastings

Gibbs Sampling

- No exact way to determine how long a burn-in period is required, especially multiple modes(high posterior region) with widely separated.

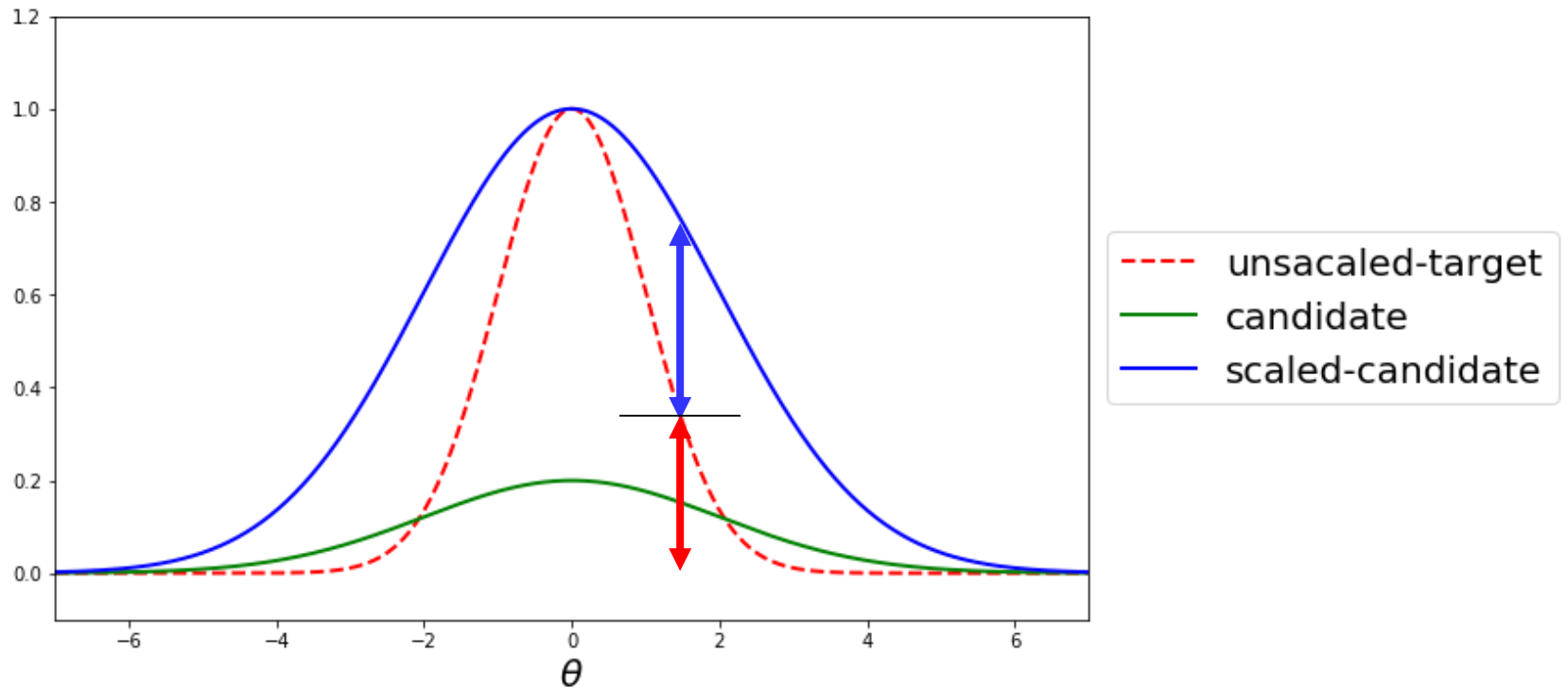- Examining sample autocorrelation function of the MCMC sample. :

- MCMC are not independent random samples.
  - ✓ This is unlike the case for samples drawn directly from the posterior by acceptance-rejection sampling.
  - ✓ This means that is it more difficult to do inferences from the MCMC sample.

- All the draws from the chain after the burn-in time can be considered to be draws from the posterior, although they are not independent from each other.

- Strategies to break out dependency in samples
  - ✓ Use all the samples drawn after burn-in time
    - ➢ Due to the dependency, inferences are not as accurate as those by true posterior
  - ✓ Draw only one sample after every MCMC simulation with burn-in→ repeat n-times
    - ➢ Takes very long time
  - ✓ Thinning approach: draw every n-th sample after burn-in

Working materials

Acceptance rate = ────────────────────