1

Introduction

Markov chain is named after Prof. Andrei A. Markov (1856-1922) who first published his result in 1906. He was born on 14 June 1856 in Ryazan, Russia and died on 20 July 1922 in St. Petersburg, Russia. Markov enrolled at the University of St. Petersburg, where he earned a master's degree and a doctorate degree. He is a professor at St. Petersburg and also a member of the Russian Academy of Sciences. He retired in 1905, but continued his teaching at the university until his death. Markov is particularly remembered for his study of Markov chains. His research works on Markov chains launched the study of stochastic processes with a lot of applications. For more details about Markov and his works, we refer our reader to the following interesting website [220].

In this chapter, we first give a brief introduction to the classical theory on both discrete and continuous time Markov chains. We then present some relationships between Markov chains of finite states and matrix theory. Some classical iterative methods for solving linear systems will also be introduced. They are standard numerical methods for solving Markov chains. We will then give the theory and algorithms for standard hidden Markov model (HMM) and Markov decision process (MDP).

1.1 Markov Chains

This section gives a brief introduction to discrete time Markov chain. Interested readers can consult the books by Ross [180] and Häggström [103] for more details.

Markov chain concerns about a sequence of random variables, which correspond to the states of a certain system, in such a way that the state at one time epoch depends only on the one in the previous time epoch. We will discuss some basic properties of a Markov chain. Basic concepts and notations are explained throughout this chapter. Some important theorems in this area will also be presented.

Let us begin with a practical problem as a motivation. In a town there are two supermarkets only, namely Wellcome and Park'n. A marketing research indicated that a consumer of Wellcome may switch to Park'n in his/her next shopping with a probability of $\alpha(> 0)$, while a consumer of Park'n may switch to Wellcome in his/her next shopping with a probability of $\beta(> 0)$. The followings are two important and interesting questions. The first question is that what is the probability that a Wellcome's consumer will still be a Wellcome's consumer in his/her nth shopping? The second question is what will be the market share of the two supermarkets in the town in the long-run? An impoartant feature of this problem is that the future behavior of a consumer depends on his/her current situation. We will see later this marketing problem can be formulated by using a Markov chain model.

1.1.1 Examples of Markov Chains

We consider a stochastic process

$$\{X^{(n)}, n = 0, 1, 2, \ldots\}$$

that takes on a *finite* or *countable* set M.

Example 1.1. Let $X^{(n)}$ be the weather of the nth day which can be

$$M = \{sunny, windy, rainy, cloudy\}.$$

One may have the following realization:

$$X^{(0)} = \text{sunny},\ X^{(1)} = \text{windy},\ X^{(2)} = \text{rainy},\ X^{(3)} = \text{sunny},\ X^{(4)} = \text{cloudy}, \ldots.$$

Example 1.2. Let $X^{(n)}$ be the product sales on the nth day which can be

$$M = \{0, 1, 2, \ldots, \}.$$

One may have the following realization:

$$X^{(0)} = 4, \quad X^{(1)} = 5, \quad X^{(2)} = 2, \quad X^{(3)} = 0, \quad X^{(4)} = 5, \ \ldots.$$

Remark 1.3. For simplicity of discussion we assume M, the *state space* to be $\{0, 1, 2, \ldots\}$. An element in M is called a *state* of the process.

Definition 1.4. *Suppose there is a fixed probability P_{ij} independent of time such that*

$$P(X^{(n+1)} = i | X^{(n)} = j, X^{(n-1)} = i_{n-1}, \ldots, X^{(0)} = i_0) = P_{ij} \quad n \geq 0$$

where $i, j, i_0, i_1, \ldots, i_{n-1} \in M$. Then this is called a Markov chain process.

Remark 1.5. One can interpret the above probability as follows: the conditional distribution of any future state $X^{(n+1)}$ given the past states

$$X^{(0)}, X^{(2)}, \ldots, X^{(n-1)}$$

and present state $X^{(n)}$, is *independent* of the *past states* and *depends* on the *present state* only.

Remark 1.6. The probability P_{ij} represents the probability that the process will make a transition to state i given that currently the process is state j. Clearly one has

$$P_{ij} \geq 0, \quad \sum_{i=0}^{\infty} P_{ij} = 1, \quad j = 0, 1, \ldots.$$

For simplicity of discussion, in our context we adopt this convention which is different from the traditional one.

Definition 1.7. *The matrix containing P_{ij}, the transition probabilities*

$$P = \begin{pmatrix} P_{00} & P_{01} & \cdots \\ P_{10} & P_{11} & \cdots \\ \vdots & \vdots & \vdots \end{pmatrix}$$

is called the one-step transition probability matrix of the process.

Example 1.8. Consider the marketing problem again. Let $X^{(n)}$ be a 2-state process (taking values of $\{0, 1\}$) describing the behavior of a consumer. We have $X^{(n)} = 0$ if the consumer shops with Wellcome on the nth day and $X^{(n)} = 1$ if the consumer shops with Park'n on the nth day. Since the future state (which supermarket to shop in the next time) depends on the current state only, it is a Markov chain process. It is easy to check that the transition probabilities are

$$P_{00} = 1 - \alpha, \quad P_{10} = \alpha, \quad P_{11} = 1 - \beta \quad \text{and} \quad P_{01} = \beta.$$

Then the one-step transition matrix of this process is given by

$$P = \begin{pmatrix} 1 - \alpha & \beta \\ \alpha & 1 - \beta \end{pmatrix}.$$

Example 1.9. (Random Walk) Random walks have been studied by many physicists and mathematicians for a number of years. Since then, there have been a lot of extensions [180] and applications. Therefore it is obvious for discussing the idea of random walks here. Consider a person who performs a random walk on the real line with the counting numbers

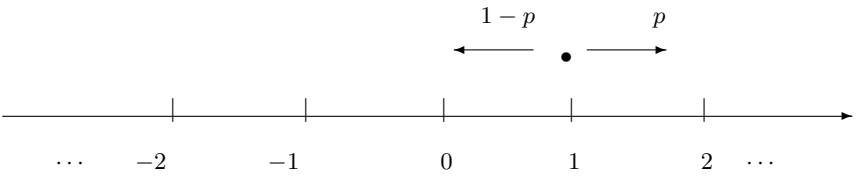

Fig. 1.1. The random walk.

$$\{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

being the state space, see Fig. 1.1. Each time the person at state i can move one step forward (+1) or one step backward (-1) with probabilities p $(0 < p < 1)$ and $(1 - p)$ respectively. Therefore we have the transition probabilities

$$P_{ji} = \begin{cases} p & \text{if } j = i + 1 \\ 1 - p & \text{if } j = i - 1 \\ 0 & \text{otherwise.} \end{cases}$$

for $i = 0, \pm 1, \pm 2, \ldots$.

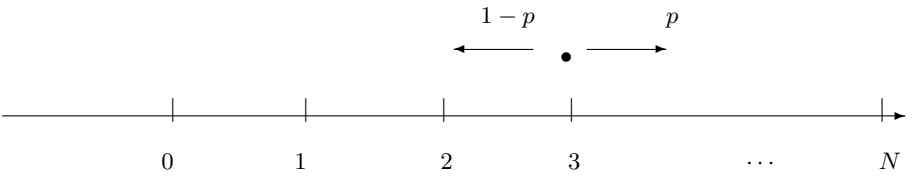

Fig. 1.2. The gambler's problem.

Example 1.10. (Gambler's Ruin) Consider a gambler gambling in a series of games, at each game, he either wins one dollar with probability p or loses one dollar with probability $(1 - p)$. The game ends if either he loses all his money or he attains a total amount of N dollars. Let the gambler's fortune be the state of the gambling process then the process is a Markov chain. Moreover, we have the transition probabilities

$$P_{ji} = \begin{cases} p & \text{if } j = i + 1 \\ 1 - p & \text{if } j = i - 1 \\ 0 & \text{otherwise.} \end{cases}$$

for $i = 1, 2, \ldots, N-1$ and $P_{00} = P_{NN} = 1$. Here state 0 and N are called the *absorbing states*. The process will stay at 0 or N forever if one of the states is reached.

1.1.2 The nth-Step Transition Matrix

In the previous section, we have defined the one-step transition probability matrix P for a Markov chain process. In this section, we are going to investigate the n-step transition probability $P_{ij}^{(n)}$ of a Markov chain process.

Definition 1.11. *Define $P_{ij}^{(n)}$ to be the probability that a process in state j will be in state i after n additional transitions. In particular $P_{ij}^{(1)} = P_{ij}$.*

Proposition 1.12. *$P^{(n)} = P^n$ where $P^{(n)}$ is the n-step transition probability matrix and P is the one-step transition matrix.*

Proof. We will prove the proposition by using mathematical induction. Clearly the proposition is true when $n = 1$. We then assume that the proposition is true for n. We note that

$$P^n = \underbrace{P \times P \times \ldots \times P}_{n \text{ times}}.$$

Then

$$P_{ij}^{(n+1)} = \sum_{k \in M} P_{ki}^{(n)} P_{jk}^{(1)} = \sum_{k \in M} P_{ki}^{n} P_{jk} = [P^{n+1}]_{ij}.$$

By the principle of mathematical induction the proposition is true for all non-negative integer n.

Remark 1.13. It is easy to see that

$$P^{(m)} P^{(n)} = P^m P^n = P^{m+n} = P^{(m+n)}.$$

Example 1.14. We consider the marketing problem again. In the model we have

$$P = \begin{pmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{pmatrix}.$$

If $\alpha = 0.3$ and $\beta = 0.4$ then we have

$$P^{(4)} = P^4 = \begin{pmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{pmatrix}^4 = \begin{pmatrix} 0.5749 & 0.5668 \\ 0.4351 & 0.4332 \end{pmatrix}.$$

Recall that a consumer is in state 0 (1) if he/she is a consumer of Wellcome (Park'n). $P_{00}^{(4)} = 0.5749$ is the probability that a Wellcome's consumer will

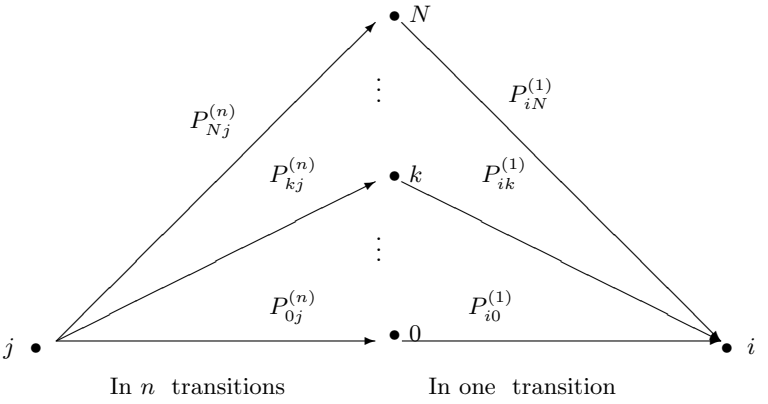

Fig. 1.3. The $(n+1)$-step transition probability.

shop with Wellcome on his/her fourth shopping and $P_{10}^{(4)} = 0.4351$ is the probability that a Wellcome's consumer will shop with Park'n on his/her fourth shopping. $P_{01}^{(4)} = 0.5668$ is the probability that a consumer of Park'n will shop with Wellcome on his/her fourth shopping. $P_{11}^{(4)} = 0.4332$ is the probability that a consumer of Park'n will shop with Park'n on his/her fourth shopping.

Remark 1.15. Consider a Markov chain process having states in $\{0, 1, 2, \ldots\}$. Suppose that we are given at time $n = 0$ the probability that the process is in state i is $a_i, i = 0, 1, 2, \ldots$. One interesting question is the following. What is the probability that the process will be in state j after n transitions? In fact, the probability that given the process is in state i and it will be in state j after n transitions is $P_{ji}^{(n)} = [P^n]_{ji}$, where P_{ji} is the one-step transition probability from state i to state j of the process. Therefore the required probability is

$$\sum_{i=0}^{\infty} P(X^{(0)} = i) \times P_{ji}^{(n)} = \sum_{i=0}^{\infty} a_i \times [P^n]_{ji}.$$

Let

$$\mathbf{X}^{(n)} = (\tilde{X}_0^{(n)}, \tilde{X}_1^{(n)}, \ldots,)$$

be the probability distribution of the states in a Markov chain process at the nth transition. Here $\tilde{X}_i^{(n)}$ is the probability that the process is in state i after n transitions and

$$\sum_{i=0}^{\infty} \tilde{X}_i^{(n)} = 1.$$

It is easy to check that

$$\mathbf{X}^{(n+1)} = P\mathbf{X}^{(n)}$$

and

$$\mathbf{X}^{(n+1)} = P^{(n+1)}\mathbf{X}^{(0)}.$$

Example 1.16. Refer to the previous example. If at $n = 0$ a consumer belongs to Park'n, we may represent this information as

$$\mathbf{X}^{(0)} = (\tilde{X}_0^{(0)}, \tilde{X}_1^{(0)})^T = (0, 1)^T.$$

What happen on his/her fourth shopping?

$$\mathbf{X}^{(4)} = P^{(4)}\mathbf{X}^{(0)} = \begin{pmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{pmatrix}^4 (0, 1)^T = (0.5668, 0.4332)^T.$$

This means that with a probability 0.4332 he/she is still a consumer of Park'n and a probability 0.5668 he/she is a consumer of Wellcome on his/her fourth shopping.

1.1.3 Irreducible Markov Chain and Classifications of States

In the following, we define two definitions for the states of a Markov chain.

Definition 1.17. *In a Markov chain, state i is said to be reachable from state j if $P_{ij}^{(n)} > 0$ for some $n \geq 0$. This means that starting from state j, it is possible (with positive probability) to enter state i in finite number of transitions.*

Definition 1.18. *State i and state j are said to communicate if state i and state j are reachable from each other.*

Remark 1.19. The definition of communication defines an equivalent relation.
(i) state i communicates with state i in 0 step because

$$P_{ii}^{(0)} = P(X^{(0)} = i | X^{(0)} = i) = 1 > 0.$$

(ii)If state i communicates with state j, then state j communicates with state i.
(iii)If state i communicates with state j and state j communicates with state k then state i communicates with state k. Since $P_{ji}^{(m)}, P_{kj}^{(n)} > 0$ for some m and n, we have

$$P_{ki}^{(m+n)} = \sum_{h \in M} P_{hi}^{(m)} P_{kh}^{(n)} \geq P_{ji}^{(m)} P_{kj}^{(n)} > 0.$$

Therefore state k is reachable from state i. By inter-changing the roles of i and k, state i is reachable from state k. Hence i communicates with k. The proof is then completed.

Definition 1.20. *Two states that communicates are said to be in the same class. A Markov chain is said to be irreducible, if all states belong to the same class, i.e. they communicate with each other.*

Example 1.21. Consider the transition probability matrix

$$\begin{matrix} 0 \\ 1 \\ 2 \end{matrix} \begin{pmatrix} 0.0 & 0.5 & 0.5 \\ 0.5 & 0.0 & 0.5 \\ 0.5 & 0.5 & 0.0 \end{pmatrix}$$

Example 1.22. Consider another transition probability matrix

$$\begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 1.0 & 0.0 & 0.5 & 0.5 \\ 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.5 & 0.5 & 0.0 \end{pmatrix}.$$

We note that from state 1, 2, 3, it is not possible to visit state 0, i.e

$$P_{01}^{(n)} = P_{02}^{(n)} = P_{03}^{(n)} = 0.$$

Therefore the Markov chain is not irreducible (or it is reducible).

Definition 1.23. *For any state i in a Markov chain, let f_i be the probability that starting in state i, the process will ever re-enter state i. State i is said to be recurrent if $f_i = 1$ and transient if $f_i < 1$.*

We have the following proposition for a recurrent state.

Proposition 1.24. *In a finite Markov chain, a state i is recurrent if and only if*

$$\sum_{n=1}^{\infty} P_{ii}^{(n)} = \infty.$$

By using Proposition (1.24) one can prove the following proposition.

Proposition 1.25. *In a finite Markov chain, if state i is recurrent (transient) and state i communicates with state j then state j is also recurrent (transient).*

1.1.4 An Analysis of the Random Walk

Recall the classical example of random walk, the analysis of the random walk can also be found in Ross [180]. A person performs a random walk on the real line of integers. Each time the person at state i can move one step forward (+1) or one step backward (-1) with probabilities p $(0 < p < 1)$ and $(1 - p)$ respectively. Since all the states are communicated, by Proposition 1.25, all states are either recurrent or they are all transient.

Let us consider state 0. To classify this state one can consider the following sum:

$$\sum_{m=1}^{\infty} P_{00}^{(m)}.$$

We note that

$$P_{00}^{(2n+1)} = 0$$

because in order to return to state 0, the number of forward movements should be equal to the number of backward movements and therefore the number of movements should be even and

$$P_{00}^{(2n)} = \binom{2n}{n} p^n (1-p)^n.$$

Hence we have

$$I = \sum_{m=1}^{\infty} P_{00}^{(m)} = \sum_{n=1}^{\infty} P_{00}^{(2n)} = \sum_{n=1}^{\infty} \binom{2n}{n} p^n (1-p)^n = \sum_{n=1}^{\infty} \frac{(2n)!}{n!n!} p^n (1-p)^n.$$

Recall that if I is finite then state 0 is transient otherwise it is recurrent. Then we can apply the Stirling's formula to get a conclusive result. The Stirling's formula states that if n is large then

$$n! \approx n^{n+\frac{1}{2}} e^{-n} \sqrt{2\pi}.$$

Hence one can approximate

$$P_{00}^{(2n)} \approx \frac{(4p(1-p))^n}{\sqrt{\pi n}}.$$

There are two cases to consider. If $p = \frac{1}{2}$ then we have

$$P_{00}^{(2n)} \approx \frac{1}{\sqrt{\pi n}}.$$

If $p \neq \frac{1}{2}$ then we have

$$P_{00}^{(2n)} \approx \frac{a^n}{\sqrt{\pi n}}$$

where

$$0 < a = 4p(1-p) < 1.$$

Therefore when $p = \frac{1}{2}$, state 0 is recurrent as the sum is infinite, and when $p \neq \frac{1}{2}$, state 0 is transient as the sum is finite.

1.1.5 Simulation of Markov Chains with EXCEL

Consider a Markov chain process with three states $\{0, 1, 2\}$ with the transition probability matrix as follows:

$$P = \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.3 & 0.1 & 0.3 \\ 0.5 & 0.4 & 0.4 \end{pmatrix}.$$

Given that $X_0 = 0$, our objective here is to generate a sequence

$$\{X^{(n)}, n = 1, 2, \ldots\}$$

which follows a Markov chain process with the transition matrix P.

To generate $\{X^{(n)}\}$ there are three possible cases:

(i) Suppose $X^{(n)} = 0$, then we have

$$P(X^{(n+1)} = 0) = 0.2 \qquad P(X^{(n+1)} = 1) = 0.3 \qquad P(X^{(n+1)} = 2) = 0.5;$$

(ii) Suppose $X^{(n)} = 1$, then we have

$$P(X^{(n+1)} = 0) = 0.5 \qquad P(X^{(n+1)} = 1) = 0.1 \qquad P(X^{(n+1)} = 2) = 0.4;$$

(iii) Suppose $X^{(n)} = 2$, then we have

$$P(X^{(n+1)} = 0) = 0.3 \qquad P(X^{(n+1)} = 1) = 0.3 \qquad P(X^{(n+1)} = 2) = 0.4.$$

Suppose we can generate a random variable U which is uniformly distributed over $[0, 1]$. Then one can generate the distribution in Case (i) when $X^{(n)} = 0$ easily as follows:

$$X^{(n+1)} = \begin{cases} 0 & \text{if} \quad U \in [0, 0.2), \\ 1 & \text{if} \quad U \in [0.2, 0.5), \\ 2 & \text{if} \quad U \in [0.5, 1]. \end{cases}$$

The distribution in Case (ii) when $X^{(n)} = 1$ can be generated as follows:

$$X^{(n+1)} = \begin{cases} 0 & \text{if} \quad U \in [0, 0.5), \\ 1 & \text{if} \quad U \in [0.5, 0.6), \\ 2 & \text{if} \quad U \in [0.6, 1]. \end{cases}$$

The distribution in Case (iii) when $X^{(n)} = 2$ can be generated as follows:

$$X^{(n+1)} = \begin{cases} 0 & \text{if} \quad U \in [0, 0.3), \\ 1 & \text{if} \quad U \in [0.3, 0.6), \\ 2 & \text{if} \quad U \in [0.6, 1]. \end{cases}$$

In EXCEL one can generate U, a random variable uniformly distributed over $[0, 1]$ by using "=rand()". By using simple logic statement in EXCEL, one can

simulate a Markov chain easily. The followings are some useful logic statements in EXCEL used in the demonstration file.

(i) "B1" means column B and Row 1.
(ii) "=IF(B1=0,1,-1)" gives 1 if B1=0 otherwise it gives -1.
(iii) "=IF(A1 > B2,0,1)" gives 0 if A1 > B2 otherwise it gives 1.
(iv) "=IF(AND(A1=1,B2>2),1,0)" gives 1 if A1=1 and B2>2 otherwise it gives 0.
(v) "=max(1,2,-1) =2 " gives the maximum of the numbers.

A demonstration EXCEL file is available at [221] for reference. The program generates a Markov chain process

$$X^{(1)}, X^{(2)}, \ldots, X^{(30)}$$

whose transition probability is P and $X^{(0)} = 0$.

1.1.6 Building a Markov Chain Model

Given an observed data sequence $\{X^{(n)}\}$, one can find the transition frequency F_{jk} in the sequence by counting the number of transitions from state j to state k in one step. Then one can construct the one-step transition matrix for the sequence $\{X^{(n)}\}$ as follows:

$$F = \begin{pmatrix} F_{11} & \cdots\cdots & F_{1m} \\ F_{21} & \cdots\cdots & F_{2m} \\ \vdots & \vdots \quad \vdots & \vdots \\ F_{m1} & \cdots\cdots & F_{mm} \end{pmatrix}. \tag{1.1}$$

From F, one can get the estimates for P_{jk} as follows:

$$P = \begin{pmatrix} P_{11} & \cdots\cdots & P_{1m} \\ P_{21} & \cdots\cdots & P_{2m} \\ \vdots & \vdots \quad \vdots & \vdots \\ P_{m1} & \cdots\cdots & P_{mm} \end{pmatrix} \tag{1.2}$$

where

$$P_{jk} = \begin{cases} \dfrac{F_{jk}}{\sum_{j=1}^{m} F_{jk}} & \text{if} \quad \sum_{j=1}^{m} F_{jk} > 0 \\ 0 \text{ if } \sum_{j=1}^{m} F_{jk} = 0. \end{cases}$$

We consider a sequence $\{X^{(n)}\}$ of three states ($m = 3$) given by

``U" is a column of random numbers in (0,1). Column E (J) [O] gives the the next state given that the current state is 0 (1) [2]. Column P gives the simulated sequence X(t) given that X(0)=0.

U	0	1	2	X(t+1)|X(t)=0	U	0	1	2	X(t+1)|X(t)=1	U	0	1	2	X(t+1)|X(t)=2	X(t)
0.55	-1	-1	2	**2**	0.065	-1	1	-1	**1**	0.82	-1	1	-1	**2**	**2**
0.74	-1	-1	2	**2**	0.523	-1	-1	2	**2**	0.96	-1	-1	2	**1**	**1**
0.72	-1	-1	2	**2**	0.55	-1	-1	2	**2**	0.18	-1	-1	2	**2**	**2**
1	-1	-1	2	**2**	0.34	-1	-1	2	**2**	0.42	-1	-1	2	**2**	**2**
0.96	-1	-1	2	**2**	0.92	-1	-1	2	**2**	0.91	-1	-1	2	**2**	**2**
0.25	-1	1	-1	**1**	0.593	0	-1	-1	**0**	0.05	0	-1	-1	**2**	**2**
0.83	-1	-1	2	**2**	0.377	-1	-1	2	**2**	0.74	-1	-1	2	**0**	**0**
0.97	-1	-1	2	**2**	0.09	-1	-1	2	**2**	0.41	-1	-1	2	**2**	**2**
0.91	-1	-1	2	**2**	0.682	-1	-1	2	**2**	0.38	-1	-1	2	**2**	**2**
0.5	-1	-1	2	**2**	0.198	-1	1	-1	**1**	0.68	-1	1	-1	**2**	**2**
0.26	-1	1	-1	**1**	0.52	0	-1	-1	**0**	0.61	0	-1	-1	**1**	**1**
0.76	-1	-1	2	**2**	0.884	-1	-1	2	**2**	0.13	-1	-1	2	**0**	**2**
0.35	-1	1	-1	**1**	0.769	0	-1	-1	**0**	0.55	-1	1	-1	**2**	**2**
0.92	-1	-1	2	**2**	0.286	-1	-1	2	**2**	0.98	-1	-1	2	**1**	**1**
0.57	-1	-1	2	**2**	0.436	-1	1	-1	**1**	0.27	-1	1	-1	**2**	**1**
0.11	0	-1	-1	**0**	0.421	0	-1	-1	**0**	0.45	0	-1	-1	**1**	**0**
0.85	-1	-1	2	**2**	0.938	-1	-1	2	**2**	0.07	-1	-1	2	**0**	**2**
0.11	0	-1	-1	**0**	0.695	0	-1	-1	**0**	0.08	0	-1	-1	**2**	**2**
0.06	0	-1	-1	**0**	0.622	0	-1	-1	**0**	0.18	0	-1	-1	**0**	**0**
0.21	-1	1	-1	**1**	0.44	0	-1	-1	**0**	0.87	0	-1	-1	**0**	**1**
0.58	-1	-1	2	**2**	0.081	-1	1	-1	**1**	0.52	-1	1	-1	**0**	**1**
0.82	-1	-1	2	**2**	0.358	-1	-1	2	**2**	0.49	-1	-1	2	**1**	**2**
0.98	-1	-1	2	**2**	0.685	-1	-1	2	**2**	0.24	-1	-1	2	**2**	**2**
0.8	-1	-1	2	**2**	0.691	-1	-1	2	**2**	0.11	-1	-1	2	**2**	**2**
0.81	-1	-1	2	**2**	0.138	-1	-1	2	**2**	0.99	-1	-1	2	**2**	**2**
0.52	-1	-1	2	**2**	0.1	-1	1	-1	**1**	0.61	-1	1	-1	**2**	**2**
0.16	0	-1	-1	**0**	0.713	0	-1	-1	**0**	0.97	0	-1	-1	**1**	**1**
0.22	-1	1	-1	**1**	0.54	0	-1	-1	**0**	0.48	0	-1	-1	**0**	**0**
0.19	0	-1	-1	**0**	0.397	0	-1	-1	**0**	0.18	0	-1	-1	**0**	**0**
0.64	-1	-1	2	**2**	0.673	-1	-1	2	**2**	0.09	-1	-1	2	**0**	**2**

Fig. 1.4. Simulation of a Markov chain.

$$\{0, 0, 1, 1, 0, 2, 1, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 0, 1\}. \tag{1.3}$$

We have the transition frequency matrix

$$F = \begin{pmatrix} 1 & 3 & 3 \\ 6 & 1 & 1 \\ 1 & 3 & 0 \end{pmatrix}. \tag{1.4}$$

Therefore one-step transition matrices can be estimated as follows:

$$P = \begin{pmatrix} 1/8 & 3/7 & 3/4 \\ 3/4 & 1/7 & 1/4 \\ 1/8 & 3/7 & 0 \end{pmatrix}. \tag{1.5}$$

A demonstration EXCEL file is available at [222] for reference.

X(t)	P00	P01	P02	P10	P11	P12	P20	P21	P22
0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0
1	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	0	0	1	0
1	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0
2	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0
F(ij)	1	6	1	4	1	3	3	1	0
P(ij)	0.125	0.75	0.125	0.5	0.125	0.375	0.75	0.25	0

Fig. 1.5. Building a Markov chain.

1.1.7 Stationary Distribution of a Finite Markov Chain

Definition 1.26. *A state i is said to have period d if $P_{ii}^{(n)} = 0$ whenever n is not divisible by d, and d is the largest integer with this property. A state with period 1 is said to be aperiodic.*

Example 1.27. Consider the transition probability matrix

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

We note that

$$P^{(n)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^n = \frac{1}{2} \begin{pmatrix} 1+(-1)^n & 1+(-1)^{n+1} \\ 1+(-1)^{n+1} & 1+(-1)^n \end{pmatrix}.$$

We note that $P_{00}^{(2n+1)} = P_{11}^{(2n+1)} = 0$, so both States 0 and 1 have a period of 2.

Definition 1.28. *State i is said to be positive recurrent if it is recurrent and starting in state i the expected time until the process returns to state i is finite.*

Definition 1.29. *A state is said to be egordic if it is positive recurrent and aperiodic.*

We recall the example of the marketing problem with $\mathbf{X}^{(0)} = (1,0)^t$. We observe that

$$\mathbf{X}^{(1)} = P\mathbf{X}^{(0)} = \begin{pmatrix} 0.7 & 0.4 \\ 0.3 & 0.6 \end{pmatrix} (1,0)^T = (0.7, 0.3)^T,$$

$$\mathbf{X}^{(2)} = P^2\mathbf{X}^{(0)} = \begin{pmatrix} 0.61 & 0.52 \\ 0.39 & 0.48 \end{pmatrix} (1,0)^T = (0.61, 0.39)^T,$$

$$\mathbf{X}^{(4)} = P^4\mathbf{X}^{(0)} = \begin{pmatrix} 0.5749 & 0.5668 \\ 0.4251 & 0.4332 \end{pmatrix} (1,0)^T = (0.5749, 0.4251)^T,$$

$$\mathbf{X}^{(8)} = P^8\mathbf{X}^{(0)} = \begin{pmatrix} 0.5715 & 0.5714 \\ 0.4285 & 0.4286 \end{pmatrix} (1,0)^T = (0.5715, 0.4285)^T,$$

$$\mathbf{X}^{(16)} = P^{16}\mathbf{X}^{(0)} = \begin{pmatrix} 0.5714 & 0.5174 \\ 0.4286 & 0.4286 \end{pmatrix} (1,0)^T = (0.5714, 0.4286)^T.$$

It seems that

$$\lim_{n\to\infty} \mathbf{X}^{(n)} = (0.57\ldots, 0.42\ldots)^T.$$

In fact this limit exists and is independent of $\mathbf{X}^{(0)}$! It means that in the long run, the probability that a consumer belongs to Wellcome (Park'n) is given by 0.57 (0.42).

We note that $\mathbf{X}^{(n)} = P\mathbf{X}^{(n-1)}$ therefore if we let

$$\lim_{n\to\infty} \mathbf{X}^{(n)} = \pi$$

then

$$\pi = \lim_{n\to\infty} \mathbf{X}^{(n)} = \lim_{n\to\infty} P\mathbf{X}^{(n-1)} = P\pi.$$

We have the following definition

Definition 1.30. *A vector*

$$\pi = (\pi_0, \pi_1, \ldots, \pi_{k-1})^t$$

is said to be a stationary distribution of a finite Markov chain if it satisfies:

(i)

$$\pi_i \geq 0 \quad \text{and} \quad \sum_{i=0}^{k-1} \pi_i = 1.$$

(ii)

$$P\pi = \pi, \quad \text{i.e.} \sum_{j=0}^{k-1} P_{ij}\pi_j = \pi_i.$$

Proposition 1.31. *For any irreducible and aperiodic Markov chain having k states, there exists at least one stationary distribution.*

Proposition 1.32. *For any irreducible and aperiodic Markov chain having k states, for any initial distribution* $\mathbf{X}^{(0)}$

$$\lim_{n\to\infty} ||\mathbf{X}^{(n)} - \pi|| = \lim_{n\to\infty} ||P^n\mathbf{X}^{(0)} - \pi|| = 0.$$

where π *is a stationary distribution for the transition matrix* P*.*

Proposition 1.33. *The stationary distribution* π *in Proposition 1.32 is unique.*

There are a number of popular vector norms $||.||$. In the following, we introduce three of them.

Definition 1.34. *The* $\mathbf{v}$ *be a vector in* R^n*, then we have* L_1*-norm,* L_∞*-norm and* 2*-norm defined respectively by*

$$||\mathbf{v}||_1 = \sum_{i=1}^{n} |v_i|,$$

$$||\mathbf{v}||_\infty = \max_i\{|v_i|\},$$

and

$$||\mathbf{v}||_2 = \sqrt{\sum_{i=1}^{n} |v_i|^2}.$$

1.1.8 Applications of the Stationary Distribution

Recall the marketing problem again. The transition matrix is given by

$$P = \begin{pmatrix} 1-\alpha & \beta \\ \alpha & 1-\beta \end{pmatrix}.$$

To solve for the stationary distribution (π_0, π_1), we consider the following linear system of equations

$$\begin{cases} (1-\alpha)\pi_0 + \beta\pi_1 & = \pi_0 \\ \alpha\pi_0 + (1-\beta)\pi_1 & = \pi_1 \\ \pi_0 + \pi_1 & = 1. \end{cases}$$

Solving the linear system of equations we have

$$\begin{cases} \pi_0 = \beta(\alpha+\beta)^{-1} \\ \pi_1 = \alpha(\alpha+\beta)^{-1}. \end{cases}$$

Therefore in the long run, the market shares of Wellcome and Park'n are respectively

$$\frac{\beta}{(\alpha+\beta)} \quad \text{and} \quad \frac{\alpha}{(\alpha+\beta)}.$$

1.2 Continuous Time Markov Chain Process

In the previous section, we have discussed discrete time Markov chain processes. In many situations, a change of state does not occur at a fixed discrete time. In fact, the duration of a system state can be a continuous random variable. In our context, we are going to model queueing systems and remanufacturing systems by continuous time Markov process. Here we first give the definition for a Poisson process. We then give some important properties of the Poisson process.

A process is called a Poisson process if

(A1) the probability of occurrence of one event in the time interval $(t, t+\delta t)$ is $\lambda\delta t + o(\delta t)$. Here λ is a positive constant and $o(\delta t)$ is such that

$$\lim_{\delta t \to 0} \frac{o(\delta t)}{\delta t} = 0.$$

(A2) the probability of occurrence of no event in the time interval $(t, t+\delta t)$ is $1 - \lambda\delta t + o(\delta t)$.

(A3) the probability of occurrences of more than one event is $o(\delta t)$.

Here an "event" can be an arrival of a bus or a departure of customer. From the above assumptions, one can derive the well-known Poisson distribution.

We define $P_n(t)$ be the probability that n events occurred in the time interval $[0, t]$. Assuming that that $P_n(t)$ is differentiable, then we can get a relationship between $P_n(t)$ and $P_{n-1}(t)$ as follows:

$$P_n(t+\delta t) = P_n(t) \cdot (1 - \lambda\delta t - o(\delta t)) + P_{n-1}(t) \cdot (\lambda\delta t + o(\delta t)) + o(\delta t).$$

Rearranging the terms we get

$$\frac{P_n(t+\delta t) - P_n(t)}{\delta t} = -\lambda P_n(t) + \lambda P_{n-1}(t) + (P_{n-1}(t) + P_n(t))\frac{o(\delta t)}{\delta t}.$$

Let δt goes to zero, we have

$$\lim_{\delta t \to 0} \frac{P_n(t+\delta t) - P_n(t)}{\delta t} = -\lambda P_n(t) + \lambda P_{n-1}(t) + \lim_{\delta t \to 0}(P_{n-1}(t) + P_n(t))\frac{o(\delta t)}{\delta t}.$$

Hence we have the differential-difference equation:

$$\frac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t) + 0, \quad n = 0, 1, 2, \ldots.$$

Since $P_{-1}(t) = 0$, we have the *initial value problem* for $P_0(t)$ as follows:

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) \quad \text{with} \quad P_0(0) = 1.$$

The probability $P_0(0)$ is the probability that no event occurred in the time interval $[0, 0]$, so it must be one. Solving the separable ordinary differential equation for $P_0(t)$ we get

$$P_0(t) = e^{-\lambda t}$$

which is the probability that no event occurred in the time interval $[0, t]$. Thus

$$1 - P_0(t) = 1 - e^{-\lambda t}$$

is the probability that at least one event occurred in the time interval $[0, t]$. Therefore the probability density function $f(t)$ for the waiting time of the first event to occur is given by the well-known exponential distribution

$$f(t) = \frac{d(1 - e^{-\lambda t})}{dt} = \lambda e^{-\lambda t}, \quad t \geq 0.$$

We note that

$$\begin{cases} \dfrac{dP_n(t)}{dt} = -\lambda P_n(t) + \lambda P_{n-1}(t), \quad n = 1, 2, \ldots \\ P_0(t) = e^{-\lambda t}, \\ P_n(0) = 0 \quad n = 1, 2, \ldots. \end{cases}$$

Solving the above differential-difference equations, we get

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}.$$

Finally, we present the important relationships among the Poisson process, Poisson distribution and the exponential distribution [52].

Proposition 1.35. *The following statements (B1),(B2), and (B3) are equivalent.*
(B1) The arrival process is a Poisson process with mean rate λ.
(B2) Let $N(t)$ be the number of arrivals in the time interval $[0, t]$ then

$$P(N(t) = n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \quad n = 0, 1, 2, \ldots.$$

(B3) The inter-arrival time follows the exponential distribution with mean λ^{-1}.

1.2.1 A Continuous Two-state Markov Chain

Consider a one-server queueing system which has two possible states: 0 (idle) and 1 (busy). Assuming that the arrival process of the customers is a Poisson process with mean rate λ and the service time of the server follows the exponential distribution with mean rate μ. Let $P_0(t)$ be the probability that the server is idle at time t and $P_1(t)$ be the probability that the server is busy at time t. Using a similar argument as in the derivation of a Poisson process, we have

$$\begin{cases} P_0(t+\delta t) = (1 - \lambda\delta t - o(\delta t))P_0(t) + (\mu\delta t + o(\delta t))P_1(t) + o(\delta t) \\ P_1(t+\delta t) = (1 - \mu\delta t - o(\delta t))P_1(t) + (\lambda\delta t + o(\delta t))P_0(t) + o(\delta t). \end{cases}$$

Rearranging the terms, one gets

$$\begin{cases} \dfrac{P_0(t+\delta t) - P_0(t)}{\delta t} = -\lambda P_0(t) + \mu P_1(t) + (P_1(t) - P_0(t))\dfrac{o(\delta t)}{\delta t} \\ \dfrac{P_1(t+\delta t) - P_1(t)}{\delta t} = \lambda P_0(t) - \mu P_1(t) + (P_0(t) - P_1(t))\dfrac{o(\delta t)}{\delta t}. \end{cases}$$

Letting δt goes to zero, we get

$$\begin{cases} \dfrac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t) \\ \dfrac{dP_1(t)}{dt} = \lambda P_0(t) - \mu P_1(t). \end{cases}$$

Solving the above differential equations, we have

$$P_1(t) = \frac{1}{\lambda + \mu}(\mu e^{-(\lambda+\mu)t} + \lambda)$$

and

$$P_0(t) = \frac{1}{\lambda+\mu}(\mu - \mu e^{-(\lambda+\mu)t}).$$

We note that the steady state probabilities are given by

$$\lim_{t\to\infty} P_0(t) = \frac{\mu}{\lambda+\mu}$$

and

$$\lim_{t\to\infty} P_1(t) = \frac{\lambda}{\lambda+\mu}.$$

In fact, the steady state probability distribution can be obtained without solving the differential equations. We write the system of differential equations in matrix form:

$$\begin{pmatrix} \frac{dP_0(t)}{dt} \\ \frac{dP_1(t)}{dt} \end{pmatrix} = \begin{pmatrix} -\lambda & \mu \\ \lambda & -\mu \end{pmatrix} \begin{pmatrix} P_0(t) \\ P_1(t) \end{pmatrix}.$$

Since in steady state, $P_0(t) = p_0$ and $P_1(t) = p_1$ are constants and independent of t, we have

$$\frac{dp_0(t)}{dt} = \frac{dp_1(t)}{dt} = 0.$$

The steady state probabilities will be the solution of the following linear system:

$$\begin{pmatrix} -\lambda & \mu \\ \lambda & -\mu \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

subject to $p_0 + p_1 = 1$.

In fact, very often we are interested in obtaining the steady state probability distribution of the Markov chain. Because a lot of system performance such as expected number of customers, average waiting time can be written in terms of the steady state probability distribution, see for instance [48, 49, 50, 52]. We will also apply the concept of steady state probability distribution in the upcoming chapters. When the number of states is large, solving the steady state probability distribution will be time consuming. Iterative methods are popular approaches for solving large scale Markov chain problem.

1.3 Iterative Methods for Solving Linear Systems

In this section, we introduce some classical iterative methods for solving large linear systems. For more detail introduction to iterative methods, we refer reader to books by Bini et al. [21], Kincaid and Cheney [130], Golub and van Loan [101] and Saad [181].

1.3.1 Some Results on Matrix Theory

We begin our discussion by some more useful results in matrix theory and their proofs can be found in [112, 101, 130]. The first results is a useful formula for solving linear systems.

Proposition 1.36. *(Sherman-Morrison-Woodbury Formula) Let M be an non-singular $n \times n$ matrix, $\mathbf{u}$ and $\mathbf{v}$ be two $n \times k$ $(l \leq n)$ matrices such that the matrix $(I_l + \mathbf{v}^T M \mathbf{u})$ is non-singular. Then we have*

$$\left(M + \mathbf{u}\mathbf{v}^T\right)^{-1} = M^{-1} - M^{-1}\mathbf{u}\left(I_l + \mathbf{v}^T M^{-1}\mathbf{u}\right)^{-1}\mathbf{v}^T M^{-1}.$$

The second result is on the eigenvalue of non-negative and irreducible square matrix.

Proposition 1.37. *(Perron-Frobenius Theorem) Let A be a non-negative and irreducible square matrix of order m. Then we have*
(i) A has a positive real eigenvalue λ which is equal to its spectral radius, i.e., $\lambda = \max_k |\lambda_k(A)|$ where $\lambda_k(A)$ denotes the k-th eigenvalue of A.
(ii) There corresponds an eigenvector $\mathbf{z}$ with all its entries being real and positive, such that $A\mathbf{z} = \lambda\mathbf{z}$.
(iii) λ is a simple eigenvalue of A.

The last result is on matrix norms. There are many matrix norms $||.||_M$ one can use. In the following, we introduce the definition of a matrix norm $||.||_{M_V}$ induced by a vector norm $||.||_V$.

Definition 1.38. *Given a vector $||.||_V$ in R^n, the matrix norm $||A||_{M_V}$ for an $n \times n$ matrix A induced by the vector norm is defined as*

$$||A||_{M_V} = \sup\{||A\mathbf{x}||_V : \mathbf{x} \in R^n \text{ and } ||\mathbf{x}||_V = 1\}.$$

In the following proposition, we introduce three popular matrix norms.

Proposition 1.39. *Let A be an $n \times n$ real matrix, then it can be shown that the matrix 1-norm, matrix ∞-norm and matrix 2-norm induced by $||.||_1$, $||.||_\infty$ and $||.||_2$ respectively by*

$$||A||_1 = \max_j\{\sum_{i=1}^{n} |A_{ij}|\},$$

$$||A||_\infty = \max_i\{\sum_{j=1}^{n} |A_{ij}|\},$$

and

$$||A||_2 = \sqrt{\lambda_{max}(AA^T)}.$$

Another other popular matrix norm is the Frobenius norm.

Definition 1.40. *The Frobenius norm of a square matrix A is defined as*

$$||A||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^2}.$$

1.3.2 Splitting of a Matrix

We begin with the concept of splitting a matrix. If we are to solve

$$A\mathbf{x} = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} & 0 \\ \frac{1}{3} & 1 & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 10 \\ 5 \end{pmatrix} = \mathbf{b}.$$

There are many ways to split the matrix A into two parts and develop iterative methods for solving the linear system.

There are at least three different ways of splitting the matrix A:

$$\begin{aligned} A &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} \frac{-1}{2} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & -\frac{1}{2} \end{pmatrix} \quad \text{(case 1)} \\ &= \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & 0 \end{pmatrix} \quad \text{(case 2)} \\ &= \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ 0 & \frac{1}{3} & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 \end{pmatrix} \quad \text{(case 3)} \\ &= S + (A - S) \end{aligned}$$

Now

$$A\mathbf{x} = (S + (A - S))\mathbf{x} = \mathbf{b}$$

and therefore

$$S\mathbf{x} + (A - S)\mathbf{x} = \mathbf{b}$$

Hence we may write

$$\mathbf{x} = S^{-1}\mathbf{b} - S^{-1}(A - S)\mathbf{x}$$

where we assume that S^{-1} exists. Then given an initial guess $\mathbf{x}^{(0)}$ of the solution of $A\mathbf{x} = \mathbf{b}$, one may consider the following iterative scheme:

$$\mathbf{x}^{(k+1)} = S^{-1}\mathbf{b} - S^{-1}(A - S)\mathbf{x}^{(k)}. \tag{1.6}$$

Clearly if $\mathbf{x}^{(k)} \to \mathbf{x}$ as $k \to \infty$ then we have $\mathbf{x} = A^{-1}\mathbf{b}$. We note that (1.6) converges if and only if there is a matrix norm $||.||_M$ such that

$$||S^{-1}(A - S)||_M < 1.$$

This is because for any square matrix B, we have

$$(I - B)(I + B + B^2 + \ldots + B^n) = I - B^{n+1}$$

and

$$\sum_{k=0}^{\infty} B^k = (I - B)^{-1} \quad \text{if} \quad \lim_{n\to\infty} B^n = 0.$$

If there exists a matrix norm $||.|_M$ such that $||B||_M < 1$ then

$$\lim_{n\to\infty} ||B^n||_M \leq \lim_{n\to\infty} ||B||_M^n = 0$$

and we have

$$\lim_{n\to\infty} B^n = 0.$$

Therefore we have the following proposition.

Proposition 1.41. *If*

$$\|S^{-1}(A - S)\|_M < 1$$

then the iterative scheme converges to the solution of $A\mathbf{x} = \mathbf{b}$.

1.3.3 Classical Iterative Methods

Throughout this section, we let A be the matrix to be split and $\mathbf{b}$ be the right hand side vector. We use $\mathbf{x}^{(0)} = (0, 0, 0)^T$ as the initial guess.

Case 1: $S = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$.

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{b} - (A - I)\mathbf{x}^{(k)} \\ &= \begin{pmatrix} 5 \\ 10 \\ 5 \end{pmatrix} - \begin{pmatrix} -\frac{1}{2} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & -\frac{1}{2} \end{pmatrix} \mathbf{x}^{(k)} \end{aligned}$$

$$\begin{aligned} \mathbf{x}^{(1)} &= (5 \;\; 10 \;\; 5)^T \\ \mathbf{x}^{(2)} &= (4.1667 \quad 6.6667 \quad 4.1667)^T \\ \mathbf{x}^{(3)} &= (4.8611 \quad 7.2222 \quad 4.8611)^T \\ \mathbf{x}^{(4)} &= (5.0231 \quad 6.7593 \quad 5.0231)^T \\ &\vdots \\ \mathbf{x}^{(30)} &= (5.9983 \quad 6.0014 \quad 5.9983)^T. \end{aligned}$$

When $S = I$, this is called the *Richardson method*.

Case 2: $S = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$

Therefore

$$\begin{aligned} \mathbf{x}^{(k+1)} &= S^{-1}\mathbf{b} - S^{-1}(A-S)\mathbf{x}^{(k)} \\ &= \begin{pmatrix} 10 \\ 10 \\ 10 \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}^{-1} \begin{pmatrix} 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & 0 \end{pmatrix} \mathbf{x}^{(k)} \\ &= (10\ 10\ 10)^T - \begin{pmatrix} 0 & \frac{2}{3} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{2}{3} & 0 \end{pmatrix} \mathbf{x}^{(k)} \end{aligned}$$

$$\begin{aligned} \mathbf{x}^{(1)} &= (10\ 10\ 10)^T \\ \mathbf{x}^{(2)} &= (3.3333 \quad 3.3333 \quad 3.3333)^T \\ \mathbf{x}^{(3)} &= (7.7778 \quad 7.7778 \quad 7.7778)^T \\ &\vdots \\ \mathbf{x}^{(30)} &= (6.0000 \quad 6.0000 \quad 6.0000)^T. \end{aligned}$$

When $S = \text{Diag}(a_{11}, \cdots, a_{nn})$. This is called the *Jacobi method*.

Case 3: $S = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ 0 & \frac{1}{3} & \frac{1}{2} \end{pmatrix}$

$$\begin{aligned} \mathbf{x}^{(k+1)} &= S^{-1}\mathbf{b} - S^{-1}(A-S)\mathbf{x}^{(k)} \\ &= \begin{pmatrix} 10 \\ \frac{20}{3} \\ \frac{50}{9} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ 0 & \frac{1}{3} & \frac{1}{2} \end{pmatrix}^{-1} \begin{pmatrix} 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 \end{pmatrix} \mathbf{x}^{(k)} \end{aligned}$$

$$\begin{aligned} \mathbf{x}^{(1)} &= (10\ \frac{20}{3}\ \frac{50}{9})^T \\ \mathbf{x}^{(2)} &= (5.5556 \quad 6.2963 \quad 5.8025)^T \\ \mathbf{x}^{(3)} &= (5.8025 \quad 6.1317 \quad 5.9122)^T \\ \mathbf{x}^{(4)} &= (5.9122 \quad 6.0585 \quad 5.9610)^T \\ &\vdots \\ \mathbf{x}^{(14)} &= (6.0000 \quad 6.0000 \quad 6.0000)^T. \end{aligned}$$

When S is the lower triangular part of the matrix A. This method is called the *Gauss-Seidel method*.

Proposition 1.42. *If A is diagonally dominant then*

$$||D^{-1}(A-D)||_\infty < 1$$

and the Jacobi method converges to the solution of $A\mathbf{x} = \mathbf{b}$.

1.3.4 Spectral Radius

Definition 1.43. *Given an $n \times n$ square matrix A the spectral radius of A is defined as*

$$\rho(A) = \max\{|\lambda| : \det(A - \lambda I) = 0\}$$

or in other words if $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the eigenvalues of A then

$$\rho(A) = \max_i\{|\lambda_i|\}.$$

Example 1.44.

$$A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

then the eigenvalues of A are $\pm i$ and $|i| = |-i| = 1$. Therefore $\rho(A) = 1$ in this case.

Proposition 1.45. *For any square matrix A, $\rho(A) = \inf_{\|\cdot\|_M} \|A\|_M$.*

Remark 1.46. If $\rho(A) < 1$ then there exists a matrix norm $||.||_M$ such that $||A||_M < 1$.

Using the remark, one can show the following proposition.

Proposition 1.47. *The iterative scheme*

$$x^{(k)} = G\mathbf{x}^{(k-1)} + \mathbf{c}$$

converges to

$$(I - G)^{-1}\mathbf{c}$$

for any starting vectors $\mathbf{x}^{(0)}$ and $\mathbf{c}$ if and only if $\rho(G) < 1$.

Proposition 1.48. *The iterative scheme*

$$\mathbf{x}^{(k+1)} = S^{-1}\mathbf{b} - S^{-1}(A - S)\mathbf{x}^{(k)} = (I - S^{-1}A)\mathbf{x}^{(k)} + S^{-1}\mathbf{b}$$

converges to $A^{-1}\mathbf{b}$ if and only if $\rho(I - S^{-1}A) < 1$.

Proof. Take $G = I - S^{-1}A$ and $\mathbf{c} = S^{-1}\mathbf{b}$.

Definition 1.49. *An $n \times n$ matrix B is said to be strictly diagonal dominant if*

$$|B_{ii}| > \sum_{j=1, j\neq i}^{n} |B_{ij}| \quad \text{for} \quad i = 1, 2, \ldots, n$$

Proposition 1.50. *If A is strictly diagonally dominant then the Gauss-Seidel method converges for any starting $\mathbf{x}^{(0)}$.*

Proof. Let S be the lower triangular part of A. From Proposition 1.48 above, we only need to show

$$\rho(I - S^{-1}A) < 1.$$

Let λ be an eigenvalue of $(I - S^{-1}A)$ and $\mathbf{x}$ be its corresponding eigenvector such that

$$\|\mathbf{x}\|_\infty = 1.$$

We want to show

$$|\lambda| < 1.$$

We note that

$$(I - S^{-1}A)\mathbf{x} = \lambda\mathbf{x}$$

and therefore

$$\begin{pmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & 0 & & \\ \vdots & & \ddots & -a_{n-1n} \\ 0 & \cdots & & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ a_{n1} & \cdots & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_n \end{pmatrix}.$$

Therefore we have

$$-\sum_{j=i+1}^{n} a_{ij}x_j = \lambda \sum_{j=1}^{i} a_{ij}x_j \quad \text{for } i = 1, \cdots, n-1.$$

Since $\|x\|_\infty = 1$, there exists i such that

$$|x_i| = 1 \geq |x_j|.$$

For this i we have

$$|\lambda||a_{ii}| = |\lambda a_{ii}x_i| \leq \sum_{j=i+1}^{n} |a_{ij}| + |\lambda| \sum_{j=1}^{i-1} |a_{ij}|$$

and therefore

$$|\lambda| \leq \sum_{j=i+1}^{n} |a_{ij}| \Bigg/ \left(|a_{ii}| - \sum_{j=1}^{i-1} |a_{ij}| \right) < 1$$

1.3.5 Successive Over-Relaxation (SOR) Method

In solving $A\mathbf{x} = \mathbf{b}$, one may split A as follows:

$$A = \underbrace{L + wD}+(1-w)D + U$$

where L is the strictly lower triangular part; D is the diagonal part and U is the strictly upper triangular part.

Example 1.51.

$$\begin{pmatrix} 2&1&0\\ 1&2&1\\ 0&1&2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0&0&0\\ 1&0&0\\ 0&1&0 \end{pmatrix}}_{L} + w\underbrace{\begin{pmatrix} 2&0&0\\ 0&2&0\\ 0&0&2 \end{pmatrix}}_{D} + (1-w)\underbrace{\begin{pmatrix} 2&0&0\\ 0&2&0\\ 0&0&2 \end{pmatrix}}_{D} + \underbrace{\begin{pmatrix} 0&1&0\\ 0&0&1\\ 0&0&0 \end{pmatrix}}_{U}$$

One may consider the iterative scheme with $S = L + wD$ as follows:

$$\mathbf{x}_{n+1} = S^{-1}\mathbf{b} + S^{-1}(S - A)\mathbf{x}_n = S^{-1}\mathbf{b} + (I - S^{-1}A)\mathbf{x}_n.$$

We remark that

$$I - S^{-1}A = I - (L + wD)^{-1}A.$$

Moreover, when $w = 1$, it is just the Gauss-Seidel method. This method is called the *SOR method*. It is clear that the method converges if and only if the iteration matrix has a spectral radius less than one.

Proposition 1.52. *The SOR method converges to the solution of* $A\mathbf{x} = \mathbf{b}$ *if and only if* $\rho(I - (L + wD)^{-1}A) < 1$.

1.3.6 Conjugate Gradient Method

Conjugate gradient (CG) methods are iterative methods for solving linear system of equations $A\mathbf{x} = \mathbf{b}$ where A is symmetric positive definite [11, 101]. This method was first discussed by Hestenes and Stiefel [109]. The motivation of the method is that it involves the process of minimizing quadratic functions such as

$$f(\mathbf{x}) = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}).$$

Here A is symmetric positive definite and this minimization usually takes place over a sequence of Krylov subspaces which is generated recursively by adding a new basic vector $A^k\mathbf{r}_0$ to those of the subspace V_{k-1} generated where

$$\mathbf{r_0} = A\mathbf{x_0} - \mathbf{b}$$

is the residue of the initial vector $\mathbf{x_0}$.

Usually, a sequence of conjugate orthogonal vectors is constructed from V_k so that CG methods would be more efficient. Computing these vectors can

be done recursively which involves only a few vectors if A is self-adjoint with respect to the inner product. The CG methods are attractive since they can give the exact solution after in most n steps in exact arithmetic where n is the size of the matrix A. Hence it can also be regarded as a direct method in this sense. But in the presence of round off errors and finite precision, the number of iterations may be greater than n. Thus, CG methods can be seen as least square methods where the minimization takes place on a particular vector subspace, the Krylov space. When estimating the error of the current solution in each step, a matrix-vector multiplication is then needed. The CG methods are popular and their convergence rates can be improved by using suitable preconditioning techniques. Moreover, it is parameter free, the recursion involved are usually short in each iteration and the memory requirements and the execution time are acceptable for many practical problems.

The CG algorithm reads:

Given an initial guess $\mathbf{x^0}$, A, $\mathbf{b}$, Max, tol:

$\mathbf{r^0} = \mathbf{b} - A\mathbf{x^0}$;

$\mathbf{v^0} = \mathbf{r^0}$;

For $k = 0$ to Max-1 do

If $||\mathbf{v^k}||_2 = 0$ then stop

$t_k =< \mathbf{r^k}, \mathbf{r^k} > / < \mathbf{v^k}, A\mathbf{v^k} >$;

$\mathbf{x^{k+1}} = \mathbf{x^k} + t_k\mathbf{v^k}$;

$\mathbf{r^{k+1}} = \mathbf{r^k} - t_k A\mathbf{v^k}$;

If $||\mathbf{r^{k+1}}, \mathbf{r^{k+1}}||_2 <$ tol then stop

$\mathbf{v^{k+1}} = \mathbf{r^{k+1}} + < \mathbf{r^{k+1}}, \mathbf{r^{k+1}} > / < \mathbf{r^k}, \mathbf{r^k} > \mathbf{v^k}$;

end;

output $\mathbf{x^{k+1}}, ||\mathbf{r^{k+1}}||_2$.

Given a Hermitian, positive definite $n \times n$ matrix H_n, when the conjugate gradient method is applied to solving

$$H_n\mathbf{x} = \mathbf{b}$$

the convergence rate of this method depends on the spectrum of the matrix H_n, see also Golub and van Loan [101]. For example if the spectrum of H_n is contained in an interval, i.e. $\sigma(H_n) \subseteq [a, b]$, then the error in the i-th iteration is given by

$$\frac{||\mathbf{e}_i||}{||\mathbf{e}_0||} \leq 2(\frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}})^i,$$

i.e. the convergence rate is linear. Hence the approximate upper bound for the number of iterations required to make the relative error

$$\frac{||\mathbf{e_i}||}{||\mathbf{e_0}||} \leq \delta$$

is given by

$$\frac{1}{2}(\sqrt{\frac{b}{a}}-1)\log(\frac{2}{\delta})+1.$$

Very often CG method is used with a matrix called preconditioner to accelerate its convergence rate. A good preconditioner C should satisfy the following conditions.

(i) The matrix C can be constructed easily;
(ii) Given right hand side vector $\mathbf{r}$, the linear system $C\mathbf{y} = \mathbf{r}$ can be solved efficiently; and
(iii) the spectrum (or singular values) of the preconditioned system $C^{-1}A$ should be clustered around one.

In the Preconditioned Conjugate Gradient (PCG) method, we solve the linear system

$$C^{-1}A\mathbf{x} = C^{-1}\mathbf{b}$$

instead of the original linear system

$$A\mathbf{x} = \mathbf{b}.$$

We expect the fast convergence rate of the PCG method can compensate much more than the extra cost in solving the preconditioner system $C\mathbf{y} = \mathbf{r}$ in each iteration step of the PCG method.

Apart from the approach of condition number, in fact, condition (iii) is also very commonly used in proving convergence rate. In the following we give the definition of clustering.

Definition 1.53. *We say that a sequence of matrices S_n of size n has a clustered spectrum around one if for all $\epsilon > 0$, there exist non-negative integers n_0 and n_1, such that for all $n > n_0$, at most n_1 eigenvalues of the matrix $S_n^* S_n - I_n$ have absolute values larger than ϵ.*

One sufficient condition for the matrix to have eigenvalues clustered around one is that

$$H_n = I_n + L_n,$$

where I_n is the $n \times n$ identity matrix and L_n is a low rank matrix ($\text{rank}(L_n)$ is bounded above and independent of the matrix size n).

Conjugate Gradient Squared Method

Given a real symmetric, positive definite matrix A of size $n \times n$, the CG method can be used to solve the linear system $A\mathbf{x} = \mathbf{b}$. But in general a non-singular matrix can be neither symmetric nor positive definite. In particular for the applications in queueing systems and re-manufacturing systems in Chapters 2 and 3. In this case, one may consider the normal equation of the original system. i.e.,

$$A^T A\mathbf{x} = A^T\mathbf{b}.$$

Here $A^T A$ is real symmetric and positive definite so that CG method could be applied, but the condition number would then be squared. Moreover, it also involves the matrix-vector multiplication of the form $A^T\mathbf{r}$. These will increase the computational cost. Thus in our context, we propose to employ a generalized CG algorithm, namely the Conjugate Gradient Squared (CGS) method, [193]. This method does not involve the matrix-vector multiplication of the form $A^T\mathbf{r}$.

The CGS algorithm reads:

Given an initial guess $\mathbf{x^0}$, A, $\mathbf{b}$, tol:

$\mathbf{x} = \mathbf{x_0}$;

$\mathbf{r} = \mathbf{b} - A\mathbf{x}$;

$\mathbf{r}' = \mathbf{s} = \mathbf{p} = \mathbf{r}$;

$\mathbf{w} = A\mathbf{p}$;

$\mu = \mathbf{r}'^T\mathbf{r}$;

repeat until $\mu <$ tol;

$\gamma = \mu$;

$\alpha = \gamma / \mathbf{r}'^t\mathbf{r}$;

$\mathbf{q} = \mathbf{s} - \alpha\mathbf{w}$;

$\mathbf{d} = \mathbf{s} + \mathbf{q}$;

$\mathbf{w} = A\mathbf{d}$;

$\mathbf{x} = \mathbf{x} + \alpha\mathbf{d}$;

$\mathbf{r} = \mathbf{r} - \alpha\mathbf{w}$;

otherwise

$\mu = \mathbf{r'}^T\mathbf{r}$;

$\beta = \mu/\gamma$;

$\mathbf{s} = \mathbf{r} - \beta\mathbf{q}$;

$\mathbf{p} = \mathbf{s} + \beta(\mathbf{q} + \beta\mathbf{p})$;
end;

1.3.7 Toeplitz Matrices

We end this subsection by introducing a class of matrices, namely Toeplitz matrices. A Toepltiz matrix T is a matrix having constant diagonals, i.e.

$$T = \begin{pmatrix} t_0 & t_1 & t_2 & \cdots & t_{n-1} & t_n \\ t_{-1} & t_0 & t_1 & \cdots & \cdots & t_{n-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ t_{-n+1} & \cdots & \cdots & \ddots & \ddots & t_1 \\ t_{-n} & t_{-n+1} & \cdots & \cdots & t_{-1} & t_0 \end{pmatrix}.$$

Toeplitz matrices and near-Toeplitz matrices have many applications in applied sciences and engineering such as the multi-channel least squares filtering in time series [171], signal and image processing problems [145]. A survey on the applications of Toeplitz systems can be found in Chan and Ng [46]. Application in solving queueing systems and re-manufacturing systems will be discussed in the Chapters 2 and 3.

In the above applications, solving a Toeplitz or near-Toeplitz system is the focus. Direct methods for solving Toeplitz systems based on the recursion formula are commonly used, see for instance, Trench [199]. For an $n \times n$ Toeplitz matrix T, these direct methods require $O(n^2)$ operations. Faster algorithms that require $O(n \log^2 n)$ operations have also been developed when the Toeplitz matrix is symmetric and positive definite.

An important subset of Toepltiz matrices is the class of circulant matrices. A circulant $n \times n$ matrix C is a Toeplitz matrix such that each column is a cyclic shift of the previous one, i.e.

$$C = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-1} & c_n \\ c_n & c_0 & c_1 & \cdots & c_{n-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & \vdots & \ddots & \ddots & c_1 \\ c_1 & c_2 & \cdots & c_n & c_0 \end{pmatrix}. \tag{1.7}$$

Very often circulant matrices are used to approximate Toeplitiz matrices in preconditioning or finding approximate solution. Because circulant matrices have the following nice property. It is well-known that a circulant matrix can be diagonalized by the discrete Fourier matrix F. More precisely,

$$FCF^* = D = \mathrm{Diag}(d_0, d_1, \ldots, d_n)$$

where F is the discrete Fourier matrix with entries given by

$$F_{j,k} = \frac{1}{\sqrt{n}} e^{-\frac{(2jk\pi)i}{n}}, \quad j, k = 0, 1, \cdots, n-1,$$

and D is a diagonal matrix with elements being the eigenvalues of C, see for instance [82]. Here F^* is the conjugate transpose of F. The matrix-vector multiplication $F\mathbf{y}$ is called the Fast Fourier Transformation (FFT) of the column vector $\mathbf{y}$ and can be done in $O(n \log n)$ operations. Consider for a unit vector

$$\mathbf{e}_1 = (1, 0, \ldots, 0)^T,$$

we have

$$C\mathbf{e}_1 = (c_0, c_n, \ldots, c_1)^T$$

and

$$F\mathbf{e}_1 = \frac{1}{\sqrt{n}}(1, 1, \ldots, 1)^T$$

because the first column of F is a column vector with all entries being equal. Therefore

$$F(c_0, c_n, \ldots, c_1)^T = FC\mathbf{e}_1 = DF\mathbf{e}_1 = \frac{1}{\sqrt{n}}(d_0, d_1, \ldots, d_n)^T$$

and hence the eigenvectors of a circulant matrix C can be obtained by using the FFT in $O(n \log n)$ operations. Moreover, the solution of a circulant linear system can also be obtained in $O(n \log n)$ operations.

The FFT can be used in the Toeplitz matrix-vector multiplication. A Toeplitz matrix can be embedded in a circulant matrix as follows:

$$\tilde{C}(\mathbf{y}, \mathbf{0})^T \equiv \begin{pmatrix} T & S_1 \\ S_2 & T \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{b} \end{pmatrix}. \tag{1.8}$$

Here matrices S_1 and S_2 are such that $\tilde{C}$ is a circulant matrix. Then FFT can be applied to obtain $\mathbf{r} = T\mathbf{y}$ in $O(n \log n)$ operations.

1.4 Hidden Markov Models

Hidden Markov Models (HMMs) are widely used in bioinformatics [135], speech recognition [173] and many other areas [149]. In a HMM, there are two types of states: the observable states and the hidden states. In a HMM, there is no one-to-one correspondence between the hidden states and the observed symbols. It is therefore no longer possible to tell what hidden state the model is in which the observation symbol is generated just by looking at the observation symbol. A HMM is usually characterized by the following elements [173]:

- N, the number of hidden states in the model. Although the states are hidden, for many practical applications there is often some physical significance to the states. For instance, the hidden states represent the CpG island and the non-CpG island in the DNA sequence. We denote the individual states as
$$S = \{s_1, s_2, \cdots, s_N\},$$
and the state at the length t as Q_t.
- M, the number of distinct observation symbols per hidden state. The observation symbols correspond to the physical output of the system being modeled. For instance, A,C,G,T are the observation symbols in the DNA sequence. We denote the individual symbols as
$$V = \{v_1, v_2, \cdots, v_M\}$$
and the symbol at the length t as O_t.
- The state transition probability distribution $[A]_{ij} = \{a_{ij}\}$ where
$$a_{ij} = P(Q_{t+1} = s_j | Q_t = s_i), \quad 1 \leq i, j \leq N.$$
- The observation symbol probability distribution in hidden state j, $[B]_{jk} = \{b_j(v_k)\}$, where
$$b_j(v_k) = P(O_t = v_k | Q_t = s_j), \quad 1 \leq j \leq N, \quad 1 \leq k \leq M.$$
- The initial state distribution $\Pi = \{\pi_i\}$ where
$$\pi_i = P(Q_1 = s_i), \quad 1 \leq i \leq N.$$

Given appropriate values of N, M, A, B and Π, the HMM can be used as a generator to give an observation sequence
$$O = \{O_1 O_2 O_3 \cdots O_T\}$$
where T is the number of observations in the sequence. For simplicity, we use the compact notation
$$\Lambda = (A, B, \Pi)$$
to indicate the complete parameter set of the HMM. According to the above specification, the first order transition probability distribution among the hidden states is used. There are three key issues in HHMMs:

- **Problem 1:**
  Given the observation sequence $O = \{O_1 O_2 \cdots O_T\}$ and a HMM, how to efficiently compute the probability of the observation sequence ?
- **Problem 2:**
  Given the observation sequence $O = \{O_1 O_2 \cdots O_T\}$ and a HMM, how to choose a corresponding state sequence $Q = \{Q_1 Q_2 \cdots Q_T\}$ which is optimal in certain sense ?
- **Problem 3:** Given the observation sequence $O = \{O_1 O_2 \cdots O_T\}$, how to choose the model parameters in a HMM?

For **Problem 1**, a forward-backward dynamic programming procedure [14] is formulated to calculate the probability of the observation sequence efficiently.

For **Problem 2**, it is the one in which we attempt to uncover the hidden part of the model, i.e., to find the "correct" state sequence. In many practical situations, we use an optimality criteria to solve the problem as good as possible. The most widely used criterion is to find a single best state sequence, i.e., maximize the likelihood $P(Q|\Lambda, O)$. This is equivalent to maximizing $P(Q, O|\Lambda)$ since

$$P(Q|\Lambda, O) = \frac{P(Q, O|\Lambda)}{P(O|\Lambda)}.$$

Viterbi algorithm [204] is a dynamic programming technique for finding this single best state sequence

$$Q = \{Q_1, Q_2, \cdots, Q_T\}$$

for the given observation sequence

$$O = \{O_1, O_2, \cdots, O_T\}.$$

For **Problem 3**, we attempt to adjust the model parameters Λ such that $P(O|\Lambda)$ is maximized by using Expectation-Maximization (EM) algorithm. For a complete tutorial on hidden Markov model, we refer readers to the paper by Rabiner [173] and the book by MacDonald and Zucchini [149].

1.5 Markov Decison Process

Markov Decision Process (MDP) has been successfully applied in equipment maintenance, inventory control and many other areas in management science [4, 209]. In this section, we will briefly introduce the MDP, interested readers can also consult the books by Altman [4], Puterman [172] and White [208].

Similar to the case of Markov chain, MDP is a system that can move from one distinguished state to any other possible states. In each step, the decision maker has to take an action from a well-defined set of alternatives. This action affects the transition probabilities of the next move and incurs an immediate

gain (or loss) and subsequent gain (or loss). The obvious problem that the decision maker facing is to determine a suitable plan of actions so that the overall gain is optimized. The process of MDP is summarized as follows:

(i) At time t, a certain state i of the Markov chain is observed.
(ii) After the observation of the state, an action, let us say k is taken from a set of possible decisions A_i. Different states may have different sets of decisions.
(iii) An immediate gain (or loss) $q_i^{(k)}$ is then incurred according to the current state i and the action k taken.
(iv) The transition probabilities $p_{ji}^{(k)}$ is then affected by the action k.
(v) When the time parameter t increases, transition occurs again and the above steps (i)-(iv) repeat.

A policy D is a rule of taking actions. It prescribes all the decisions that should be made throughout the process. Given the current state i, the value of an optimal policy $v_i(t)$ is defined as the total expected gain obtained with t decisions or transitions remained. For the case of one-period remaining, i.e. $t = 1$, the value of an optimal policy is given by

$$v_i(1) = \max_{k \in A_i} \{q_i^{(k)}\}. \tag{1.9}$$

Since there is only one-period remained, an action maximizing the immediate gain will be taken. For the case of two-period remaining, we have

$$v_i(2) = \max_{k \in A_i} \{q_i^{(k)} + \alpha \underbrace{\sum_j p_{ji}^{(k)} v_j(1)}_{\text{subsequent gain}}\} \tag{1.10}$$

where α is the discount factor. Since that the subsequent gain is associated with the transition probabilities which are affected by the actions, an optimal policy should consider both the immediate and subsequent gain. The model can be easily extended to a more general situation, the process having n transitions remained.

$$v_i(n) = \max_{k \in A_i} \{q_i^{(k)} + \alpha \underbrace{\sum_j p_{ji}^{(k)} v_j(n-1)}_{\text{subsequent gain}}\}. \tag{1.11}$$

From the above equation, the subsequent gain of $v_i(n)$ is defined as the expected value of $v_j(n-1)$. Since the number of transitions remained is countable or finite, the process is called the discounted finite horizon MDP. For the infinite horizon MDP, the value of an optimal policy can be expressed as

$$v_i = \max_{k \in A_i} \{q_i^{(k)} + \alpha \sum_j p_{ji}^{(k)} v_j\}. \tag{1.12}$$

The finite horizon MDP is a dynamic programming problem and the infinite horizon MDP can be transformed into a linear programming problem. Both of them can be solved easily by using EXCEL spreadsheet.

1.5.1 Stationary Policy

A stationary policy is a policy that the choice of alternative depends only on the state the system is in and is independent of n. For instance, a stationary policy D prescribes the action $D(i)$ when the current state is i. Define $\bar{D}$ as the associated one-step-removed policy, then the value of policy $w_i(D)$ is defined as

$$w_i(D) = q_i^{D(i)} + \alpha \sum_j p_{ji}^{D(i)} w_j(\bar{D}). \tag{1.13}$$

Given a Markov decision process with infinite horizon and discount factor α, $0 < \alpha < 1$, choose, for each i, an alternative k_i such that

$$\max_{k \in A_i} \{q_i^{(k)} + \alpha \sum_j p_{ji}^{(k)} v_j\} = q_i^{(k_i)} + \alpha \sum_j p_{ji}^{(k_i)} v_j.$$

Define the stationary policy D by $D(i) = k_i$. Then for each i, $w_i(D) = v_i$, i.e. the stationary policy is an optimal policy.

http://www.springer.com/978-0-387-29335-6

Markov Chains: Models, Algorithms and Applications
Ching, W.-K.; Ng, M.K.
2006, XIV, 208 p. 18 illus., Hardcover
ISBN: 978-0-387-29335-6