Chapter 5: Markov Chain Monte Carlo (MCMC)

Where we are in the semester

- First half: **Bayesian inference basics**
  - priors, likelihoods, posteriors
  - posterior summaries: mean, variance, credible intervals, prediction

Where we are in the semester

- First half: **Bayesian inference basics**
  - priors, likelihoods, posteriors
  - posterior summaries: mean, variance, credible intervals, prediction
- Second half: **computational aspects become central**
  - posteriors are often complicated
  - we still want to **sample** from them or do **inference**

Where we are in the semester

- First half: **Bayesian inference basics**
  - priors, likelihoods, posteriors
  - posterior summaries: mean, variance, credible intervals, prediction
- Second half: **computational aspects become central**
  - posteriors are often complicated
  - we still want to **sample** from them or do **inference**
- Plan: **Chapter 5 (MCMC)** for about 3 weeks, then Chapter 6.

Big message: after we can *write down* a posterior, we must learn how to *compute with it*.

Sampling so far: two classic tools

In Chapter 4 we saw two sampling methods:

1. **Inverse transform sampling**
   - requires a tractable CDF F and (generalised) inverse F^{-1}
   - then sample $U \sim \mathrm{Unif}(0,1)$ and set $X = F^{-1}(U)$

Sampling so far: two classic tools

In Chapter 4 we saw two sampling methods:

1. **Inverse transform sampling**
   - requires a tractable CDF F and (generalised) inverse F^{-1}
   - then sample $U \sim \mathrm{Unif}(0,1)$ and set $X = F^{-1}(U)$
2. **Rejection sampling**
   - find an easy proposal density g and constant M with $\pi(x) \leq Mg(x)$
   - propose from g, accept/reject using the ratio $\pi/(Mg)$

Sampling so far: two classic tools

In Chapter 4 we saw two sampling methods:

1. **Inverse transform sampling**
   - requires a tractable CDF F and (generalised) inverse F^{-1}
   - then sample $U \sim \mathrm{Unif}(0,1)$ and set $X = F^{-1}(U)$
2. **Rejection sampling**
   - find an easy proposal density g and constant M with $\pi(x) \leq Mg(x)$
   - propose from g, accept/reject using the ratio $\pi/(Mg)$

Common requirement: both need access to the target density *in a form we can evaluate* (and often compare) reliably.

The Bayesian complication: normalising constants

In Bayesian inference we frequently write the posterior only *up to proportionality*:

$$\pi(\theta \mid y) \;\propto\; \underbrace{p(y \mid \theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}}.$$

The Bayesian complication: normalising constants

In Bayesian inference we frequently write the posterior only *up to proportionality*:

$$\pi(\theta \mid y) \;\propto\; \underbrace{p(y \mid \theta)}_{\text{likelihood}}\,\underbrace{p(\theta)}_{\text{prior}}.$$

To be an actual density, we need the normalising constant:

$$\pi(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{C}, \qquad C = \int p(y \mid \theta)p(\theta)\,d\theta.$$

The Bayesian complication: normalising constants

In Bayesian inference we frequently write the posterior only *up to proportionality*:

$$\pi(\theta \mid y) \;\propto\; \underbrace{p(y \mid \theta)}_{\text{likelihood}}\,\underbrace{p(\theta)}_{\text{prior}}.$$

To be an actual density, we need the normalising constant:

$$\pi(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{C}, \qquad C = \int p(y \mid \theta)p(\theta)\,d\theta.$$

- In **1D**, sometimes C is doable (analytically or numerically).
- In **high dimensions** (e.g. $\theta \in \mathbb{R}^{50}$), computing C is often hopeless or extremely costly.

The Bayesian complication: normalising constants

In Bayesian inference we frequently write the posterior only *up to proportionality*:

$$\pi(\theta \mid y) \;\propto\; \underbrace{p(y \mid \theta)}_{\text{likelihood}}\,\underbrace{p(\theta)}_{\text{prior}}.$$

To be an actual density, we need the normalising constant:

$$\pi(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{C}, \qquad C = \int p(y \mid \theta)p(\theta)\,d\theta.$$

- In **1D**, sometimes C is doable (analytically or numerically).
- In **high dimensions** (e.g. $\theta \in \mathbb{R}^{50}$), computing C is often hopeless or extremely costly.

Goal: simulate from $\pi(\theta \mid y)$ while only needing the *unnormalised* density

$$\tilde{\pi}(\theta) = p(y \mid \theta)p(\theta).$$

The MCMC idea in one sentence

Instead of sampling *directly* from the target distribution π:

The MCMC idea in one sentence

Instead of sampling *directly* from the target distribution π:

Key idea

Construct a **stochastic process** $(X_0, X_1, X_2, \dots)$ such that, as time n grows,

the distribution of X_n gets closer and closer to π.

The MCMC idea in one sentence

Instead of sampling *directly* from the target distribution π:

Key idea

Construct a **stochastic process** $(X_0, X_1, X_2, \dots)$ such that, as time n grows,

the distribution of X_n gets closer and closer to π.

- We simulate the process for a long time.
- Once it is "close enough", we treat later states X_n as (approximate) samples from π.

The MCMC idea in one sentence

Instead of sampling *directly* from the target distribution π:

Key idea

Construct a **stochastic process** $(X_0, X_1, X_2, \dots)$ such that, as time n grows,

the distribution of X_n gets closer and closer to π.

- We simulate the process for a long time.
- Once it is "close enough", we treat later states X_n as (approximate) samples from π.

This is the core of **Markov Chain Monte Carlo**.

Why "Markov Chain"?

A Markov chain is a process with **limited memory**:

- the future depends on the **current state**
- not on the entire past history

Why "Markov Chain"?

A Markov chain is a process with **limited memory**:

- the future depends on the **current state**
- not on the entire past history

This is a sweet spot:

- **Independence** (too strong): next state ignores even the present.
- **Full history dependence** (too complex): hard to analyse / simulate / design.
- **Markov dependence**: tractable structure & very flexible.

What is MCMC trying to do?

Goal: produce samples from a target probability distribution π when direct sampling is hard.

Two ingredients that will keep appearing:

1. **Markov property:** the next step depends only on the current state.
2. **Accept/reject mechanism:** propose a move, then accept with a probability chosen so that the chain spends time in states according to π.

Today we start with an example (the King on islands) and then formalise the Markov-chain concepts.

Example 5.1: The King and the ring of islands

There are **10 islands** arranged in a **ring**.

- Island 1 has "size" 1 (smallest), island 2 has size 2, . . . , island 10 has size 10 (largest).
- The King wants to spend **time proportional to island size**.

Example 5.1: The King and the ring of islands

There are **10 islands** arranged in a **ring**.

- Island 1 has "size" 1 (smallest), island 2 has size 2, ..., island 10 has size 10 (largest).
- The King wants to spend **time proportional to island size**.

Constraint (climate conscious travel):

- From any island, he can only travel to a **neighbour** (clockwise or anti-clockwise).
- From island 1, neighbours are islands 2 and 10.

Example 5.1: The King and the ring of islands

There are **10 islands** arranged in a **ring**.

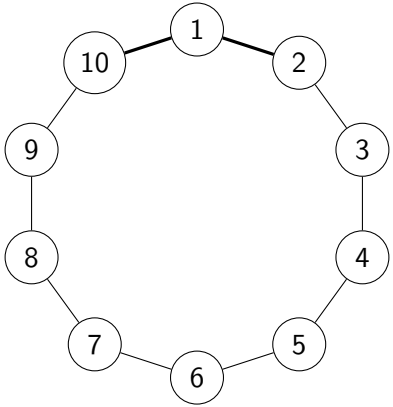
- Island 1 has "size" 1 (smallest), island 2 has size 2, ..., island 10 has size 10 (largest).
- The King wants to spend **time proportional to island size**.

Constraint (climate conscious travel):

- From any island, he can only travel to a **neighbour** (clockwise or anti-clockwise).
- From island 1, neighbours are islands 2 and 10.

Interpretation: The island number is the *state* of a Markov chain. We are designing rules so that the chain visits island k about $k/(1+2+\cdots+10)$ of the time.

Picture: the ring structure



Allowed moves: only to neighbours
(local proposals on the ring).

Motivating example: King on 10 islands

Imagine 10 islands arranged in a ring, labelled $1, 2, \ldots, 10$.

- Island i has "size" proportional to i (so island 10 is largest).
- The King wants a **random** travel rule such that:

$$\mathbb{P}(\text{being on island } i \text{ in the long run}) \propto i.$$

Motivating example: King on 10 islands

Imagine 10 islands arranged in a ring, labelled $1, 2, \ldots, 10$.

- Island i has "size" proportional to i (so island 10 is largest).
- The King wants a **random** travel rule such that:

$$\mathbb{P}(\text{being on island } i \text{ in the long run}) \ \propto \ i.$$

Interpretation: we want a long-run frequency of visits that matches a target distribution

$$\pi(i) = \frac{i}{\sum_{j=1}^{10} j}.$$

A simple stochastic travel rule

Suppose the King is currently on island i.

A simple stochastic travel rule

Suppose the King is currently on island i.

1. **Propose a move** by flipping a fair coin:
   - Heads: propose to go clockwise ($i \to i + 1$)
   - Tails: propose to go anti-clockwise ($i \to i - 1$)

   (wrap around so $1 - 1 = 10$ and $10 + 1 = 1$).

A simple stochastic travel rule

Suppose the King is currently on island i.

1. **Propose a move** by flipping a fair coin:
   - Heads: propose to go clockwise ($i \to i+1$)
   - Tails: propose to go anti-clockwise ($i \to i-1$)

   (wrap around so $1-1=10$ and $10+1=1$).
2. **Accept or reject** the proposal:

$$\text{accept } i \to j \text{ with probability } \alpha(i,j) = \min\left\{1, \frac{j}{i}\right\}.$$

   If rejected, stay at i.

A simple stochastic travel rule

Suppose the King is currently on island i.

1. **Propose a move** by flipping a fair coin:
   - Heads: propose to go clockwise ($i \to i+1$)
   - Tails: propose to go anti-clockwise ($i \to i-1$)

   (wrap around so $1-1=10$ and $10+1=1$).
2. **Accept or reject** the proposal:

$$\text{accept } i \to j \text{ with probability } \alpha(i,j) = \min\Big\{1, \frac{j}{i}\Big\}.$$

   If rejected, stay at i.

Intuition: moves to bigger islands tend to be accepted; moves to smaller islands are sometimes rejected.

Why this is Markov

Let X_n be the island visited in week n.

Why this is Markov

Let X_n be the island visited in week n.

$$\mathbb{P}(X_{n+1} = j \mid X_n = i,\ X_{n-1}, \ldots, X_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i).$$

Why this is Markov

Let X_n be the island visited in week n.

$$\mathbb{P}(X_{n+1} = j \mid X_n = i,\ X_{n-1}, \ldots, X_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i).$$

- The next decision uses only:
  - where you are now ($X_n = i$),
  - a fresh coin flip (proposal),
  - a fresh uniform draw (accept/reject).

Why this is Markov

Let X_n be the island visited in week n.

$$\mathbb{P}(X_{n+1} = j \mid X_n = i,\ X_{n-1}, \ldots, X_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i).$$

- The next decision uses only:
  - where you are now ($X_n = i$),
  - a fresh coin flip (proposal),
  - a fresh uniform draw (accept/reject).
- The whole past route is irrelevant once you know the current island.

Why this is Markov

Let X_n be the island visited in week n.

$$\mathbb{P}(X_{n+1} = j \mid X_n = i,\ X_{n-1}, \ldots, X_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i).$$

- The next decision uses only:
  - where you are now ($X_n = i$),
  - a fresh coin flip (proposal),
  - a fresh uniform draw (accept/reject).
- The whole past route is irrelevant once you know the current island.

This is the **Markov property**.

(Optional) What you would simulate in R

A direct implementation of the island process (10,000 steps)

(Optional) What you would simulate in R

A direct implementation of the island process (10,000 steps)

Later we will explain **why** this converges to $\pi(i) \propto i$ (this is Metropolis–Hastings in disguise).

Definition: Markov chain (discrete time)

A **stochastic process** $(X_n)_{n\geq 0}$ is just a sequence of random variables:

$$X_0, X_1, X_2, \ldots$$

taking values in a **state space** S.

Definition: Markov chain (discrete time)

A **stochastic process** $(X_n)_{n\geq 0}$ is just a sequence of random variables:

$$X_0, X_1, X_2, \ldots$$

taking values in a **state space** S.

It is a **Markov chain** if for every event $A \subseteq \mathsf{S}$,

$$\mathbb{P}(X_{n+1} \in A \mid X_n, X_{n-1}, \ldots, X_0) = \mathbb{P}(X_{n+1} \in A \mid X_n).$$

Definition: Markov chain (discrete time)

A **stochastic process** $(X_n)_{n\geq 0}$ is just a sequence of random variables:

$$X_0, X_1, X_2, \ldots$$

taking values in a **state space** S.

It is a **Markov chain** if for every event $A \subseteq \mathsf{S}$,

$$\mathbb{P}(X_{n+1} \in A \mid X_n, X_{n-1}, \ldots, X_0) = \mathbb{P}(X_{n+1} \in A \mid X_n).$$

In words: the distribution of the next state, given the entire past, depends only on the current state.

State space: what can S be?

In this module, typical examples of state spaces include:

- finite sets: $\{1, 2, \ldots, N\}$ (like the islands)
- the real line: $\mathbb{R}$
- Euclidean space: $\mathbb{R}^d$ (parameters in Bayesian models)
- intervals or subsets of $\mathbb{R}^d$

State space: what can S be?

In this module, typical examples of state spaces include:

- finite sets: $\{1, 2, \ldots, N\}$ (like the islands)
- the real line: $\mathbb{R}$
- Euclidean space: $\mathbb{R}^d$ (parameters in Bayesian models)
- intervals or subsets of $\mathbb{R}^d$

For intuition-building, we start with **finite** S where we can draw pictures and use matrices. Later, we generalise to continuous state spaces (what we need for Bayesian posteriors).

Definition: homogeneous Markov chain

The Markov property alone does *not* force the rules to be the same at each time step. A chain is **homogeneous** (time-homogeneous) if its transition probabilities do not change over time.

Definition: homogeneous Markov chain

The Markov property alone does *not* force the rules to be the same at each time step. A chain is **homogeneous** (time-homogeneous) if its transition probabilities do not change over time.

Formally, for all $n \geq 0$ and all events A,

$$\mathbb{P}(X_{n+1} \in A \mid X_n = x) \quad \text{is the same rule for every } n.$$

Definition: homogeneous Markov chain

The Markov property alone does *not* force the rules to be the same at each time step. A chain is **homogeneous** (time-homogeneous) if its transition probabilities do not change over time.

Formally, for all $n \geq 0$ and all events A,

$$\mathbb{P}(X_{n+1} \in A \mid X_n = x) \quad \text{is the same rule for every } n.$$

In this course: whenever we say "Markov chain" we will mean a **homogeneous** Markov chain.

Finite Markov chains and transition probabilities

Assume S $= \{1, 2, \ldots, N\}$.

Finite Markov chains and transition probabilities

Assume $\mathsf{S} = \{1, 2, \ldots, N\}$.
For a homogeneous Markov chain, define the **one-step transition probabilities**:

$$p_{ij} := \mathbb{P}(X_{n+1} = j \mid X_n = i), \qquad i, j \in \{1, \ldots, N\}.$$

Because of homogeneity, p_{ij} does not depend on n.

Finite Markov chains and transition probabilities

Assume S $= \{1, 2, \ldots, N\}$.
For a homogeneous Markov chain, define the **one-step transition probabilities**:

$$p_{ij} := \mathbb{P}(X_{n+1} = j \mid X_n = i), \qquad i, j \in \{1, \ldots, N\}.$$

Because of homogeneity, p_{ij} does not depend on n.

- Each p_{ij} is a probability, so $0 \leq p_{ij} \leq 1$.
- From state i, you must go somewhere in one step:

$$\sum_{j=1}^{N} p_{ij} = 1 \quad \text{for each fixed } i.$$

Transition matrix

Collect the transition probabilities into the **transition matrix**

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix}.$$

Transition matrix

Collect the transition probabilities into the **transition matrix**

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix}.$$

Two key structural properties:

- all entries satisfy $0 \leq p_{ij} \leq 1$;
- each row sums to 1:

$$P\mathbf{1} = \mathbf{1} \quad (\text{where } \mathbf{1} = (1, \ldots, 1)^\top).$$

Transition matrix

Collect the transition probabilities into the **transition matrix**

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N2} & \cdots & p_{NN} \end{pmatrix}.$$

Two key structural properties:

- all entries satisfy $0 \leq p_{ij} \leq 1$;
- each row sums to 1:

$$P\mathbf{1} = \mathbf{1} \quad (\text{where } \mathbf{1} = (1, \ldots, 1)^\top).$$

So P is a **row-stochastic** matrix.

State-space diagram (graph picture)

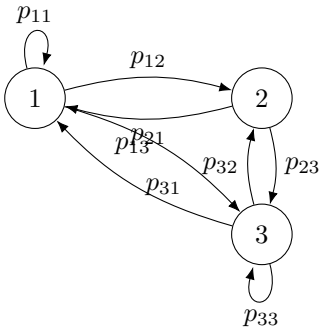
For a finite Markov chain we often visualise the dynamics with a directed graph:

- nodes = states
- arrows $i \to j$ labelled with probability p_{ij} (only draw if $p_{ij} > 0$)

State-space diagram (graph picture)

For a finite Markov chain we often visualise the dynamics with a directed graph:

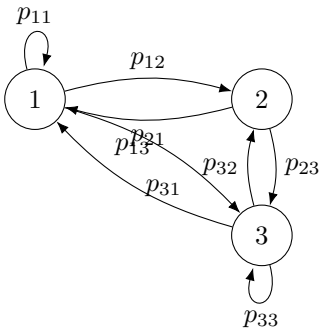
- nodes = states
- arrows $i \to j$ labelled with probability p_{ij} (only draw if $p_{ij} > 0$)



State-space diagram (graph picture)

For a finite Markov chain we often visualise the dynamics with a directed graph:

- nodes = states
- arrows $i \to j$ labelled with probability p_{ij} (only draw if $p_{ij} > 0$)



This picture often makes it easier to *reason* about how the chain moves.

Example: a 3-state chain

Consider the transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

Example: a 3-state chain

Consider the transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

Interpretation row-by-row:

- From state 1: stay at 1 with prob $1/2$, go to 2 with prob $1/4$, go to 3 with prob $1/4$.

Example: a 3-state chain

Consider the transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

Interpretation row-by-row:

- From state 1: stay at 1 with prob $1/2$, go to 2 with prob $1/4$, go to 3 with prob $1/4$.
- From state 2: you never stay; you go to 1 or 3 with prob $1/2$ each.

Example: a 3-state chain

Consider the transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

Interpretation row-by-row:

- From state 1: stay at 1 with prob $1/2$, go to 2 with prob $1/4$, go to 3 with prob $1/4$.
- From state 2: you never stay; you go to 1 or 3 with prob $1/2$ each.
- From state 3: you never go to 1; you go to 2 with prob $1/3$ or stay at 3 with prob $2/3$.

Example: a 3-state chain

Consider the transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & \frac{2}{3} \end{pmatrix}.$$

Interpretation row-by-row:

- From state 1: stay at 1 with prob $1/2$, go to 2 with prob $1/4$, go to 3 with prob $1/4$.
- From state 2: you never stay; you go to 1 or 3 with prob $1/2$ each.
- From state 3: you never go to 1; you go to 2 with prob $1/3$ or stay at 3 with prob $2/3$.

Next we will use these objects (transition probabilities, matrices, diagrams) to talk about **long-run behaviour**.

How this connects to Bayesian computation

In MCMC we want to sample from a target distribution π (often a posterior).

How this connects to Bayesian computation

In MCMC we want to sample from a target distribution π (often a posterior).

- We design a Markov chain with transition rule P (or kernel, in continuous spaces).

How this connects to Bayesian computation

In MCMC we want to sample from a target distribution π (often a posterior).

- We design a Markov chain with transition rule P (or kernel, in continuous spaces).
- The chain is constructed so that π is the **long-run / stationary** distribution.

How this connects to Bayesian computation

In MCMC we want to sample from a target distribution π (often a posterior).

- We design a Markov chain with transition rule P (or kernel, in continuous spaces).
- The chain is constructed so that π is the **long-run / stationary** distribution.
- Then, after running long enough, the chain spends time in regions according to π.

How this connects to Bayesian computation

In MCMC we want to sample from a target distribution π (often a posterior).

- We design a Markov chain with transition rule P (or kernel, in continuous spaces).
- The chain is constructed so that π is the **long-run / stationary** distribution.
- Then, after running long enough, the chain spends time in regions according to π.

Why this is powerful

Many MCMC algorithms only require $\tilde{\pi}(\theta)$, the **unnormalised** posterior, and never compute the normalising constant C.

What comes next

To make MCMC work, we need to answer:

- What does it mean for a chain to have a **long-run distribution**?
- When does a chain **converge** to that distribution?
- How do we **design** transition rules so that the target π is stationary?

What comes next

To make MCMC work, we need to answer:

- What does it mean for a chain to have a **long-run distribution**?
- When does a chain **converge** to that distribution?
- How do we **design** transition rules so that the target π is stationary?

We will start with finite Markov chains (intuition + matrix tools), then move to the continuous setting used in Bayesian inference.

What comes next

To make MCMC work, we need to answer:

- What does it mean for a chain to have a **long-run distribution**?
- When does a chain **converge** to that distribution?
- How do we **design** transition rules so that the target π is stationary?

We will start with finite Markov chains (intuition + matrix tools), then move to the continuous setting used in Bayesian inference.

And we will return to the islands example as our first concrete instance of a **Metropolis–Hastings** chain.