

Lecture 13 - B

§8.2, 8.3 Markov Chains

T. Takahashi

Department of Mathematics and Statistics
University of New Mexico



We will answer these questions:

- What is a Markov chain?
- How do transition matrices model state evolution?
- Why do Markov chains always have a special eigenvalue?
- How do eigenvectors describe long-term behavior?



Markov Chains



Definition: Markov Chain

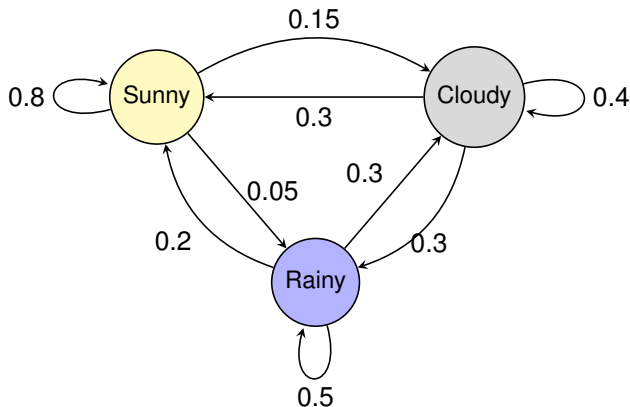
- A **Markov Chain** models how a system randomly transitions between a finite set of states.
- The probability of the next state depends only on the current state (not the past).
- States are denoted by integers $\{1, 2, \dots, n\}$
- Probability of jumping from state j to state i is P_{ij}

Example: Weather Prediction

- **System:** Weather conditions observed every 24 hours
- **States:** $\{1=\text{Sunny}, 2=\text{Cloudy}, 3=\text{Rainy}\}$
- **Probability:** $P_{32} = 0.3 \implies$ 30% chance of rain if today is cloudy



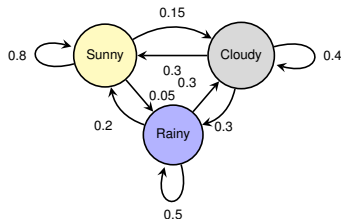
Transition Diagram



- Vertices (Sunny, Cloudy, Rainy) represent states
- Number on the arrows represent chance of switching between states
- Example: If today is Sunny, tomorrow will be sunny with an 80% chance



Transition Matrix



Transition Diagram

$$P = \begin{pmatrix} .8 & .3 & .2 \\ .15 & .4 & .3 \\ .05 & .3 & .5 \end{pmatrix}$$

Transition Matrix

- Summarize a diagram into a 3×3 **transition matrix** P
- Element P_{ij} is the probability of going to state i from state j .
- The columns of P must add up to 1



Transition Matrix

		current		
		Sunny	Cloudy	Rainy
future	Sunny	0.8	0.3	0.2
	Cloudy	0.15	0.4	0.3
	Rainy	0.05	0.3	0.5

- Column represents current state
- Row represents future state
- P_{ij} represents odds of transitioning to state i from state j
- $P_{12} = 0.3$ is the chance of being cloudy tomorrow if today is sunny



Modeling Example

Example

On any given day Gary is either cheerful (C), so-so (S), or glum (G).

- If he is cheerful today, then he will be C, S, or G tomorrow with respective probabilities 0.5, 0.4, 0.1.
- If he is feeling so-so today, then he will be C, S, or G tomorrow with probabilities 0.3, 0.4, 0.3.
- If he is glum today, then he will be C, S, or G tomorrow with probabilities 0.2, 0.3, 0.5.

- System: Gary's mood each day
- States: {C,S,G}
- Transition Matrix:

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



Evolution of a Markov Chain



State Vectors

- We may think of the Markov chain as an *operator* that describes how a **state vector** x_0 will evolve
- A state vector describes a distribution

$$x_0 = \begin{bmatrix} x_{01} \\ x_{02} \\ x_{03} \end{bmatrix}$$

where the elements are positive weights satisfying $x_{01} + x_{02} + x_{03} = 1$

- The product

$$x_1 = Px_0$$

describes the **new distribution** after one discrete time step



Exam Grade Example

Example

- If a student gets an A on an exam, the chances are 0.7, 0.2, 0.1 of getting an A, B, or C on the next exam.
 - If a student gets a B, the chances are 0.3, 0.5, 0.2.
 - If a student gets a C, the chances are 0.1, 0.6, 0.3.
-
- Suppose half the class gets an A, and half the class gets a C

$$x_0 = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

- On the next exam

$$x_1 = Px_0 = \begin{bmatrix} 0.7 & 0.3 & 0.1 \\ 0.2 & 0.5 & 0.6 \\ 0.1 & 0.2 & 0.3 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.2 \end{bmatrix}$$

40% will get an A, 40% will get a B, 20% will get a C



Keep Iterating

- After Exam 1:

$$x_0 = \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \end{bmatrix}$$

- After Exam 2:

$$x_1 = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.2 \end{bmatrix}$$

- After Exam 3:

$$x_2 = Px_1 = \begin{bmatrix} 0.42 \\ 0.40 \\ 0.18 \end{bmatrix}$$

- After Exam 4:

$$x_3 = Px_2 = \begin{bmatrix} 0.432 \\ 0.392 \\ 0.176 \end{bmatrix}$$



One Day vs Many Days

- The transition matrix P describes what happens in one time step
- What if we want to know what happens after:
 - 2 days?
 - 10 days?
 - A year?

Question

How do we compute the probability of moving between states over multiple days?



Matrix Powers

- If $x_1 = Px_0$ then x_2 will be given by

$$x_2 = Px_1$$

$$x_2 = P(Px_0)$$

$$x_2 = P^2x_0$$

- P^2 describes probabilities after two transitions
- In general P^k describes probabilities after k transitions
- Entry (i, j) of P^k is:

$$P_{ij}^k = \text{Pr}(\text{state } i \text{ after } k \text{ steps} \mid \text{given state } j \text{ initially})$$

Key Idea

Matrix multiplication combines probabilities across time.



Example

Example

If today is sunny, what is the probability that it will be rainy three days later?

$$P = \begin{pmatrix} .8 & .3 & .2 \\ .15 & .4 & .3 \\ .05 & .3 & .5 \end{pmatrix}$$

- Calculate $P^3 = PPP$

$$P^3 = \begin{bmatrix} 0.6365 & 0.4815 & 0.4400 \\ 0.2153 & 0.2665 & 0.2775 \\ 0.1483 & 0.2520 & 0.2825 \end{bmatrix}$$

- Sunny today $\implies$ column 1
- Rainy 3 days from now $\implies$ row 3
- The probability of being rainy 3 days if today is sunny is

$$P_{31}^3 = 0.1483 = 14.5\% \text{ chance}$$



Absorbing States



Broken Machine Example

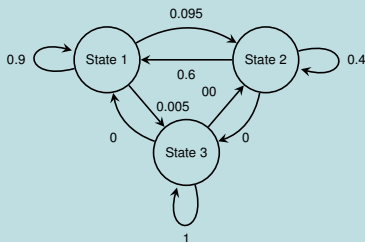
The Setting up the matrix

Example

An office printer has three states:

- State 1: Working
- State 2: Temporarily Broken
- State 3: Permanently Broken

Each day it is in one of these states. Find the transition matrix.



- $P_{i,j}$ contains probability going to state i from j



Broken Machine Example

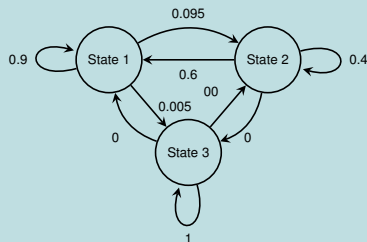
The Setting up the matrix

Example

An office printer has three states:

- State 1: Working
- State 2: Temporarily Broken
- State 3: Permanently Broken

Each day it is in one of these states. Find the transition matrix.



- $P_{i,j}$ contains probability going to state i from j

$$P = \begin{pmatrix} .9 & .6 & 0 \\ .095 & .4 & 0 \\ .005 & 0 & 1 \end{pmatrix}$$



Broken Machine Example

Word Problem

Example

Question If the machine is working on Day 1, what is the probability that it is still functional (that is, not permanently broken) a year from now?

- Take the 365'th power of the transition matrix:

$$P^{365} = \begin{pmatrix} .9 & .6 & 0 \\ .095 & .4 & 0 \\ .005 & 0 & 1 \end{pmatrix}^{365} = \begin{pmatrix} 0.1779 & 0.1792 & 0 \\ 0.0284 & 0.0286 & 0 \\ \mathbf{0.7937} & 0.7922 & 1 \end{pmatrix}$$

- $P_{3,1}^{365} = 0.7937 \implies$ there's a 79.22% chance that the machine is permanently broken after 365 days starting from state 1.

Answer

If the machine is working on Day 1, it will not be permanently broken with probability of about 20.63%.

Absorbing States

- This is the matrix one year from now

$$P^{365} = \begin{pmatrix} 0.1779 & 0.1792 & 0 \\ 0.0284 & 0.0286 & 0 \\ \mathbf{0.7937} & 0.7922 & 1 \end{pmatrix}$$

- This is the matrix after two years

$$P^{365} = \begin{pmatrix} 0.0367 & 0.0370 & 0 \\ 0.0059 & 0.0059 & 0 \\ \mathbf{0.9574} & 0.9571 & 1 \end{pmatrix}$$

- As as the number of days $n \rightarrow \infty$, the odds of the printer being permanently broken approaches 1

Definition: Absorbing State

A state i in a Markov chain is an **absorbing state** if it has the transition probability $P_{ii} = 1$

Absorbing States

- Suppose we begin with

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- As we iterate $x_n = Px_{n-1}$ we eventually get

$$\lim_{n \rightarrow \infty} x_n = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Absorbing States

- Suppose we begin with

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

- As we iterate $x_n = Px_{n-1}$ we eventually get

$$\lim_{n \rightarrow \infty} x_n = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- This is connected to the *power method*. The vector

$$x = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

is the dominant eigenvector of the matrix P , i.e. $Px = \lambda_1 x$



Power Method and Markov Chains



Steady State Vector

- In many Markov chains, the state distribution stabilizes:

$$x_{k+1} = Px_k \longrightarrow x_{k+1} \approx x_k$$

- When this happens, the system reaches a **steady state**
- A steady state vector x must satisfy:

$$Px = x$$

which is the same as

$$Px = \lambda x$$

where $\lambda = 1$

Definition: Steady State

- ▶ A steady state is an eigenvector of P with eigenvalue $\lambda = 1$.
- ▶ When normalized, the steady state tells the long-term distribution of states.

Connection to the Power Method

- Iterating a Markov chain means repeatedly computing:

$$x_{k+1} = Px_k$$

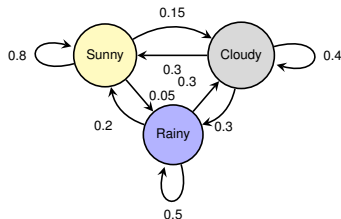
- This is exactly the power method
- The dominant eigenvector corresponds to steady state distribution of states in the Markov chain
- If x_0 is already normalized (sum of its elements is one) then you do not have to normalize with each iteration

Key Idea

If P is the transition matrix for a Markov chain, then power method will give the steady state distribution of states.



Weather Example



$$P = \begin{pmatrix} .8 & .3 & .2 \\ .15 & .4 & .3 \\ .05 & .3 & .5 \end{pmatrix}$$

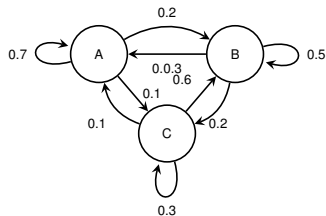
- Start with $x_0 = [1 \ 0 \ 0]^T$ and iterate $x_{n+1} = Px_n$

$$x_1 = \begin{bmatrix} 0.8 \\ 0.15 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0.695 \\ 0.195 \\ 0.11 \end{bmatrix}, \dots, \quad x_{16} = \begin{bmatrix} 0.56 \\ 0.24 \\ 0.20 \end{bmatrix}$$

- 56% days are sunny, 24% days are cloudy, 20% days are rainy



Exam Grade Example



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

- Start with $x_0 = [0.5 \ 0 \ 0.5]^T$ and iterate $x_{n+1} = Px_n$

$$x_1 = \begin{bmatrix} 0.4 \\ 0.4 \\ 0.2 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0.42 \\ 0.40 \\ 0.18 \end{bmatrix}, \dots, \quad x_{10} = \begin{bmatrix} 0.4423 \\ 0.3846 \\ 0.1731 \end{bmatrix}$$

- If there were ∞ exams the final distribution will be 44% A, 38% B, and 17% C



Broken Machine Example

```
1 >> P = [.9, .6, 0; .095, .4, 0; .005, 0, 1];
2 >> [V,E] = eigs(P)
3 V =
4      0      -0.6496      0.7096 % columns are eigenvectors
5      0      -0.1036     -0.7045
6      1.0000      0.7532     -0.0051
7
8 E =
9      1.0000      0      0 % eigenvalues
10      0      0.9957      0
11      0      0      0.3043
```

Key Idea

- The largest eigenvalue is $\lambda = 1$ and it has the eigenvector $v = [0 \ 0 \ 1]^T$
- The vector $v = [0 \ 0 \ 1]^T$ is giving us the steady state *probability distribution* of the states

Summary

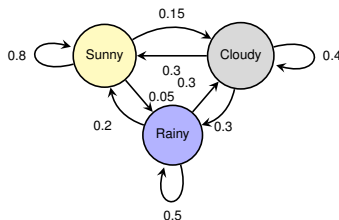


Markov Chains

- Suppose an object changes between states at discrete time steps

$$X = \{\text{sunny, cloudy, rainy}\}$$

- Probability of jumping between states are known



- System can be described with a *transition matrix* P



- **Power Method** is a simple algorithm to find the *dominant* eigenvector of a matrix
- Doing the power method on the *transition matrix* P yields the *steady-state distribution* of the system



Fun with Markov



Application: Text Prediction

- Many text models assume the next word depends only on the current word
- This is exactly a Markov chain assumption

Example

States = words in a vocabulary

$$P_{ij} = \Pr(\text{next word is } i \mid \text{current word is } j)$$

Key Idea

- Markov chains model local structure, not global meaning.
- If something can be described as a locally dependent sequence, you can use *Markov chains*



Application: Text Prediction

Setup:

- 1 Choose a text
- 2 List all unique words
- 3 Count how often each word follows another word
- 4 Normalize counts to form a transition matrix P

Text Generation:

- 1 Pick a starting word
- 2 Look up the corresponding probability column
- 3 Randomly choose the next word using those probabilities
- 4 Repeat



Application: Text Prediction

Example

Cat in the Hat — Dr. Seuss

The sun did not shine.

It was too wet to play.

So we sat in the house

All that cold, cold, wet day...

Generated Text

- The cat in bed, Oh! What made for fish!
- It stop you sir. Step this way. We fight these.
- The cat have come look. Then he said, "the box."
- It up the tags, it up the things, and he was gone with fear.



Application: Music Recommendation

- Spotify chooses what song to play next based on transition between music types
- The **states** are chosen to be categories:
 - a genre (Rock, Pop, Jazz)
 - an artist
 - a song cluster (similar style or mood)
- Each listen represents a transition from one state to another

Key Idea

Assume that the song you want to listen to next depends on what you just finished listening to.



How the Transition Matrix Is Built

- Music apps record user data:

Rock $\rightarrow$ Rock $\rightarrow$ Pop $\rightarrow$ Jazz

- Each transition is counted:
 - Rock $\rightarrow$ Rock (120,000 times)
 - Rock $\rightarrow$ Pop (30,000 times)
 - Pop $\rightarrow$ Jazz (5,000 times)
- These counts form a **transition count matrix**

From Counts to Probabilities

Each column is normalized so probabilities sum to 1.



Example

Let the states be music genres:

{Rock, Indie, Pop, Hip-Hop, Jazz}

We define the transition probabilities

$$P_{ij} = \Pr(\text{next song is genre } i \mid \text{current song is genre } j)$$

- Columns represent the **current genre**
- Rows represent the **next genre**
- Each column sums to 1

Example (illustrative numbers):

$$P = \begin{bmatrix} 0.60 & 0.20 & 0.10 & 0.10 & 0.30 \\ 0.20 & 0.50 & 0.20 & 0.10 & 0.20 \\ 0.10 & 0.20 & 0.50 & 0.30 & 0.10 \\ 0.05 & 0.05 & 0.15 & 0.40 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.10 & 0.35 \end{bmatrix}$$



Web Pages

THE WEB AS A DIRECTED GRAPH

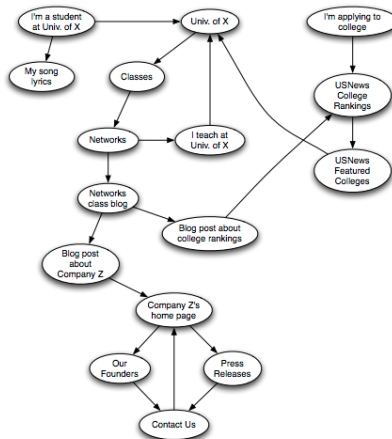


Figure: Web pages can be turned into a matrix.



Web Structure

- Let W be the set of web pages that can be reached by following a chain of hyperlinks starting from a page on Google.
- Let n be the number of pages in W .

Connectivity and Transition Matrices

- Let G be the $n \times n$ **connectivity matrix** of W , where

$$G_{i,j} = \begin{cases} 1 & \text{if there is a hyperlink from page } j \text{ to page } i, \\ 0 & \text{otherwise.} \end{cases}$$

- Rescale the **transition matrix** G by normalizing each column of G by the number of outgoing links from page j :

$$G_{i,j} = \frac{G_{i,j}}{\text{outdegree}(j)}$$

How Google Search Works

- 1 Find a list of n websites containing keywords
- 2 Generate a directed graph from the webpages
- 3 Generate a transition matrix P
- 4 Do power method on P
- 5 Sort the dominant eigenvector from greatest to least
- 6 Display websites corresponding to sorted eigenvector

Result: A list of websites ordered in terms of relevancy

