Introduction to Markov Chains

Corey Zhang
University of Delaware

Dec, 2023

Abstract

Markov chains are a simple and powerful way to model systems that change randomly over time, like customers in a store or calls on a phone line. This article is covering how they work, what happens in the long run, and how they apply to problems. We will use easy examples, like a supermarket queue, and keep all the math clear to show exactly how things are calculated.

1 What Are Markov Chains?

Imagine that you are watching a supermarket checkout line. The number of people waiting changes as customers arrive or are served. A Markov chain is a way to model this kind of system, where the future depends only on what is happening now, not what happened before. It's like saying, "If there are 3 people in line now, what is the chance that there will be 2, 3 or 4 next moment?" The past doesn't matter, just the current number.

We focus on systems with a fixed number of states (like 0 to 10 customers) that change at specific discrete times. The chance of moving from state i to state j, called the transition probability P_{ij} is written as:

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

X_n is the state at time n. This is called the Markov property, means that the next step only cares about where you are now.

In our supermarket, suppose that customers arrive with chance p and are served with chance q. If there are i customers:

- **Arrival** (go to $i+1$): $p(1-q)$
- **Departure** (go to $i-1$): $(1-p)q$
- **Stay put**: $bq + (1-p)(1-q)$

At the edges (0 customers, no departures; 10, no arrivals), also notes that the chance of two customers arrive at one moment equal to zero. To find the chance of going from 1 to 2 to 3 customers, multiply the probabilities: $P_{12} \cdot P_{23}$.

To figure out the chance of being in state j after n steps from state i, called $R_{ij}(n) = P(X_n = j | X_0 = i)$, we use a formula that adds up all possible paths:

$$R_{ij}(n) = \sum_k R_{ik}(n-1) P_{kj}$$

This breaks the problem into smaller steps.

Consider a simple two-state system (say, "busy" and "not busy") with:

$$P_{11} = 0.5, \quad P_{12} = 0.5, \quad P_{21} = 0.2, \quad P_{22} = 0.8$$

After many steps, the chances start converge: $R_{11}(n) \approx 2/7$, $R_{12}(n) \approx 5/7$. This hints that the system might reach a steady state, where probabilities don't change anymore.

2 What Happens in the Long Run?

Over time, many Markov chains converges which means each state stays constant. These are called **steady-state probabilities**, π_j, and they tell us how often we expect to find the system in state j. For a well-behaved chain (one where all states connect and there's no repeating pattern), $R_{ij}(n)$ approaches π_j, no matter where you start. These probabilities satisfy the **balance equations**:

$$\pi_j = \sum_k \pi_k P_{kj}$$

plus the rule that probabilities add to 1:

$$\sum_j \pi_j = 1$$

Think of π_j as how often you'd find the system in state j if you checked it after a long time. For the two-state example, we solve:

$$\pi_1 = 0.5\pi_1 + 0.2\pi_2, \quad \pi_2 = 0.5\pi_1 + 0.8\pi_2, \quad \pi_1 + \pi_2 = 1$$

This gives $\pi_1 = 2/7$, $\pi_2 = 5/7$, meaning the system is in state 1 about 28.6% of the time and state 2 about 71.4%. and they are independent of the initial state in this example.

2.1 Types of States

States can be **recurrent** or **transient**. In the long run, the system leaves transient states and settles into a recurrent group where all states connect. If there's only one such group, the starting point doesn't matter in the end. But if there are multiple groups, where you begin decides which group you end up in.

2.2 Avoiding Repeating Patterns

A chain has a **repeating pattern** (periodicity) if states split into groups that cycle. This causes the transition probabilities not converge. A chain avoids this if any state can stay put ($P_{ii} > 0$).

3 Birth/Death Processes: A Simpler Case

Some Markov chains, called **birth/death processes**, are easier to analyze because states are numbers (like 0 to m customers), and you can only move up one (a "birth"), down one (a "death"), or stay put(again, we set the chance of getting two at the same moment to 0.

- **Up** (to $i+1$): $P_i = p(1-q)$
- **Down** (to $i-1$): $Q_i = (1-p)q$
- **Stay**: $1 - P_i - Q_i$

This also models population growth.

To find steady-state probabilities, we balance the value between states i and $i+1$:

$$\pi_i P_i = \pi_{i+1} Q_{i+1}$$

This gives a pattern:

$$\pi_{i+1} = \pi_i \cdot \frac{P_i}{Q_{i+1}}$$

If the up and down chances are the same everywhere ($P_i = p$, $Q_i = q$), we use a load factor $\rho = p/q$ such that:

$$\pi_i = \pi_0 \rho^i, \quad \pi_0 = \frac{1}{\sum_{i=0}^{m} \rho^i}$$

Some special cases:

- If $\rho = 1$, then $\pi_i = 1/(m+1)$, so every state is equal likely, this is also called "random walk".

- If m is huge and $\rho < 1$ (service beats arrivals):

$$\pi_0 = 1 - \rho, \quad \pi_i = (1 - \rho)\rho^i$$

The average number of customers is:

$$E[\text{state}] = \frac{\rho}{1 - \rho}$$

This shows a stable queue when service is faster than arrivals, with fewer customers waiting.

4 Real-World Example: Phone Lines

Imagine setting up phone lines for a town. Calls come randomly (Poisson process, rate λ), and each call lasts a random time (exponential, average $1/\mu$). We want enough lines (B) so people rarely get a busy signal, but not too many to save money. We model this as a birth/death chain by splitting time into tiny slots of length δ. The state is the number of busy lines (0 to B), with transitions:

- **Up** (new call): $\lambda\delta$
- **Down** (call ends): $i\mu\delta$
- **Stay**: $1 - \lambda\delta - i\mu\delta$

The steady-state probabilities come from:

$$\pi_i \lambda \delta = \pi_{i+1}(i+1)\mu\delta \implies \pi_{i+1} = \pi_i \cdot \frac{\lambda}{(i+1)\mu}$$

Solving:

$$\pi_i = \pi_0 \cdot \frac{(\lambda/\mu)^i}{i!}, \quad \pi_0 = \frac{1}{\sum_{i=0}^{B} \frac{(\lambda/\mu)^i}{i!}}$$

The chance of a busy signal is π_B, which we want small (like 1%). For $\lambda = 30$ calls per minute, $\mu = 1/3$ per minute (calls last 3 minutes on average), the average number of calls is $\lambda/\mu = 90$. Calculations show $B \approx 106$ lines keep $\pi_B \approx 0.01$, handling random spikes in calls.

5 How Long Until Something Happens?

Sometimes, we want to know the chance of ending up in a specific group of states or how long it takes to get there. In a chain with absorbing states and transient states, the chance of reaching a specific group from state i, called a_i, we get:

$$a_i = \sum_j P_{ij} a_j$$

with $a_i = 1$ if i is in that group, 0 if in another. The expected time to reach any recurrent state, μ_i, is:

$$\mu_i = 1 + \sum_j P_{ij} \mu_j$$

with $\mu_i = 0$ for absorbing states. If there are multiple absorbing groups, we can combine them into one to simplify the calculation.

In a chain where all states connect, we might need to know how long it takes to first reach state s (τ_s) or return to it (recurrence time). These are similar equations, treating s as a trap by making its outgoing moves loop back.