

# Chapter 2 - Simulating Statistical Models

## Markov Chains.

---

Prof. Alex Alvarez, Ali Raisolsadat

School of Mathematical and Computational Sciences  
University of Prince Edward Island

## Definition

A **stochastic process** is a collection of random variables

$$X = \{X_t\}_{t \in T}, \quad X_t \in S$$

where

- $T$  = **index set** (often time:  $\mathbb{N}$  for discrete,  $\mathbb{R}^+$  for continuous)
- $S$  = **state space** (possible values of  $X_t$ )

## Examples

- Tossing a coin repeatedly:  $X_t \in \{0, 1\}$
- Daily stock price:  $X_t \in \mathbb{R}^+$
- Temperature over time:  $X_t \in \mathbb{R}$

## Key Idea

The random variables  $X_t$  are usually **dependent**, and interesting processes have *structured dependence*.

## Classical Assumption

Many results in probability theory assume that random variables are **independent**, which makes computations much easier.

## But in Reality...

- Independence is often **not realistic** (e.g., stock prices, weather, genetics).
- Random variables are usually **dependent**.

## The Challenge

Therefore, adding dependence between random variables makes computations more difficult.

**Goal:** Find dependence structures that are both **realistic and computationally workable**.

## BEHOLD: The most awesome Markov Chains

They provide a **simple yet powerful model of dependence**: the future depends only on the present, not on the entire past.

# Dependence in a Markov Chain

## Intuition

A Markov chain has **no memory**. That is to say that the **most recent state** only matters for predicting the next one.

## Definition 1 (Discrete-Time Markov Chain)

A stochastic process

$$X = \{X_0, X_1, X_2, \dots\}, \quad X_n \in S$$

satisfies the **Markov property** if for all  $n \geq 0$ ,

$$P(X_{n+1} \in A_j \mid X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) = P(X_{n+1} \in A_j \mid X_n = i_n).$$

## Notes

- The distribution of  $X_0$  is called the **initial distribution**.
- Sometimes  $X_0$  is known (and not random).
- The condition above is called the **Markov property**, named after Russian mathematician Andrey Markov.

# Simple Random Walk (definition)

## Simple random walk

Let  $\{Y_k\}_{k \geq 1}$  be i.i.d. random variables (“steps”) with

$$\mathbb{P}(Y_k = 1) = p, \quad \mathbb{P}(Y_k = -1) = 1 - p$$

Fix an initial state  $X_0 \in \mathbb{Z}$ . Define the *random walk*

$$X_n = X_0 + \sum_{k=1}^n Y_k, \quad n \geq 1$$

Thus each step moves the walker by  $+1$  or  $-1$ .

### Intuition

The next position  $X_{n+1}$  is obtained by adding one fresh independent step  $Y_{n+1}$  to the current position  $X_n$ .

### Proposition 1

The random walk  $\{X_n\}$  defined above satisfies the Markov property: for all  $n \geq 0$  and integers  $i_0, \dots, i_n, j$ ,

$$\mathbb{P}(X_{n+1} = j \mid X_0 = i_0, \dots, X_n = i_n) = \mathbb{P}(X_{n+1} = j \mid X_n = i_n)$$

**Proof.** Start from the left-hand conditional probability and use the construction  $X_{n+1} = X_n + Y_{n+1}$ :

$$\begin{aligned}\mathbb{P}(X_{n+1} = j \mid X_0 = i_0, \dots, X_n = i_n) &= \mathbb{P}(i_n + Y_{n+1} = j \mid X_0 = i_0, \dots, X_n = i_n) \\ &= \mathbb{P}(Y_{n+1} = j - i_n \mid X_0 = i_0, \dots, X_n = i_n).\end{aligned}$$

Because  $Y_{n+1}$  is independent of the past (the  $\{Y_k\}$  are i.i.d.), the conditioning on  $X_0, \dots, X_n$  is irrelevant, so

$$\mathbb{P}(Y_{n+1} = j - i_n \mid X_0, \dots, X_n) = \mathbb{P}(Y_{n+1} = j - i_n).$$

## Why the Markov Property Holds (Random Walk Example)

The event  $\{X_{n+1} = j\}$  is the same as

$$\{i_n + Y_{n+1} = j\} \iff \{Y_{n+1} = j - i_n\}$$

Thus,

$$\mathbb{P}(X_{n+1} = j \mid X_n = i_n) = \mathbb{P}(Y_{n+1} = j - i_n)$$

Since  $Y_{n+1}$  is independent of the past, the history  $(X_0, \dots, X_{n-1})$  does not matter. Hence the Markov property holds all  $n \geq 0$  and integers  $i_0, \dots, i_n, j$ .

## Why the Markov Property Holds – Example

**A Gambler's Fortune.** Let  $X_n$  be the amount of money a gambler has after  $n$  bets. At each step:

$$X_{n+1} = X_n + Y_{n+1}, \quad Y_{n+1} = \begin{cases} \text{heads} & (\text{win } \$1) \\ \text{tails} & (\text{lose } \$1) \end{cases}$$

If the gambler currently has \$10, then the next outcome depends only on the result of the next bet ( $Y_{n+1}$ ), not on how they got to \$10.

But we can still see that for an unbiased coin ( $p = \frac{1}{2}$ )

$$P(X_n > 10) = \sum_{k=\lfloor n/2 \rfloor + 1}^n \binom{n}{k} \left(\frac{1}{2}\right)^n; \quad P(X_n > 10) \approx \frac{1}{2} \quad \text{as } n \rightarrow \infty; \text{ by CLT}$$

### Takeaway

The process is Markov because the *future fortune depends only on the present fortune and the next bet*, not the entire betting history.



**Interpretation:** The future is independent of the past **given the present**.

- Natural in many applications:
  - Total gains/losses in gambling
  - Cumulative processes such as queues
- The Markov property makes probability calculations much easier:
  - Compute probabilities and expectations step by step
  - No need to assume all random variables are independent

Let us consider first the case of finite/discrete state space  $S$ .

Conditional probabilities of the type  $P(X_{n+1} = j | X_n = i)$  are called one-step transition probabilities **from state  $i$  to state  $j$** .

If these one-step transition probabilities do not depend on time variable  $n$ , then we say that the Markov Chain is **homogeneous** and we write:

$$P(X_{n+1} = j | X_n = i) = p_{ij}$$

## Calculating probabilities

If the Markov chain is homogeneous, we have:

$$\begin{aligned} & P(X_0 = i_0, X_1 = i_1, \dots, X_{n+1} = i_{n+1}) \\ = & P(X_0 = i_0) \cdot \prod_{k=0}^n P(X_{k+1} = i_{k+1} | X_k = i_k) \\ = & P(X_0 = i_0) \cdot \prod_{k=0}^n p_{i_k i_{k+1}} \end{aligned}$$

The probabilities of the whole trajectories of the Markov Chain is determined by the transition probabilities and the initial distribution of  $X_0$ .

## One-step transition probability matrix

Markov Chain with state space  $S = \{1, 2, \dots, M\}$

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MM} \end{pmatrix}$$

Properties:

- Rows add up to 1
- All entries are non-negative

## Example: Weather

Define three possible states for the weather:

1) Sunny                  2) Rainy                  3) Snowy

Assume that the daily weather can be described by a homogeneous Markov Chain with transition probability matrix:

$$P = \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}$$

**Definition:** The probability of moving from state  $i$  to  $j$  in two steps is

$$\mathbb{P}(X_{n+2} = j \mid X_n = i).$$

**Derivation:**

$$\begin{aligned}\mathbb{P}(X_{n+2} = j \mid X_n = i) &= \sum_k \mathbb{P}(X_{n+2} = j \mid X_{n+1} = k, X_n = i) \mathbb{P}(X_{n+1} = k \mid X_n = i) \\ &= \sum_k \mathbb{P}(X_{n+2} = j \mid X_{n+1} = k) p_{ik} \\ &= \sum_k p_{kj} p_{ik}\end{aligned}$$

This is entry  $(i, j)$  of matrix  $P^2$ . The two-step transition probability is  $P^2$ .

### Theorem: m-step Transition Probability Matrix

If  $P$  is the one-step transition probability matrix of a homogeneous Markov chain, then the  $m$ -step transition probability matrix is  $P^m$ .

## Example: Wheather (cont.)

If today (Monday) is sunny, what is the probability that Thursday will be sunny again?

Thursday is three days from today so we compute the 3-step transition probability matrix  $P^3$ .

$$P^3 = \begin{pmatrix} 0.3220 & 0.4680 & 0.2100 \\ 0.2100 & 0.5320 & 0.2580 \\ 0.2580 & 0.4680 & 0.2740 \end{pmatrix}$$

**Answer:** 0.3220, which is the entry (1, 1) of matrix  $P^3$ .

Example: Wheather (cont.)

Limiting behaviour (rounded to 4 decimal places).

$$P^{10} = \begin{pmatrix} 0.2502 & 0.4999 & 0.2498 \\ 0.2498 & 0.5001 & 0.2501 \\ 0.2501 & 0.4999 & 0.2499 \end{pmatrix}$$

$$P^{20} = \begin{pmatrix} 0.2500 & 0.5000 & 0.2500 \\ 0.2500 & 0.5000 & 0.2500 \\ 0.2500 & 0.5000 & 0.2500 \end{pmatrix}$$

**Observation:** The effect of the initial state becomes negligible.



## Distribution of the Chain

Suppose that  $s \in \mathbb{R}^M$  (row vector) represents the distribution (probability mass function) of the Markov chain at time  $n$ .

The distribution of the chain at time  $n + 1$  will be  $sP$

In general the distribution of the chain at time  $n + m$  will be  $sP^m$

## Stationary Distribution

**Definition:** A distribution vector  $s \in \mathbb{R}^M$  is said to be stationary for a Markov chain with one-step probability distribution  $P$  if  $s = sP$ .

**Interpretation:** If the distribution (p.m.f.) of  $X_0$  is  $s$ , then the distribution of  $X_n$  is  $s$  for all  $n$ .

Example: Weather (cont.)

$$\begin{pmatrix} 0.25 & 0.5 & 0.25 \end{pmatrix} \begin{pmatrix} 0.6 & 0.3 & 0.1 \\ 0.1 & 0.7 & 0.2 \\ 0.2 & 0.3 & 0.5 \end{pmatrix} = \begin{pmatrix} 0.25 & 0.5 & 0.25 \end{pmatrix}$$

This means that  $s = \begin{pmatrix} 0.25 & 0.5 & 0.25 \end{pmatrix}$  is a stationary distribution for that Markov chain.

In general there may be more than one stationary distribution for a Markov Chain, but in most “interesting” cases there is only one stationary distribution.

Under certain conditions, it can be proven a connection between the stationary distribution and the limiting behaviour of a Markov chain.

## Generation of Markov Chains

The following algorithm can be used to generate paths of a Markov Chain with arbitrary length  $n$  (meaning that we have to generate the sequence  $X_0, X_1, \dots, X_n$ )

---

### Algorithm 1 Generate a Markov Chain trajectory

---

- 1: Generate  $X_0$  according to the initial distribution.
- 2: **for**  $k = 1$  to  $n$  **do**
- 3:     Generate  $X_k \in S$  with probability

$$P(X_k = j) = p_{X_{k-1}, j}$$

- 4: **end for**
  - 5: **return**  $(X_0, X_1, \dots, X_n)$
-

## Example: Simulating a Markov Chain

Consider a Markov Chain taking values in  $S = \{1, 2\}$  with initial distribution

$$P(X_0 = 1) = 0.3, \quad P(X_0 = 2) = 0.7,$$

and transition matrix

$$P = \begin{bmatrix} 0.5 & 0.5 \\ 0.6 & 0.4 \end{bmatrix}$$

## Example: Simulating a Markov Chain - R and Python

Generate a path of this Markov Chain with length  $n = 40$ .

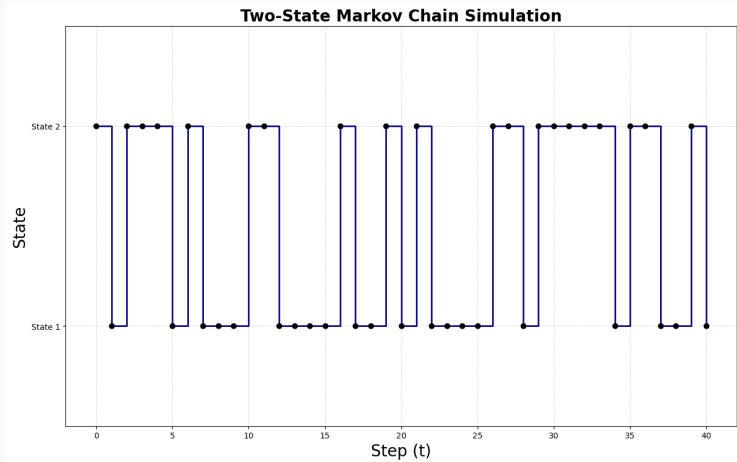
### R Code

```
1  n <- 40
2  P <- matrix(c(0.5,0.6,0.5,0.4),2,2)
3  X <- vector()
4  U <- runif(1)
5  U1 <- runif(n)
6  if (U < 0.3) {X[1] <- 1}
7  else {X[1] <- 2}
8
9  for (i in c(1:n)) {
10     if (U1[i] < P[X[i],1]) {
11         X[i+1] <- 1
12     } else {
13         X[i+1] <- 2
14     }
15 }
16 A <- c(0:n)
17 plot(A, X, type="s")
```

### Python Code

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  n = 40
5  P = np.array([[0.5, 0.5],
6                [0.6, 0.4]])
7  X = np.zeros(n+1, dtype=int)
8  U = np.random.rand()
9  if U < 0.3: X[0] = 1
10 else: X[0] = 2
11
12 U1 = np.random.rand(n)
13
14 for i in range(n):
15     if U1[i] < P[X[i]-1, 0]:
16         X[i+1] = 1
17     else:
18         X[i+1] = 2
19
20 plt.step(range(n+1), X)
21 plt.show()
```

## Example: Markov Chain Path



Simulated trajectory of a 2-state Markov Chain with  $n = 40$ .

## Remarks:

- The code shown before is not the most compact/efficient for the generation of Markov Chain paths.
- A more efficient approach is to use **categorical sampling**:
  - In **R**: use `sample()` or `rmultinom()`.
  - In **Python**: use `numpy.random.choice()` (or `torch.multinomial()` for GPU).
- **Complexity**: generating a single path of length  $n$  requires  $O(n)$  categorical draws. The vectorized implementations are much faster when simulating many independent paths.
- Efficient Markov Chain simulation is especially important in **MCMC** (Markov Chain Monte Carlo), where one often needs to generate millions of steps or multiple chains.



Consider a Markov Chain taking values in  $S = \{1, 2, 3\}$  with initial distribution  $X_0 = 2$  and

$$P = \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \\ 0.5 & 0.1 & 0.4 \end{bmatrix}$$

Generate a path of this Markov Chain with length  $n = 40$ .