4
Discrete Time Markov Chains

Figure 4.1: Russian Mathematician Andrei Andreevich Markov (*courtesy: Wikipedia*).

MARKOV CHAINS were first formulated as a stochastic model[1] by Russian mathematician Andrei Andreevich Markov. Markov spent most of his professional career at St. Petersburg University and the Imperial Academy of Science. During this time, he specialized in the theory of numbers, mathematical analysis, and probability theory. His work on Markov chains utilized finite square matrices (stochastic matrices) to show that the two classical results of probability theory, namely, the *weak law of large numbers* and the *central limit theorem*, can be extended to the case of sums of *dependent* random variables.

Markov chains have wide scientific and engineering applications in statistical mechanics, financial engineering, weather modeling, artificial intelligence, and so on. In this chapter we will look at a few applications as we build the concepts of Markov chains. Additionally, we will also implement a technique (using Markov chains) to solve a simple and practical engineering problem related to aircraft control and automation.

[1] A. A. Markov. "Extension of the law of large numbers to dependent quantities (in Russian)". In: *Izvestiia Fiz.-Matem. Obsch. Kazan Univ., (2nd Ser.)* 15 (1906), pp. 135–156

4.1 Chapter objectives

The chapter objectives are listed as follows.

1. Students will learn the definition and application of Markov processes.
2. Students will learn the definition of *stochastic matrix* (a.k.a. *probability transition matrix*) and perform simple matrix calculations to compute conditional probabilities.
3. Students will learn to solve engineering and scientific problems based on discrete time Markov chains using multi-step transition probabilities.
4. Students will learn to compute return times and hitting times to Markov states.
5. Students will learn to classify different Markov states.
6. Students will learn to use the techniques of discrete time Markov chains introduced in this chapter to solve a complex engineering problem related to flight control operations.

4.2 Chapter project: Automatic prediction of control laws of an aircraft using the Viterbi algorithm

4.2.1 Prologue: tracking aircraft control laws

In this project, we will learn to implement the *Viterbi algorithm* to automatically predict the operational flight control law by analysing the *pitch* data of the aircraft. The Viterbi algorithm is a practical application of discrete time Markov chains. This project illustrates, in a simplified manner, the framework within which an aircraft's operational performance is monitored in real time by ground crew. It also demonstrates, as an example, how mathematical technology and engineering redundancies work together towards developing newer and safer flight experiences for passengers. Modern aircrafts use fly-by-wire control systems. This means that control surfaces like ailerons, rudders, etc. are maneuvered under the command of electronic signals originating from the flight control computers instead of the pilot's manual inputs. The scope of the fly-by-wire system is determined by the active flight control law. There are primarily three different control laws in an Airbus A-330 aircraft: (i) normal law, (ii) alternate law, and (iii) direct law.

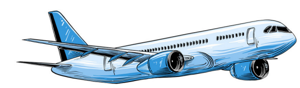

Figure 4.2: An Airbus A-330 is shown here steadily climbing towards its cruising altitude. The aircraft is under the command of the autopilot and operating under *Normal* flight control law.

Normal law offers a variety of automated protection to the flight envelope, eg. automatic stall protection, bank angle protection, high speed protection, etc. The aircraft can be commanded by the autopilot in this mode. Normal law may be applicable in *ground, flight* or *flare (landing)* modes. The aircraft performance is considered to be optimal if it is operational under normal law during its flight. On the other hand, when one or more of the sensors or control surfaces are impaired, then the flight may be forced to operate under alternate law. Some automatic protections like stall protection, bank angle protection, and high speed protection may be lost depending on the exact nature of the failure(s). This may happen as a consequence of faults in the horizontal stabilizer, a single elevator fault, loss of a yaw-damper actuator, loss of slat or flap position sensors, etc. (ALT-1 law) or due to engine flame outs, faults in two inertial or two air-data reference units, damage to all spoilers, aileron faults, etc. (ALT-2 law). Aircraft operation under this alternate control law will often require some direct intervention by the pilot(s). Finally, a further degradation of flight control affairs results in the activation of the direct law. Under this law, autopilot function is always lost and most of the automatic flight envelope protections are lost. Pilots will have to manually recover and fly the airplane. This type of degradation of flight controls may result from dual-engine flame outs, dual elevator failures, etc.

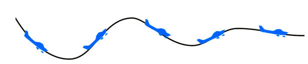

Figure 4.3: An erratic pitch profile of an aircraft reveals a likely faulty elevator and a flight operation governed by alternate or direct law.

During the course of a flight, tracking and recording the operational control law of an aircraft can reflect the performance profile of the aircraft.[2] One of the ways this real time health of the aircraft may be captured by ground control and maintenance crew is by analysing the binary pitch up/ down data transmitted to it via the satellite based *Aircraft Communications Addressing and Reporting System* (ACARS). Eg., frequent changes in the pitch of the aircraft (nose up-nose down motions) may be a result of faulty elevators (part of the horizontal stabilizer), thereby activating the alternate or direct control laws. This example illustrates that the pitch data can serve as an important metric for predicting the active control law. In this project, we will use the Viterbi algorithm to predict the profile of the active control law during flight by analyzing the real time pitch data profile.

[2] An aircraft whose flight signature is predominantly governed by normal law can be assumed to be in better operational health than an aircraft whose flight signature is interspersed with several stints of alternate and direct control modes.

We will return to this case-study in a subsequent section of this chapter after we have developed some conceptual background on Markov chains.

4.3 Definition: Markov chain

A Markov chain (or equivalently, a Markov process) is a stochastic process describing a series of possible events whereby the probability of an event at a given instant depends solely on the outcome of the previous event.

In this chapter, we will consider only those Markov processes whose outcomes are discrete events. These are known as *Discrete Time Markov Chains* (DTMC). Eg., we may think of a game of badminton. The outcome of a given shot in a rally is a discrete event such as a *smash*, a *drop*, a *lift*, etc. Any given shot by a player likely depends on the previous shot by the opponent, and most likely not so much on any previous shot played by either player. To elucidate this further, consider a *smash* played by player A. It is nearly impossible that a return shot by player B will also be a smash, it can perhaps be a lift or a drop. So essentially, a game of badminton can be modelled as a Markov process. We will analyze this particular example in greater detail later in this chapter.

Figure 4.4: A *smash* by the far-end player cannot be returned by a *smash* by the near-end player in a game of badminton which can be modelled as a Markov process.

Mathematically, we may denote a sequence of events as $\{X_n\}_{n \in I, n \geq 0}$, where the subscript n is a non-negative integer that indexes time. X_n is a random variable which may take a value from the set of possible outcomes (events), $\mathcal{S} = \{x_n, x_{n-1}, ..., x_1, x_0\}$. Then, $\{X_n\}$ describes a Markov process when

$$P\left(X_n = x_n \middle| X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, ..., X_0 = x_0\right) = P\left(X_n = x_n \middle| X_{n-1} = x_{n-1}\right). \qquad (4.1)$$

Consequentially, a Markov process is characterized by the property of memorylessness.

4.3.1 *Stochastic matrix or probability transition matrix,* $\mathbb{P}$

In the previous paragraph, the set $\mathcal{S}$ forms the sample space comprising all possible outcomes (events) of the Markov process. The elements of this set are known as the states of the Markov chain. The probability (or likelihood) of transition between two states is compiled in matrix form. This matrix is known as the *stochastic matrix* or the *probability transition matrix* and is denoted by p_{ij} (equivalently, $p_{i,j}$) or simply, $\mathbb{P}$. Here, the subscripts i, j refer to the fact that we are considering a transition from state i to state j. i, j can take values from $\{1, 2, 3, .., n\}$. It may be noted here that the sum of the entries of any row of $\mathbb{P}$ must be identical to one due to one of the axioms of probability. The following example will illustrate how we may construct this matrix.

Figure 4.5: A game of roulette in a casino. A gambler's fortune may be modelled as a Markov process.

4.3.2 *Example: Gambler's ruin*

Consider a game of gambling in which, on any turn, we win ₹ 1 with probability $p = 0.4$ or lose ₹ 1 with probability $p' = (1 - p) = 0.6$. Suppose we adopt a strategy that we quit playing upon making a fortune of ₹ 10 while the casino throws us out if we have lost all our money. Construct a suitable stochastic matrix.
Let X_n := amount of money we have after 'n plays'. $\mathcal{S} = \{0, 1, 2, ..., 10\}$. Then clearly, for all $i = \{1, 2, ..., 9\}$, the following is true based on the rules of the gambling game,

$$P\left(X_{n+1} = i+1 \middle| X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0\right) = P\left(X_{n+1} = i+1 \middle| X_n = i\right) = p_{i,i+1} = 0.4. \qquad (4.2)$$

Likewise, $p_{i,i-1} = 0.6$. Additionally, $p_{0,0} = 1$ because there is no chance of making money when we have lost everything since the casino will throw us out. Similarly, $p_{10,10} = 1$ because ₹ 10 is the maximum allowable fortune we can accumulate. States 0 and 10 are known as absorbing states as there is no *escape* from these two states in this gambling model. Finally, we may now write the stochastic matrix as follows,

$$\mathbb{P} = \begin{array}{cc} & \begin{array}{ccccccc} j=0 & j=1 & j=2 & \cdot & \cdot & \cdot & j=10 \end{array} \\ \begin{array}{c} i=0 \\ i=1 \\ i=2 \\ \cdot \\ \cdot \\ \cdot \\ i=9 \\ i=10 \end{array} & \begin{pmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 0.6 & 0 & 0.4 & \cdot & \cdot & \cdot & 0 \\ 0 & 0.6 & 0 & 0.4 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 0.6 & 0 & 0.4 \\ 0 & 0 & \cdot & \cdot & 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{4.3}$$

4.4 *Multi-step transition probabilities*

The stochastic matrix $\mathbb{P}$ pertains to a single step in the stochastic transition of the underlying process. But in many practical cases, we may be interested in knowing the likelihood of transitions between states in more than one step. Eg., mathematically, the probability of transitioning from state i to state j in $m(>1)$ steps may be expressed as follows.

$$p^m_{i,j} \equiv p^m(i,j) = P\big(X_{n+m} = j \big| X_n = i\big); \quad m > 1. \tag{4.4}$$

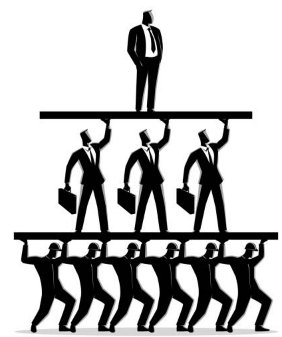

Figure 4.6: Class motility in society can also be analyzed by a Markov model.

4.4.1 *Example: Social motility*

Consider that X_n represents the social class of a family in the n^{th} generation. Assume that there are broadly three social groups based on income, viz., lower=1, middle=2, upper=3. Based on a certain demographic and income analysis, the motility within this society was captured succinctly by the following stochastic matrix.

$$\mathbb{P} = \begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.5 & 0.2 \\ 0.2 & 0.4 & 0.4 \end{pmatrix} \end{array}. \tag{4.5}$$

If Ginny's parents were of middle income class, what is the probability that Ginny belongs to the upper income class and her children belong to the lower income group? Here we are asked to find $P\big(X_2 = 1, X_1 = 3 \big| X_0 = 2\big)$, where X_0 refers to Ginny's parents generation, X_1 refers to her generation while X_2 refers to her children's generation. We will derive the answer from first principles and subsequently claim that the same may be deduced by formal inspection of the respective $p_{i,j}$ entries.

$$P(X_2 = 1, X_1 = 3 \mid X_0 = 2) \underset{\text{def. } P(A|B) = \frac{P(A,B)}{P(B)}}{=} \frac{P(X_2 = 1, X_1 = 3, X_0 = 2)}{P(X_0 = 2)}$$

$$\underset{\text{multiply \& divide by same term}}{=} \frac{P(X_2 = 1, X_1 = 3, X_0 = 2)}{P(X_1 = 3, X_0 = 2)} \times \frac{P(X_1 = 3, X_0 = 2)}{P(X_0 = 2)}$$

$$\underset{\text{conditional probability}}{=} P(X_2 = 1 \mid X_1 = 3, X_0 = 2) \times P(X_1 = 3 \mid X_0 = 2)$$

$$\underset{\text{Markov property}}{=} P(X_2 = 1 \mid X_1 = 3) \times P(X_1 = 3 \mid X_0 = 2)$$

$$\underset{\text{entries of } \mathbb{P}}{=} p_{3,1} p_{2,3} \underset{\text{terms re-arranged}}{=} p_{2,3} p_{3,1} = 0.2 \times 0.2 = 0.04.$$

Now that we have seen all the intermediary steps starting from the expression that we originally set out to evaluate, it may be convenient to notice that the answer could have been formally read out by multiplying the probability entries for the successive transitions 2 (middle) $\rightarrow$ 3 (upper) and 3 (upper) $\rightarrow$ 1 (lower) .
Let us now suppose that Sheila, from the same society, belongs to the lower income group; what is the probability that her grandchildren will belong to upper class? This probability can be easily evaluated to be $p^2(1,3)$, i.e., $(1,3)$ entry of the $\mathbb{P}^2$ matrix.

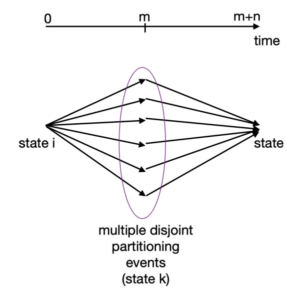

Figure 4.7: Schematic illustration of the Chapman-Kolmogorov equation to compute multi-step transition probabilities through intermediary partitioning events $X_m = k,\ k \in \mathcal{S}$.

4.4.2 *Chapman-Kolmogorov equation*

Multi-step transition probabilities for Markov models may also be computed by considering the Chapman-Kolmogorov equation given below.

$$p_{i,j}^{m+n} = \sum_{k \in S} p_{i,k}^{m} p_{k,j}^{n}. \tag{4.6}$$

Let us find out why the above may be true. Consider that while transitioning from state i to state j, we pass through an intermediary state k. This k may be any of the states from the set $\mathcal{S}$. We will consider the intermediary event $X_m = k$ as a partitioning event.[3]

$$\begin{aligned} p_{i,j}^{m+n} &= P(X_{n+m} = j \mid X_0 = i) \\ &= \sum_{k \in \mathcal{S}} P(X_{m+n} = j, X_m = k \mid X_0 = i) \\ &= \sum_{k \in \mathcal{S}} P(X_{m+n} = j \mid X_m = k, X_0 = i) P(X_m = k \mid X_0 = i) \\ &\underset{\text{Markov property}}{=} \sum_{k \in \mathcal{S}} p_{k,j}^{n} p_{i,k}^{m} = \sum_{k \in S} p_{i,k}^{m} p_{k,j}^{n}. \end{aligned}$$

[3] **cf.** law of total probability:

$$P(A) = \sum_{B_i \in \mathcal{S}} P(A|B_i) P(B_i),$$

where the B_is are the partitioning events!

The Chapman-Kolmogorov equation will be revisited when we study continuous time

Markov chains (CTMC). This equation will be used to derive the *detailed balance* condition and finds many applications as a tool to investigate behavior of equilibrium systems.

4.5 Distribution of states

Consider that for a given Markov model, the stochastic matrix, $\mathbb{P}$ is known. Further, let us suppose that an initial probability distribution of states is also known. We may be interested in knowing the probability distribution of states at a later time (or in the long run).

Mathematically, we may denote the initial distribution of k-states, $\{s_1 \;\; s_2 \;\; \dots \;\; s_k\}$, by

$$\vec{\mu}^{(0)} = (\mu_1^{(0)} \;\; \mu_2^{(0)} \;\; \dots \;\; \mu_k^{(0)}) = \big(P(X_{01} = s_1) \;\; P(X_{02} = s_2) \;\; \dots \;\; P(X_{0k} = s_k)\big),$$

where $\sum_{i=1}^{k} \mu_i^{(0)} = 1$ due to one of the axioms of probability. Here, the 0 in the superscripts and subscripts represent the <u>initial</u> time instant and the indices $i = \{1, 2, \dots, k\}$ refer to the states. Then the probability distribution of states after n-steps may be written as

$$\vec{\mu}^{(n)} = \vec{\mu}^{(0)} \mathbb{P}^n. \tag{4.7}$$

To find the long run distribution of states, we take the limit $n \to \infty$ in equation (4.7).

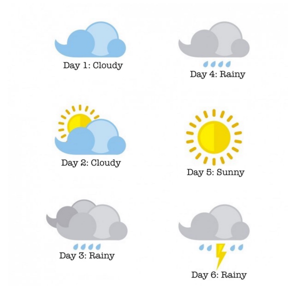

Figure 4.8: Daily weather forecasts can be modelled based on a Markov model.

4.5.1 Example: Weather model

Consider a simple weather model that predicts weather on a given day as follows: (i) the weather stays the same on any given day as the previous day 75% of the time, and (ii) the weather changes from day to day 25% of the time. For simplicity we may only consider two weather patterns in this model, viz., sunny, and rainy. What is the long time weather forecast given that on a certain day it is observed to be sunny?
We must first begin by constructing the stochastic matrix $\mathbb{P}$ based on the transitions between the weather patterns (states): s for sunny, and r for rainy.

$$\mathbb{P} = \begin{array}{c c} & \begin{array}{cc} s & r \end{array} \\ \begin{array}{c} s \\ r \end{array} & \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix} \end{array}. \tag{4.8}$$

$$\begin{aligned}
\vec{\mu}^{(1)} &= \vec{\mu}^{(0)} \mathbb{P} = (1 \;\; 0) \begin{pmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix} = (0.75 \;\; 0.25) \\
\vec{\mu}^{(2)} &= \vec{\mu}^{(0)} \mathbb{P}^2 = \vec{\mu}^{(1)} \mathbb{P} = (0.625 \;\; 0.375) \\
&. \\
&. \\
\vec{\mu}^{(\infty)} &= \vec{\mu}^{(0)} \mathbb{P}^{\infty} = (0.5 \;\; 0.5)
\end{aligned}$$

This is the equilibrium distribution of states.[4] Thus, the long time weather forecast is that it is equally likely to be sunny or rainy according to this model.

[4] What is $\vec{\mu}^{(\infty)} \mathbb{P}$?

From all of the aforementioned discussions on the applications of a Markov model, we may glean that the stochastic matrix encodes almost all of the information about the underlying Markov process.

4.5.2 *Example: Badminton game, what is a winning strategy?*

Consider a Markov model of a game of badminton. For simplicity, let us consider that a player chooses to play one of three shots, viz., *smash (S)*, *drop (D)*, and *lift (L)*. The objective is to devise a winning strategy given a match situation. Based on the data generated over several games, the following table lists the probability of a return shot played by a player given a certain type of shot played by their opponent.

shot	return shot	probability
D	D	1/3
D	L	1/3
D	S	0
L	D	1/5
L	L	1/5
L	S	2/5
S	L	2/5
S	D	1/5
S	S	0

1. Identify an appropriate state space for the Markov model.
2. Construct the stochastic matrix $\mathbb{P}$.
3. Given a *lifted* serve, what are the chances that there is a winner in three shots?
4. In a rally, if a player receives a lift from their opponent, which shot option maximizes his chance of winning the rally in the return shot?

Figure 4.9: Should I *smash* or feign and *drop* to win the rally now?

Solution: The Markov analysis is as follows.

1. State space, $\mathcal{S} = \{D, L, S, W\}$ where W refers to a *winning* shot.
2. The stochastic matrix may be computed based on the data table.[5]

$$\mathbb{P} = \begin{array}{c} \\ D \\ L \\ S \\ W \end{array} \begin{array}{c} \begin{array}{cccc} D & L & S & W \end{array} \\ \begin{bmatrix} 1/3 & 1/3 & 0 & 1/3 \\ 1/5 & 1/5 & 2/5 & 1/5 \\ 1/5 & 2/5 & 0 & 2/5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \tag{4.9}$$

3. $\vec{\mu}^{(3)} = \vec{\mu}^{(0)}\mathbb{P}^3 = (0.1316 \;\; 0.1476 \;\; 0.1067 \;\; 0.6142) \equiv p^3_{LW}$ which is the (L, W) entry of the $\mathbb{P}^3$ matrix.[6]

4. Essentially we must compute $P(X_2 = W, X_1 = ?|X_0 = L)$ where ? may be any one of L, D or S. So we will compute each of $(p_{LD} \times p_{DW})$, $(p_{LS} \times p_{SW})$, $(p_{LL} \times p_{LW})$ and pick the strategy that maximizes the resulting probability. It turns out that the respective probabilities are $0.06667,\ 0.16,\ 0.04$. Hence, a return *smash* will most likely win us the rally in that shot.[7]

[5] It is a good practice to check that each of the rows of $\mathbb{P}$ add up to unity!
[6] In an actual game, an overall optimal strategy must account for maximizing the chances of winning the rally in a certain number of shots as well as possibly continuing the rally long enough to tire out the opponent!
[7] An AI assisted training session based on such a Markov model will enable a player to devise optimal shot selections during an actual game!

Figure 4.10: A bank teller queue may be a Markov model with recurring events such as a state when the queue is empty.

4.6 *Recurring events in Markov chains*

In a finite state system with stochastic transitions between states, a Markov chain may visit certain states (events) multiple times. The time interval (steps) between such successive visits to a given state may itself be a random number. Repeated visits and the inter-arrival time between such visits may be of interest depending on the application. Eg., a kiosk of a bank teller may have a long queue of customers waiting to be served. The number of persons in the queue may represent the states of a Markov chain. The arrival process can be modelled in terms of a Poisson process while the departures depend on exponentially distributed service times. The bank operations policy may rely on analyzing the time interval between successive visits to state 0 and/or by the sojourn time (total time a customer spends in the system) which may involve analyzing recurring events in the Markov model of the bank teller system.[8]

[8] It turns out that the Markov model of a teller system may be a *continuous* time Markov chain (CTMC) depending on how we model the time elapse process. We will study CTMCs in greater detail in a subsequent chapter.

4.6.1 *Definition: Hitting probability*

Let $\{X_n\}_{n\in\{0,\mathbb{I}^+\}}$ represent a Markov chain with state space $\mathcal{S}$. Let $A \subset \mathcal{S}$. Further, let us define the following:

$$\begin{aligned} T_A \quad &:= \quad \text{first time the chain hits A starting from outside (or inside) A} \\ &:= \quad \min\{n \geq 0 | X_n \in A\}, \end{aligned} \tag{4.10}$$

with $T_A = 0$ if $X_0 \in A$ and $T_A = \infty$ if $\{n \geq 0 | X_n \in A\} = \{\}$. Then, we may define the *probability of hitting state A* through a state $l \in A$ starting from a state $k \in \mathcal{S}$ as below,

$$g_k(l) = P(X_{T_A} = l | X_0 = k). \tag{4.11}$$

initial state

final state

The above definition helps us to calculate the chance of hitting a certain state A beginning from a given state. Why is this useful to know? In the context of the bank teller example above, if we find that $g_s(0) = 0$ for any $s > 0$,[9] then this will likely invite the attention of the bank manager to change the operational policy of the bank (eg., by introducing additional tellers). This may be required to ensure that the bank teller receives a much needed break from serving customers during a long shift.

[9] number of customers s in an operational queue is positive

4.6.2 *Iterative formula for hitting probability*

Let $k \in S \setminus A$.[10] We have $T_A \geq 1$ given $X_0 = k$.

[10] $S \setminus A \equiv S - \{A\}$, i.e. the set $\mathcal{S}$ minus the contents of the set A.

$$
\begin{aligned}
g_k(l) = P\big(X_{T_A} = l \big| X_0 = k\big) &\underset{\text{sum over partitioning events}}{=} \sum_{m\in\mathcal{S}} P\big(X_{T_A} = l, X_1 = m \big| X_0 = k\big) \\
&\underset{\textbf{cf. }\text{the law of total probability}}{=} \sum_{m\in\mathcal{S}} P\big(X_{T_A} = l \big| X_1 = m, X_0 = k\big) P\big(X_1 = m \big| X_0 = k\big) \\
&\underset{\text{Markov property}}{=} \sum_{m\in\mathcal{S}} \underbrace{P\big(X_{T_A} = l \big| X_1 = m\big)}_{g_m(l)} \underbrace{P\big(X_1 = m \big| X_0 = k\big)}_{p_{km}}. \\
&= \sum_{m\in\mathcal{S}} g_m(l) p_{km}. \qquad (4.12)
\end{aligned}
$$

Therefore, the iterative formula for $g_k(l)$ is given as follows.

$$\boxed{g_k(l) = \sum_{m\in\mathcal{S}} p_{km} g_m(l)} \text{ where } k \in \mathcal{S} \setminus A, \;\; l \in A. \qquad (4.13)$$

4.6.3 *Definition: Absorbing state*

We have seen absorbing states in earlier examples. Formally, consider the case $p_{kl} = \mathbb{I}_{k=l}$[11] for all $k, l \in A$. Here, the state k is an absorbing state, i.e. $\{X_n\}$ is trapped (absorbed) in $A \subset \mathcal{S}$.[12]

[11] Here $\mathbb{I}_{k=l} = 1$ only when $k = l$. $\mathbb{I}$ is an *indicator* function.

[12] It may be interesting to note the following:
$$\boxed{\sum_{l\in A} g_k(l) + P\big(T_A = \infty \big| X_o = k\big) = 1}$$

4.6.4 *Iterative formula for mean hitting times and mean absorption times*

We have stated earlier that the time duration between successive visits to a certain state of a Markov chain is a random variable. In many applications, we may be interested to know the expected value of this random variable. Let us define this expected value as

$$h_k(A) := E\big(T_A \big| X_0 = k\big). \qquad (4.14)$$

Clearly, $h_k(A) = 0 \;\; \forall k \in A \subset \mathcal{S}$. Further, $\forall k \in \mathcal{S} \setminus A$,

$$h_k(A) = E\big(T_A \big| X_0 = k\big) \underset{\text{sum over partitioning events}}{=} \sum_{m \in \mathcal{S}} E\big(T_A, X_1 = m \big| X_0 = k\big)$$

$$\underset{\textbf{cf.}\text{ the law of total expectation}}{=} \sum_{m \in \mathcal{S}} E\big(T_A \big| X_1 = m, X_0 = k\big) P\big(X_1 = m \big| X_0 = k\big)$$

$$\underset{\text{Markov property}}{=} \sum_{m \in A} \underbrace{E\big(T_A \big| X_1 = m\big)}_{1} p_{km} + \sum_{m \in \mathcal{S} \setminus A} \underbrace{E\big(T_A \big| X_1 = m\big)}_{1 + h_m(A)} p_{km}$$

$$= \sum_{m \in A} p_{km} + \sum_{m \in \mathcal{S} \setminus A} (1 + h_m(A)) p_{km}$$

$$= \sum_{m \in \mathcal{S}} p_{km} + \sum_{m \in \mathcal{S} \setminus A} h_m(A) p_{km}$$

$$\underset{\text{axiom: } \sum_{i \in \mathcal{S}} p_i = 1}{=} 1 + \sum_{m \in \mathcal{S} \setminus A} p_{km} h_m(A) + \sum_{m \in A} p_{km} \cancel{h_m(A)}^{0}$$

$$\underset{\text{adding a term whose value is } 0}{=} 1 + \sum_{m \in S} p_{km} h_m(A). \tag{4.15}$$

Summarizing, we have

$$\boxed{h_k(A) = 1 + \sum_{m \in S} p_{km} h_m(A)} \text{ for all } k \in \mathcal{S} \setminus A. \tag{4.16}$$

In the above derivation $E\big(T_A \big| X_1 = m\big) = 1$ in the first summation because the Markov chain has already moved one step forward ($X_0 \to X_1$) and has then hit the desired state $m \in A$. In the second summation, $E\big(T_A \big| X_1 = m\big) = 1 + h_m(A)$ because the chain has moved one step forward and the counting process must be reset again until the chain hits the desired state A.

4.6.5 *Definition: First return time and its mean*

The time (in number of steps) taken to make a first return to a certain state $y \in \mathcal{S}$ is defined as follows:

$$T_y^r := \min\big\{n \geq 1 \big| X_n = y\big\}; \quad y \in \mathcal{S}, \tag{4.17}$$

with $T_y^r = \infty$ if $X_n \neq y \;\; \forall n \geq 1$. Note $T_y^r = T_y$ if $X_0 \neq y$.

T_y^r is a random variable. Let us define $\mu_x(y) = E\big(T_y^r \big| X_0 = x\big) \geq 1$. Clearly, when $x = y$, we have the definition of mean return time. Following the spirit of the derivation in sec. 4.6.4, it is possible to derive an iterative formula for the mean return time. Here, we simply state the final result.

$$\boxed{\mu_x(y) = 1 + \sum_{m \in S, m \neq y} p_{xm} \mu_m(y)} \tag{4.18}$$

In the following section we will work out an illustrative example using the aforementioned iterative formulae.

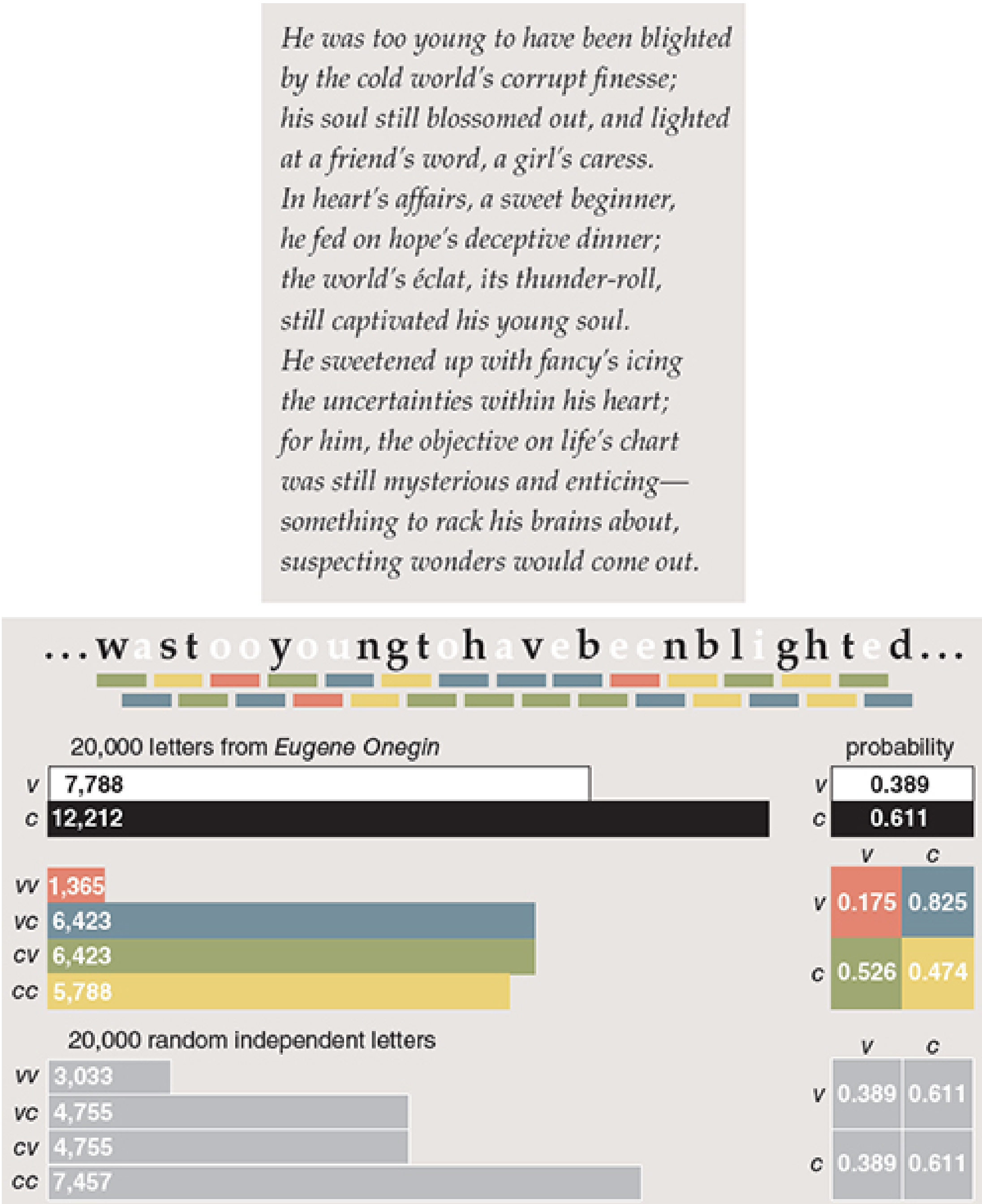

Figure 4.11: Markov conducted a statistical experiment to understand the structure of language by analyzing Alexander Pushkin's poem *Eugene Onegin*. Shown here is a single stanza from the poem (English translated version) with the characteristic rhyming of the words. A Markov model analysis of the language used in the poem (colored illustrations) shows the distinctively different *alternation* noticed in the poem compared to a random sequence of letters (grey illustrations) of the same length. Here v refers to vowels and c refers to consonants. The stochastic matrices $\mathbb{P}$ are shown on the right. Can you compute the mean hitting times and mean return times to consonants and vowels? (courtesy: *First Links in the Markov Chain* by *Brian Hayes*, AMERICAN Scientist, **101 (2)**, pg. 92, March-April 2013, DOI: 10.1511/2013.101.92)

4.6.6 *Example: Return times in a game of badminton*

Let us divide each side of a badminton court into two quadrants (*play zones*) labelled (1) and (2) as shown in the picture alongside (**cf.** Figure 4.15). These quadrants may be regarded as states of a simple Markov model. Data collected over several games of badminton may enable us to populate a simple stochastic matrix as follows.

$$\mathbb{P} = \begin{array}{c c} & \begin{array}{cc} (1) & (2) \end{array} \\ \begin{array}{c} (1) \\ (2) \end{array} & \begin{bmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix} \end{array} \tag{4.19}$$

1. During a rally, given a shot by a player from play zone (1), after how many shots (on an average) does the shuttle return to quadrant (1) on either side of the net?

   Now, let us re-define the Markov chain by constructing new states corresponding to the direction of shots played. Eg., the states of the new model are (11) (corresponding to a shot $1 \to 1$) and so on.

2. Construct the stochastic matrix $\mathbb{P}_{\text{new}}$ for this new model.

3. Given a valid service $1 \quad \to \quad 1$, what is the average number of shots before either player can expect a shot $2 \to 1$ from their opponent?

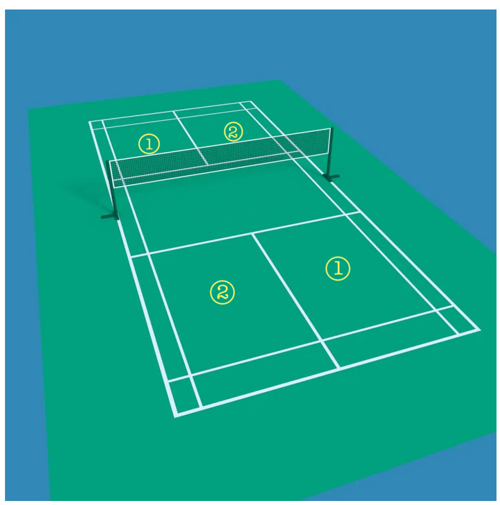

Figure 4.12: A badminton court with designated *play zones* (states). In a rally, the shuttle may be considered to make random visits to these states. A player's preparedness to receive the shuttle, effectively, may be guided by mean return and hitting time estimates which may be included during their practice sessions before an official game.

Solution:

1. We begin with the simple model $\mathbb{P}$ and use equation (4.18) to compute $\mu_1(1)$.

$$\begin{aligned} \mu_1(1) &= 1 + p_{12}\mu_2(1) = 1 + \frac{1}{3}\mu_2(1) \\ \mu_2(1) &= 1 + p_{22}\mu_2(1) = 1 + \frac{2}{3}\mu_2(1) \end{aligned}$$

whence $\mu_2(1) = 3$ and $\mu_1(1) = 2$, i.e. on an average, every second shot in a rally returns to play zone 1 given a serve from quadrant 1.

2. Let us now construct $\mathbb{P}_{\text{new}}$. The relevant states are 11, 12, 21, and 22. Now, for a state transition 11 $\rightarrow$ 11, a sequence of following shots must be played: $1 \rightarrow 1$ and $1 \rightarrow 1$. Note that the naming convention of the states is set up in such a way that the last numeral of the current state must be the same as the first numeral of the next state. This means transitions like 11 $\rightarrow$ 21, 21 $\rightarrow$ 12, etc. are not possible. Consequently, the transition 11 $\rightarrow$ 11 can happen with a probability $2/3$ because this transition is wholly concurrent with the event whereby the most recent shot is $1 \rightarrow 1$ which has a probability $2/3$ associated with it. Similar arguments enable us to construct $\mathbb{P}_{\text{new}}$ as follows.

$$\mathbb{P}_{\text{new}} = \begin{array}{c} \\ 11 \\ 12 \\ 21 \\ 22 \end{array} \begin{array}{c} \begin{array}{cccc} 11 & 12 & 21 & 22 \end{array} \\ \begin{bmatrix} 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 2/3 \\ 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 2/3. \end{bmatrix} \end{array} \tag{4.20}$$

3. Now using equation (4.18), we have the following system of equations:

$$\begin{aligned} \mu_{11}(21) &= 1 + \frac{2}{3}\mu_{11}(21) + \frac{1}{3}\mu_{12}(21), \\ \mu_{12}(21) &= 1 + \frac{2}{3}\mu_{22}(21), \text{ and} \\ \mu_{22}(21) &= 1 + \frac{2}{3}\mu_{22}(21); \end{aligned}$$

whose solution is $\mu_{22}(21) = 3$, $\mu_{12}(21) = 3$ and $\mu_{11}(21) = 6$. Therefore, given a valid service $1 \rightarrow 1$, either player will have to wait 6 shots on an average before they may expect a shot $2 \rightarrow 1$.

4.7 *Chapter project: Automatic prediction of aerodynamic control laws of an aircraft using the Viterbi algorithm*

4.7.1 *Interlude: the Viterbi Algorithm*

Now that we have learnt the fundamental ideas of Markov chains, we are in a position to deduce the Viterbi algorithm. We will use the Viterbi algorithm to predict the control laws prevailing in the aircraft cockpit.

I. **Stochastic and Emission matrices**: We will consider a certain stochastic process with the following state space of dimension K, $S = \{s_1, s_2, ..., s_K\}$. Associated with this process is a T dimensional observation set $\mathbf{Y} = \{y_1, y_2, ..., y_T\}$ from amongst a possible N dimensional observation space $O = \{o_1, o_2, ..., o_N\}$. Note: $y_n \in O$. Further, consider an initial probability distribution given by $\mathbf{\Pi} = \{\pi_1, \pi_2, ..., \pi_K\}$.[13] The probability transition matrix $\mathbb{P}$ is a $K \times K$ matrix with entries

$$p_{ij}(t) := \text{probability of transitioning from state } s_i \text{ to state } s_j = Prob(x_t = s_j | x_{t-1} = s_i),$$

and the emission matrix $\mathbb{E}$ is a $K \times N$ matrix with entries

$$e_{ij}(t) := \text{probability of observing } o_j \text{ from state } s_i = Prob(y_t = o_j | x_{t-1} = s_i).$$

Succinctly, we will often write $s_i \equiv i$ and $o_j \equiv j$ where it must be understood that $x_t = i$ refers to the random variable x_t taking the state s_i and $y_t = j$ refers to the random variable y_t being assigned the observable o_j. The goal of the prediction algorithm is to forecast the most likely sequence of states (events) $\mathbf{X} = \{x_1, x_2, ..., x_T\}, x_n \in S$ given a prescribed sequence of observables $\mathbf{Y}$, i.e. we need to compute

$$\text{argmax}_{\mathbf{X}} Prob(\mathbf{X}|\mathbf{Y}) = \text{argmax}_{\mathbf{X}} Prob(\mathbf{Y}|\mathbf{X}) Prob(\mathbf{X}) = \text{argmax}_{\mathbf{X}} Prob(\mathbf{Y}, \mathbf{X}).$$

Here $\text{argmax}(f(x))$ returns the value of x at which the function $f(x)$ attains its maximum. In this project, we will implement the Viterbi algorithm to predict the most likely sequence of states that corresponds to a sequence of associated observables assuming a Markovian stochastic model (also known as the *Hidden Markov Model* (HMM)).

II. **Construction and essential calculations of the Viterbi algorithm**: In what follows, we will fix the notation $Prob(X_1 = x_1) \equiv Prob(x_1) \equiv \pi_1$. Note that if $T = 2$, then

$$\begin{aligned} Prob(\mathbf{Y}, \mathbf{X}) &\equiv Prob(y_1, y_2, x_1, x_2) \\ &= Prob(y_1, y_2, x_2 | x_1) Prob(x_1) \\ &= Prob(y_1, y_2 | x_2, x_1) Prob(x_2 | x_1) Prob(x_1) \\ &= Prob(y_1 | y_2, x_2, x_1) Prob(y_2 | x_2, x_1) p_{12} \pi_1 \\ &= Prob(y_1 | x_1, x_2, y_2) Prob(y_2 | x_2) p_{12} \pi_1 \\ &= Prob(y_1 | x_1) Prob(y_2 | x_2) p_{12} \pi_1 \end{aligned} \tag{4.21}$$

In general, we have

$$\begin{aligned} Prob(\mathbf{Y}, \mathbf{X}) &\equiv Prob(\mathbf{Y} = y_1, ..., y_T, \mathbf{X} = x_1, ..., x_T) \\ &= \underbrace{Prob(x_1)}_{\pi_1} Prob(y_1 | x_1) \underbrace{Prob(x_2 | x_1)}_{p_{12}} Prob(y_2 | x_2) \cdots Prob(y_T | x_T) \end{aligned}$$

The Viterbi algorithm involves *recursively* computing the Viterbi entries $V_{k,t}$

$$\begin{aligned} V_{k,t} &:= \max Prob((y_1, ..., y_t), (x_1, ..., x_t = k)) \\ &= \text{probability of the best (most likely) sequence of states (ending with state } k \text{, i.e. } x_t = k) \\ &\quad \text{corresponding to the sequence of observables } (y_1, ..., y_t). \end{aligned}$$

II.1. **Recursive computation of $V_{k,t}$**:

By comparing the terms on the right hand side of eq. (4.22) and the definition of the Viterbi entries above, we see that $V_{k,t}$ can be obtained recursively and consequently using

[13] As a trial example, we may think of a state space $S = \{\text{rainy, sunny}\}$, an observational space $O = \{\text{walking, shopping, cleaning}\}$ and a sequence of observations of activity patterns of Billoo, the handyman as Y = {walking, walking, shopping, walking, cleaning} over the past five days. The objective here is to find the most likely sequence of (hidden) states X corresponding to the sequence of observables Y. E.g., one possible likely outcome may be X = {sunny, sunny, cloudy, sunny, rainy}. But instead of guesswork, the readers are encouraged to use the predictions of the Viterbi algorithm to list the weather pattern for the corresponding days.

Figure 4.13: American electrical engineer Andrew Viterbi invented the *Viterbi algorithm* which is a *dynamic programming algorithm* originally used for convolutional codes over noisy digital communication systems. It has since found multiple applications in natural language processing, computational linguistics, bioinformatics, speech recognition, etc. (*courtesy: Wikipedia*).

the argmax function, we can find the most likely sequence of events. The algorithm includes calculation of the following three important terms.

- $V_{k,t} = \max_{\alpha \in S}\big(Prob(y_t = j \big| x_t = k) p_{\alpha k} V_{\alpha,t-1}\big) = \max_{\alpha \in S}\big(e_{kj} p_{\alpha k} V_{\alpha,t-1}\big)$

  with $V_{k,1} \overset{set}{=} Prob(y_1 = o_m \big| x_1 = k)\pi_k = e_{km}\pi_k$, and
- $x_T = \operatorname{argmax}_{\alpha \in S}\big(V_{\alpha,T}\big)$.
- x_{t-1} = back_pointer(x_t, t) = value of x used to compute $V_{k,t}$ $\forall t > 1$.

II.2. **Aesthetics of dynamic programming algorithms**: The Viterbi algorithm belongs to a class of algorithms known as *dynamic programming*. This class of algorthms was developed by American applied mathematician Richard Bellman in 1953. The classic problem solved by this family of algorithms is the *travelling salesman problem*.[14] Another classic puzzle that can be solved by dynamic programming methodology is the tower of Hanoi game. These are just a few examples of many diverse applications of the dynamic programming method in general, and the Viterbi algorithm in particular.

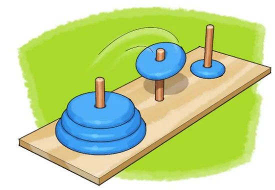

Figure 4.14: The *tower of Hanoi* game (*courtesy: Science Buddies, Sabine De Brabandere,* Scientific American, 2017.)

[14] *The Traveling Salesman Problem: A Computational Study* by *David L. Applegate, et. al.,* *Princeton University Press, 2007.*

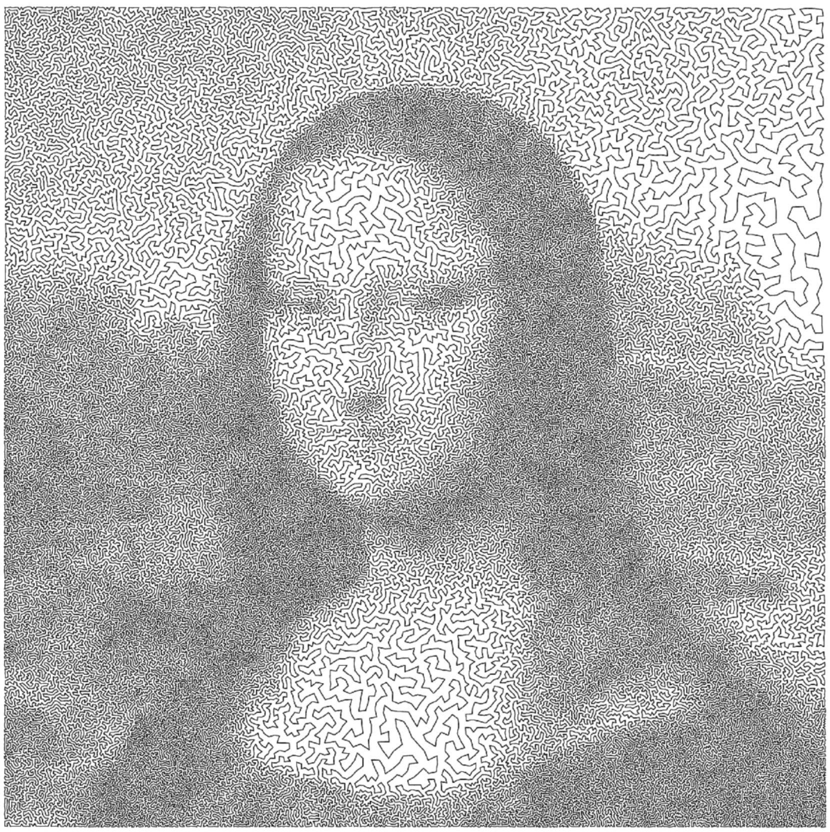

Figure 4.15: Robert Bosch of Oberlin College and his collaborators have used the travelling salesman problem to generate artwork. This is yet another example of the proximal inter-relationship between mathematics, computational algorithms and aesthetics (*courtesy:* `http://www.math.uwaterloo.ca/tsp/data/art/`. Further reading: *Continuous line drawings via the traveling salesman problem* by *Robert Bosch* and *Adrianne Herman*, Operations Research Letters, **32**, pp. 302–303, 2004.

In the subsequent sections, we will present you with a strategy to implement the Viterbi algorithm in your computer, test your code using the weather model example presented in the margin in the previous page, and finally use the tested version of the algorithm to predict the outcomes of the prevailing control laws in the aircraft as introduced in the prologue.

Software Implementation

Pseudocode of the Viterbi algorithm:

INPUT: $S, \mathbf{\Pi}, \mathbb{E}, \mathbb{P}, \mathbf{Y} = \{y_1, y_2, ..., y_T\}$.

Part I: Initialization.

```
 for each i of K states
   viterbi_prob(i,1) = π_i * e_{iy_1}
   viterbi_path(i,1) = 0
end for
```

Part II: Compute Viterbi probabilities and Viterbi path.

```
 for each j of T-1 observations starting with T=2
   for each i of K states
      viterbi_prob(i,j) = max_{α∈S}(e_{iy_j} * p_{αi} * viterbi_prob(α, j-1))
      viterbi_path(i,j) = argmax_{α∈S}(e_{iy_j} * p_{αi} * viterbi_prob(α, j-1))
   end for
end for
x_T = s_{z_T} where z_T := argmax_{α∈S}(viterbi_prob(α, T))
```

The appearance of e_{ij} in the computation of `viterbi_path(i,j)` is unnecessary because it is non-negative and independent of α (so you may choose to skip it).

Part III: Retracking the most likely path $\mathbf{X}$.

```
 for each j of T-1 observations from T to 2
   x_{j-1} = s_{z_{j-1}} where z_{j-1} = viterbi_path(z_j, j)
end for
```

OUTPUT: $X = \{x_1, x_2, ..., x_T\}$

The student may test the veracity of the Viterbi implementation by first attempting the trial example introduced earlier.[15]

[15] Use the Markovian model explained above to predict the weather for the last five days. Assume the initial weather distribution $\mathbf{\Pi} = \{0.43, 0.57\}$, the probability transition matrix $\mathbb{P} = \begin{pmatrix} 0.2 & 0.8 \\ 0.4 & 0.6 \end{pmatrix}$, where state 1 is *rainy* and state 2 is *sunny*, and the probability emission matrix $\mathbb{E} = \begin{pmatrix} 0.2 & 0.4 & 0.4 \\ 0.3 & 0.25 & 0.45 \end{pmatrix}$, where the columns (observations) are labelled in order of *walking*, *shopping* and *cleaning*, respectively.

4.8 *Classification of Markov states and advanced topics*

The behavior of a Markov chain is characterized by the properties of the stochastic matrix $\mathbb{P}$ and its states. The states of a Markov chain can be classified based on the entries of $\mathbb{P}$.

4.8.1 *Definition: Communicating states*

A state $j \in \mathcal{S}$ is accessible from a state $i \in \mathcal{S}$, i.e. $(i) \mapsto (j)$, if there exists a finite integer $n \geq 0$, s.t. $p_{ij}^n := P(X_n = j | X_0 = i) > 0$.[16]

Further, if $(i) \mapsto (j)$ and $(j) \mapsto (i)$, then $(i) \leftrightarrow (j)$, i.e. i and j communicate. When two states communicate with each other, they are said to belong to the same class.

[16] $(i) \mapsto (i)$ even if $p_{ii} = 0$.

4.8.2 Example: Communicating states

Consider below a stochastic matrix of a Markov chain with states labelled as $1, 2, 3$, and 4.

$$\mathbb{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1/3 & 1/3 & 1/3 & 0 \\ 1/2 & 0 & 0 & 1/2 \\ 2/5 & 1/5 & 0 & 2/5 \\ 1/4 & 1/4 & 1/2 & 0 \end{bmatrix} \end{array} \tag{4.22}$$

In this model:

1. $3 \leftrightarrow 3$ even though $p_{33} = 0$.
2. $2 \mapsto 3$ even though $p_{23} = 0$ because $p_{24} > 0$ and $p_{43} > 0$; hence there exists N s.t. $p_{23}^N > 0$. In fact, $p_{23}^2 = 0.4167 > 0$.
   Further $p_{32} = 1/5 > 0$. Therefore, $2 \leftrightarrow 3$.

Since the binary relation $\leftrightarrow$ satisfies *reflexivity*, *symmetry*, and *transitivity* properties; $\leftrightarrow$ is an equivalence relation.[17]

[17] The equivalence relation $\leftrightarrow$ induces a partition of $\mathcal{S}$ into disjoint subsets $A_1, A_2, ..., A_m$ s.t. $\mathcal{S} = \bigcup_{i=1}^m A_i$. Additionally, the following are true.

1. $i \leftrightarrow j$ for all $i, j \in A_q$,
2. $i \not\leftrightarrow j$ whenever $i \in A_p$ and $j \in A_q$ and $p \neq q$.

4.8.3 Definition: Irreducible and Reducible Markov chains

A Markov chain is irreducible if all states belong to one class, i.e. if all states communicate with each other.[18]

[18] An irreducible Markov chain is also known as an ergodic chain.

Example: Consider a Markov chain with a stochastic matrix,

$$\mathbb{P} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 2/3 & 0 & 0 & 1/3 \\ 1/3 & 0 & 0 & 0 & 2/3 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

In this example all states communicate with each other.

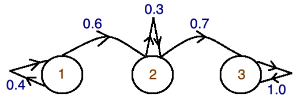

Figure 4.16: In this Markov model, state 3 is an absorbing state and does not communicate with states 1 and 2.

A Markov chain that is not irreducible is said to be reducible, i.e. there is at least one state (or a group of states) from which the chain cannot re-visit other states not in that group.

In Figure 4.16, we have used a graphical representation of Markov chain showing the states within circles and the probabilities of transition between states are represented by the numbers along with the arrows.

4.8.4 *Mean number of returns to a state*

Let $q_{ij} \equiv p_{ij}^n = P\big(X_n = j \big| X_0 = i\big)$ for some $n \geq 1$ represents the probability of return to state j in a finite time starting from state i. Now let us define the number of visits to state j by the chain $\{X_n\}_{n \in I,\ n \geq 0}$ as follows

$$\begin{aligned} E\big(R_j \big| X_0 = i\big) &= \sum_{m=0}^{\infty} m P\big(R_j = m \big| X_0 = i\big) \\ &= \sum_{m=1}^{\infty} m q_{ij} q_{jj}^{m-1}(1 - q_{jj}) \\ &= (1 - q_{jj}) q_{ij} \sum_{m=1}^{\infty} m q_{jj}^{m-1} \\ &= (1 - q_{jj}) q_{ij} \frac{1}{(1 - q_{jj})^2} \\ &= \frac{q_{ij}}{1 - q_{jj}}. \end{aligned} \tag{4.23}$$

Here we have used the identity $\sum_{m=1}^{\infty} m r^{m-1} = \frac{1}{(1-r)^2}$, where $|r| \leq 1$. The terms to the right hand side of the second equality may require further explanation. Firstly, the first visit to state j from state i must happen in $n \leq m$ steps with probability q_{ij}. This must be followed by $m - 1$ re-visits to state j starting from state j with probability q_{jj}^{m-1}. This is true because the count for the re-visits to state j happens beginning with state j as the chain is reset as $X_0 = j$ after the first visit to state j. Since the summand of interest pertains to m visits to state j (and no more), we must account for the probability $(1 - q_{jj})$ of no additional visits to state j after the m^{th} visit.

4.8.5 *Definition: Recurrent states*

$i \in \mathcal{S}$ is recurrent if $q_{ii} = p_{ii}^n = 1$.[19] Additionally,

1. state i is recurrent if and only if $E\big(R_i \big| X_0 = i\big) = \infty$,

2. state i is recurrent if and only if $P\big(R_i = \infty \big| X_0 = i\big) = 1$.

[19] Let $\{X_n\}_{n \geq 0,\ n \in I}$ be a Markov chain with finite state space $\mathcal{S}$; then $\{X_n\}_{n \geq 0,\ n \in I}$ has at least one recurrent state.

A recurrent state $i \in \mathcal{S}$ is said to be positive recurrent if $\mu_i(i) < \infty$, and is null recurrent if $\mu_i(i) = \infty$.

4.8.6 *Definition: Transient states*

A state $i \in \mathcal{S}$ is transient when it is *not* recurrent, i.e.,

$$P\big(R_i = \infty \big| X_0 = i\big) < 1.$$

Further,

1. $i \in \mathcal{S}$ is transient if and only if
$$E\big(R_i \big| X_0 = i\big) < \infty.$$

2. $i \in \mathcal{S}$ is transient if and only if
$$\sum_{n=1}^{\infty} p_{ii}^n < \infty.$$

4.8.7 *Periodicity of a Markov chain*

The period of a state i is the greatest common divisor (denominator) of all integers $n > 0$ for which $p_{ii}^n > 0$.[20]

A Markov chain is aperiodic if it has period one.

[20] Periodicity is a class property. Eg., if states i and j belong to the same class, then they have the same period.

Example: Consider a Markov chain with $\mathbb{P} = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$ Here $p_{ii} = 0$, and $p_{ii}^2 = 0$ but $p_{ii}^3 = 1 > 0$ and so on. Thus, the chain has period three.

4.9 *Chapter project: Automatic prediction of aerodynamic control laws of an aircraft using the Viterbi algorithm*

4.9.1 *Epilogue: results of the Viterbi code for predicting the aircraft control laws*

Consider the following probability transition matrix $\mathbb{P}$ and probability emission matrix $\mathbb{E}$ that is available from the Airbus database.

$$\mathbb{P} = \begin{pmatrix} 0.7 & 0.1 & 0.2 \\ 0.4 & 0.5 & 0.1 \\ 0.2 & 0.3 & 0.5 \end{pmatrix}, \quad \mathbb{E} = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \\ 0.2 & 0.8 \end{pmatrix}$$

and $\mathbf{\Pi} = \{0.8,\ 0.1,\ 0.1\}$. At a certain time, the company receives the following sequence of pitch measurements at five minute intervals. Devise a model using the Viterbi algorithm to predict the corresponding sequence of control laws that will likely be activated during the same time instant.

Pitch data: 'up', 'down', 'down', 'down', 'down', 'up', 'up', 'down', 'down', 'down', 'down'.

4.10 *Selected bibliography*

1. *Markov Chains* by *J. R. Norris*, Cambridge University Press, 2017.
DOI: `https://doi.org/10.1017/CBO9780511810633`

2. *Understanding Markov Chains: Examples and Applications* by *Nicolas Privault*, Springer (second edition), 2018.
DOI: `https://doi.org/10.1007/978-981-13-0659-4`

3. *Essentials of Stochastic Processes* by *Richard Durrett*, Springer (second edition), 2012.
DOI: `10.1007/978-1-4614-3615-7`

4. *Probability and Measure* by *P. Billingsley*, John Wiley and Sons (second edition), 1990.

4.11 Exercise problems

1. Consider the Markov chain $\{X_n\}_{n\geq 0}$ with state space $\mathcal{S} = \{1, 2\}$ and transition matrix

$$\mathbb{P} = \begin{array}{c c} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 2 \end{array} & \begin{bmatrix} 0.3 & 0.7 \\ 0.5 & 0.5 \end{bmatrix} \end{array}$$

   (a) Compute $P(X_7 = 2, X_5 = 1 | X_4 = 1, X_3 = 2)$.
   (b) Compute $E(X_2 | X_1 = 2)$.

2. (***Mean hitting times***) Our bunny, whose name is Honey, hops around on a triangle. At each step he moves to one of the other two vertices at random (his decision is based on the flip of a fair two-sided coin). What is the expected time taken by Honey to get from vertex 1 to vertex 2?

3. Consider a Markov chain $\{X_n\}_{n\geq 0}$ on the state space $\{0, 1, 2, 3, 4\}$ with stochastic matrix

$$\mathbb{P} = \begin{array}{c c} & \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 1/4 & 1/4 & 1/4 & 1/4 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

   (a) Draw the graph of this chain.
   (b) Find the periods of states 0, 1, 2, and 3.
   (c) Which states are absorbing, recurrent and transient?
   (d) Is the Markov chain reducible? Why?

4. (***Snakes and ladder***) Consider a nine-squares snakes and ladder board as shown in Figure 4.17. At each turn, a player tosses a fair coin and advances one or two steps forward depending on whether the outcome is a tail or a head respectively. Upon landing at the base of a ladder, the player can climb to the top of the ladder, whereas falling at the mouth of a snake brings them down to the tail of the snake.

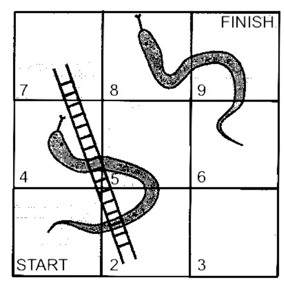

Figure 4.17: How fast will this game be done on an average?

   (a) Construct an appropriate Markov model and $\mathbb{P}$.
   (b) In how many turns on an average can the game be completed by a player?
   (c) What is the probability that a player who has made it to square 6 will complete the game before falling to the "START"?

5. (***Population genetics***)[21] In a certain genetics model, we consider an $n - by - n$ array of cells. Each cell is initially colored any one of k different colors. At each step, a cell is chosen at random. This cell then chooses one of its eight neighbors at random and assumes the color of that neighbor. At the boundaries, we may consider a periodic wrapping of left-right and top-bottom columns and rows. The missing diagonal neighbor of any corner cell may be replaced by the cell that is in the diametrically opposite corner. With these

[21] *Results for The Stepping Stone Model for Migration in Population Genetics* by *S. Sawyer*, Annals of Probability, **vol. 4**, pp. 699–728, 1979.

boundary adjustments, each cell in the array is adjacent to exactly eight other cells. A state in this Markov chain is a description of the color of each cell. The number of states is k^{n^2}. Even for small n, the size of the state space is enormous. We will analyze this model with the help of a computer simulation. Begin with a random initial configuration of $k = 2$ colors with $n = 20$ and comment on the long run behavior of the model in terms of the prevalent colors. Repeat the simulation by taking $k = 5$ and $n = 50$. State your observations with possible reasons.

6. (***Time to freedom***) Our squirrel from an earlier chapter is stuck in a maze with four cells, labelled as $1, 2, 3$, and 4 as shown in Figure 4.18. The outside world is the squirrel's pathway to freedom (labelled as 0). The route to freedom can only be accessed through cell 4. The squirrel starts initially in cell 2. From each cell, the squirrel can move to either of the neighboring cells with equal probability. We assume that at each move the squirrel acts independent of the past (our squirrel is not keen to learn from its past mistakes). How many moves will our squirrel make (on an average) before attaining freedom?

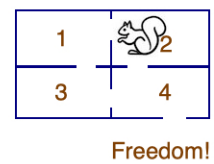

Figure 4.18: How many hops to freedom?

7. (***Wandering Daisy***) Our friend Daisy has decided to take a day off from work on a cloudy day. Instead, she is having thoughts about visiting the apple orchard just outside her hamlet but is concerned about the impending rain. Let us analyze her prospects and see if she would make it to the orchard given this uncertainty in her mind.

   Deterministic mind: Daisy starts walking from her home towards the orchard. Half way through she changes her mind and starts returning home because she thinks that it might rain soon. The clouds begin clearing up soon or so it seems. So half way through her return, she changes her mind and starts walking towards the orchard again. Once again half way through that she starts returning home and so on. Construct a mathematical model of her location at every inflection point and comment on her eventual destination. *Hint: Identify Daisy's home as the point "zero" and the orchard as the point "one". Formulate a sequence $\{X_n\}$ where X_n denotes the position at the n^{th} inflection point. What is the limiting value of X_n as $n \to \infty$?*

   Stochastic mind version 1:[22] Daisy starts at "zero" (home), goes half way through and then flips a fair coin. If the coin comes out heads, she continues towards "one" (orchard) and if the coin comes out tails, she turns back towards "zero". Again half way through whatever direction she is headed, she flips a fair coin and either continues in that direction or goes in the opposite way based on the outcome of the toss. Construct a mathematical model of her location at every inflection point and comment on her eventual destination. *Hint: Formulate a sequence X_n where X_n denotes the position at the n^{th} inflection point. Is X_n a Markov chain? Why? What is the limiting value of X_n as $n \to \infty$?*

   Stochastic mind version 2: Now, each time Daisy has to decide on the direction, she uses a biased coin with probability $p(\neq 1/2)$ for heads and then goes a fraction α of the distance in that direction. Construct a mathematical model of her location at every inflection point and comment on her eventual destination.

8. (***Invariant or stationary distributions***) Given a probability measure $\vec{\pi}$, we say that $\vec{\pi}^{(\infty)}$ is invariant or stationary if

$$\vec{\pi}^{(\infty)}\mathbb{P} = \vec{\pi}^{(\infty)}.$$

[22] The random sequence X_n generated by Daisy in the stochastic case is a model that is applicable in many real life practical situations. Eg., let Y_m be a random sequence denoting the level of water in a tank at time instances m, $m \geq 0, m \in \mathbb{I}$. Let $\tau_1, \tau_2, \tau_3, ...$ be the times at which the sequence $\{Y_m\}$ has a local minimum or maximum. Let $X_m = Y_{\tau_m}$. Suppose the tank has a global minimum at "zero" and a global maximum normalised to be "one". Then the sequence $\{Y_{\tau_m}\}$ must behave analogous to Daisy's wanderings. The model for the tank can be similarly thought of as a model for stock prices, amount of rainfall, inventory level or any other randomly varying sequence in a bounded interval.

Consider a Markov chain $\{X_n\}_{n\geq 0}$ with states arranged on the vertices of a triangle. The transition probabilities between the states are as follows: $p_{12} = 1,\ p_{23} = 1/2,\ p_{31},\ p_{33} = 1/2,\ p_{22} = 1/2$.

(a) Construct $\mathbb{P}$.

(b) Find the stationary distribution of states $\vec{\pi}^{(\infty)}$.

9. (***Diffusion as a Markov process***) Gas molecules move about randomly in a box which is divided into two halves symmetrically by a partition. A hole is made in the partition. Suppose there are N molecules in the box.

(a) Show that the number of molecules on one side of the partition just after a molecule has passed through the hole evolves as a Markov chain.

(b) What are the transition probabilities?

(c) What is the invariant distribution of this chain?

10. (***Time reversal and entropy***) Past and future are independent of each other in a Markov process. So this entails an inherent time symmetry.[23] However, convergence to stationary distribution of states is asymmetrical in time, i.e. a highly organised state decays to a disorganised one (the invariant distribution) when viewed backwards. This is analogous to a situation that embodies an increase in entropy. Therefore, time symmetry in the absolute sense demands that we begin at equilibrium. Consider a Markov chain $\{X_n\}_{0\leq n\leq N}$ with

$$\mathbb{P} = \begin{pmatrix} 0 & 2/3 & 1/3 \\ 1/3 & 0 & 2/3 \\ 2/3 & 1/3 & 0 \end{pmatrix}$$

and $\vec{\pi} = (1/3\ \ 1/3\ \ 1/3)$ is invariant.

(a) Compute the stochastic matrix $\hat{\mathbb{P}}$ of the chain $Y_n = X_{N-n}$.

(b) Is the chain reversible? (*Hint: Check if* $\hat{\mathbb{P}} = \mathbb{P}$*?*)

□

[23] A Markov chain in equilibrium, run backwards, is again a Markov chain. The transition matrix may, however, be different.