

A Term Paper
on
Markov Chains and their Applications

by
Azmam Biswas
Enrollment No.: 2022MAM008
under the supervision of
Prof. M. Mitra



INDIAN INSTITUTE OF ENGINEERING SCIENCE AND
TECHNOLOGY, Shibpur
Department of Mathematics

Submitted for the partial fulfillment of the requirements for the award of the
degree of
M.Sc. in Applied Mathematics

CERTIFICATE

This is to certify that the Term Paper on **"Markov Chains and their Applications"** submitted by Azmain Biswas to the Department of Mathematics, Indian Institute of Engineering Science and Technology, Shibpur, Howrah-711103, for 2nd semester of Master of Science in Applied Mathematics is a record of project work carried out by him under my supervision. The result of this project work or any part thereof has not been submitted for any degree or diploma.

Prof. M. Mitra
Department of Mathematics
Indian Institute of Engineering Science and Technology
Shibpur, Howrah.

Acknowledgement

Thanks to my supervisor, Prof. M. Mitra, for a very interesting project. It has been a great pleasure to work under his guidance, and I have learned a lot from his way of thinking and his approach to a problem.

I am also indebted to Prof. Pritha Das, Head of the Department of Mathematics, Indian Institute of Engineering Science and Technology, Shibpur, for her help.

At various stages, I received encouragement and inspiration from most of the faculty members of the department. I would like to thank all of them for their invaluable words of advice.

I have benefited greatly from the advantages of the T_EX-group software typing systems, the use of which has contributed much to simplify the typing of long mathematical expressions involved in my work.

I would like to thank my parents and friends, for being so supportive of my career and life choices, as well as for making me happy.

Azmair Biswas

Contents

1	Introduction	4
2	Markov Chain	5
2.1	Definition	5
2.2	Homogeneous Markov Chain	5
2.3	Chapman-Kolmogorov Equations	6
2.4	Classification of states	9
3	Stationary Probability	12
3.1	Reversibility	14
4	Some Examples of Markov Chain	17
4.1	The Simple Random Walk	17
4.1.1	Random Walk in Simulation	18
4.2	The Gambler's Ruin Problem	20
4.2.1	Becoming infinitely rich or getting ruined	21
4.2.2	Gambler's Ruin Problem Simulation	21
5	Application of Markov Chain	23
5.1	Google PageRank	23
5.2	Text Generator	25
5.3	Other Applications	25
A	Random Walk in Python	28
B	The Gambler's Ruin Problem	29
C	Google PageRank	30
D	Text Generator	31

Chapter 1

Introduction

The Weak Law of Large Numbers(WLLN) and the Central Limit Theorem(CLT) were proved around 1880's. At that time those were the focal probabilistic issues. In 1902 Russian mathematician and philosopher P.A. Nekrasov claimed that pairwise independent of random variables were necessary condition for CTL to hold. This motivated Andrei A. Markov to construct a counterexample to disprove the claim of P.A. Nekrasov. So he developed Markov Chain in 1907 to disprove a "Mathematical proof of freewill" which had been proposed by P.A.Nekrasov to defend the church.

A Markov chain is a stochastic model that describes a sequence of possible events or transitions from one state to another of a system. The probability of transitions from state to state only depends on the current state of the system. A Markov Chain is used to unravel predictions about future states of a stochastic process using only knowledge of the present state. This property of "forgetting" past states is known as the memoryless property. This memoryless property allows for efficient and concise modeling of systems with a large number of states and complex transition dynamics.

Markov chains provide a wide range of analytical methods and tools for comprehending how systems behave over time. The long-term behaviour and stability of a Markov chain can be ascertained by computing transition probabilities and studying steady-state distributions. Additionally, ideas like recurrent and transient states, absorbing states, and mean first passage time shed light on a number of the underlying system's properties and characteristics.

Markov chains have a wide range of uses in many different fields. They have been critical in the modelling and analysis of numerous real-world phenomena, including financial markets, biological processes, social networks, and line-waiting systems. Additionally, Markov chains have found application in decision-making algorithms, search engines, data analysis, and speech recognition.

This paper aims to provide a comprehensive introduction to Markov chains, covering their fundamental concepts, properties, and applications.

In the upcoming section, we will look at the definition and fundamental ideas of Markov chains, as well as look at their characteristics, analytical methods, and a variety of uses. Readers will have a strong foundation by the end to further investigate.

Chapter 2

Markov Chain

2.1 Definition

Definition 2.1.1 (Markov Chain). A discrete time stochastic process $\{X_n, n = 1, 2, 3, \dots\}$ is defined to be *Discrete Time Markov Chain* or simply *Markov Chain* if it takes value the state space \mathbf{S} , and for every $n \geq 0$ it satisfy the property

$$\mathbf{P}(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0) = \mathbf{P}(X_{n+1} = j | X_n = i) \quad (2.1)$$

Unless otherwise mentioned we take the state space \mathbf{S} to be $\{0, 1, 2, 3, \dots\}$. If $X_n = i$ we say that the process is in i th state at time n . In the definition eq. (2.1) may be interpreted as for Markov Chain, the conditional distribution of any future state X_{n+1} , given the past states X_0, X_1, \dots, X_{n-1} and the present state X_n , is independent of the past and only depend on the present state. This property is called *Markovian Property*. In other word for Markov chain predicting the future we only need information about the present state.

Note: Assumption we are making for Markov chain to forget the past as long as present is known is very strong assumption.

2.2 Homogeneous Markov Chain

Definition 2.2.1 (Homogeneous Markov Chain). We say a Markov chain $\{X_n, n \geq 0\}$ is homogeneous if $\mathbf{P}(X_{n+1} = j | X_n = i) = \mathbf{P}(X_2 = j | X_1 = i) \forall n > 0$.

The quantity $\mathbf{P}(X_{n+1} = j | X_n = i)$ is called the *transition probability* from state i to state j . For homogeneous Markov Chain we can specify the transition probabilities $\mathbf{P}(X_{n+1} = j | X_n = i)$ by a sequence of value $p_{ij} = \mathbf{P}(X_{n+1} = j | X_n = i)$.

Then the transition probabilities are $p_{ij}, 1 \leq i, j \leq \infty$ for transition from state i to state j . The matrix $P = (p_{ij})$ is called The *Transition Matrix* of chain. Since probabilities are nonnegative and since the process must make a transition into some state, we have

$$p_{ij} \geq 0, \quad i, j \geq 0, \quad \text{and} \quad \sum_{j=0}^{\infty} p_{ij} = 1, \quad \forall i = 0, 1, 2, \dots$$

For, infinite state Markov chain the probability transition matrix will be infinite order. Then,

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \dots \\ p_{10} & p_{11} & p_{12} & \dots \\ \vdots & \vdots & \vdots & \dots \\ p_{i0} & \dots & p_{ij} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Example 2.2.1 (Rain and sunny). Suppose that whether it rains today depends on previous weather conditions only from the last two days. Specifically, suppose that if it has rained for the past two days, then it will rain tomorrow with probability 0.7; if it rained today but not yesterday, then it will rain tomorrow with probability 0.5; if it rained yesterday but not today, then it will rain tomorrow with probability 0.4; if it has not rained in the past two days, then it will rain tomorrow with probability 0.2.

We can transform it into a Markov chain by letting the state on any day be determined by the weather conditions during both that day and the preceding one. For instance, we can say that the process is in

State 0: if it rained both today and yesterday

State 1: if it rained today but not yesterday

State 2: if it rained yesterday but not today

State 3: if it rained neither today nor yesterday

The it will be 4-state Markov chain whose transition matrix will be,

$$P = \begin{bmatrix} 0.7 & 0 & 0.3 & 0 \\ 0.5 & 0 & 0.5 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0.2 & 0 & 0.8 \end{bmatrix}$$

2.3 Chapman-Kolmogorov Equations

We have already defined the one step transition probabilities p_{ij} . We now define the n-step transition probabilities p_{ij}^n to be the probability that a process in state i will be in state j after n additional transitions. i.e.

$$p_{ij}^n = \mathbf{P}(X_{n+m} = j | X_m = i), \quad n \geq 0, \quad i, j \geq 0.$$

By definition of Markov chain we get,

$$\begin{aligned} p_{ij}^{n+m} &= \mathbf{P}(X_{n+m} = j | X_0 = i) \\ &= \sum_{k=0}^{\infty} \mathbf{P}(X_{n+m} = j, X_n = k | X_0 = i) \quad (\text{By theorem of total probability}) \\ &= \sum_{k=0}^{\infty} \mathbf{P}(X_{n+m} = j | X_n = k, X_0 = i) \mathbf{P}(X_n = k | X_0 = i) \\ &= \sum_{k=0}^{\infty} \mathbf{P}(X_{n+m} = j | X_n = k) \mathbf{P}(X_n = k | X_0 = i) \end{aligned}$$

Then,

$$p_{ij}^{n+m} = \sum_{k=0}^{\infty} p_{kj}^m p_{ik}^n. \quad (2.2)$$

If we take $n = m = 1$. Then

$$p_{ij}^2 = \sum_{k=0}^{\infty} p_{kj} p_{ik} \quad (2.3)$$

the above expression is (i, j) element of P^2 matrix then we see eq. (2.3) in matrix form,

$$P^2 = P \cdot P$$

Hence, eq. (2.2) can also written in matrix form,

$$P^{n+m} = P^n \cdot P^m$$

Where P^n and P^m are the n -step and m -step transition matrix respectively.

Example 2.3.1 (Transition matrix of 4-state Markov chain). Consider the 4-state Markov chain depicted in Fig 2.1 When no probabilities are written over the arrows, as in this case, it means all arrows originating from a given state are equally likely. For example, there are 3 arrows originating from state 1, so the transitions $1 \rightarrow 3$, $1 \rightarrow 2$, and $1 \rightarrow 1$ all have probability $1/3$. Therefore, the transition matrix of the chain is.

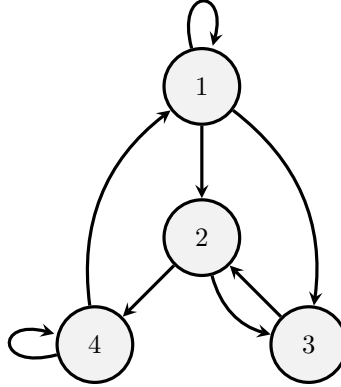


Figure 2.1: Example of a 4-state Markov chain

$$P = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

To compute the probability that the chain is in state 3 after 5 steps, starting at state 1, we would look at the $(1,3)$ entry of P^5 .

$$P^5 = \begin{bmatrix} \frac{853}{3888} & \frac{509}{1944} & \frac{52}{243} & \frac{395}{1296} \\ \frac{173}{864} & \frac{85}{432} & \frac{31}{108} & \frac{91}{288} \\ \frac{37}{144} & \frac{29}{72} & \frac{1}{9} & \frac{11}{48} \\ \frac{499}{2592} & \frac{395}{1296} & \frac{71}{324} & \frac{245}{864} \end{bmatrix}$$

so, $p_{13}^5 = \mathbf{P}(X_5 = 3 | X_0 = 1) = \frac{52}{243}$.

To get the marginal distributions of X_0, X_1, \dots , we need to specify not just the transition matrix, but also the initial conditions of the chain. This can be done by setting the initial state X_0 to be a particular state, or by randomly choosing X_0 according to some distribution. Let (t_1, t_2, \dots, t_N) be the PMF of X_0 displayed as a vector, that is, $t_i = \mathbf{P}(X_0 = i)$. Then the marginal distribution of the chain at any time can be computed from the transition matrix.

Proposition 2.3.1 (Marginal Distribution of X_n). *Define $\mathbf{t} = (t_1, t_2, \dots, t_n)$ by $t_i = \mathbf{P}(X_0 = i)$, and view \mathbf{t} as a row vector. Then the marginal distribution of X_n is given by the vector $\mathbf{t}P^n$. That is the j -th component of $\mathbf{t}P^n$ is $\mathbf{P}(X_n = j)$.*

Proof. By the law of total probability we get,

$$\begin{aligned}\mathbf{P}(X_n = j) &= \sum_{i=0}^N \mathbf{P}(X_n, X_0 = i) \\ &= \sum_{i=0}^N \mathbf{P}(X_0 = i) \mathbf{P}(X_n = j | X_0 = i) \\ &= \sum_{i=0}^N t_i p_{ij}^n\end{aligned}$$

Which is the j -th component of $\mathbf{t}P^n$. □

Example 2.3.2 (Marginal distribution of 4-state Markov chain). Again consider the 4-state Markov chain in Fig. 2.1. Suppose the initial conditions are $\mathbf{t} = (1/4, 1/4, 1/4, 1/4)$. Let X_n be the position of the chain at time n . Then distribution of X_1 is

$$\begin{aligned}\mathbf{t}P &= \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \end{bmatrix} \\ &= \begin{bmatrix} \frac{5}{24} & \frac{1}{3} & \frac{5}{24} & \frac{1}{4} \end{bmatrix}\end{aligned}$$

The marginal distribution of X_5 is

$$\begin{aligned}\mathbf{t}P^5 &= \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} \frac{853}{3888} & \frac{509}{1944} & \frac{52}{243} & \frac{395}{1296} \\ \frac{173}{864} & \frac{85}{432} & \frac{31}{108} & \frac{91}{288} \\ \frac{37}{144} & \frac{29}{72} & \frac{1}{9} & \frac{11}{48} \\ \frac{499}{2592} & \frac{395}{1296} & \frac{71}{324} & \frac{245}{864} \end{bmatrix} \\ &= \begin{bmatrix} \frac{707}{3254} & \frac{472}{1619} & \frac{101}{486} & \frac{1469}{5184} \end{bmatrix}\end{aligned}$$

2.4 Classification of states

Definition 2.4.1. We say that state j is *accessible* from state i , written as $i \rightarrow j$, If $p_{ij}^n > 0$ for some $n \geq 0$.

We assume every state is accessible from itself since

$$p_{ii}^0 = \mathbf{P}(X_0 = i | X_0 = i) = 1.$$

Definition 2.4.2. Two states i and j are said to *communicate*, written as $i \longleftrightarrow j$, if they are accessible from each other.

i.e.

$$i \longleftrightarrow j \implies i \rightarrow j \quad \& \quad j \rightarrow i$$

Communication is an equivalence relation. That means that,

1. $i \longleftrightarrow i$,
2. if $i \longleftrightarrow j$ then $j \longleftrightarrow i$,
3. if $i \longleftrightarrow j$ and $j \longleftrightarrow k$ then $i \longleftrightarrow k$.

First two properties are obvious from definition, let for some $n, m \geq 0$ then, $p_{ij}^n, p_{jk}^m > 0$ by assumption. By Chapman-Kolmogorov equation.

$$p_{ik}^{n+m} = \sum_{r=0}^{\infty} p_{ir}^n p_{rk}^m \geq p_{ij}^n p_{jk}^m > 0.$$

Hence state k is accessible from state i . By same one can prove that i is also accessible from K .

Therefore, the states of a Markov chain can be partitioned into communicating classes such that only members of the same class communicate with each other. i.e. two states i & j belong to same class if and only if $i \longleftrightarrow j$.

Example 2.4.1. Consider the markov chain define in the picture fig. 2.2.

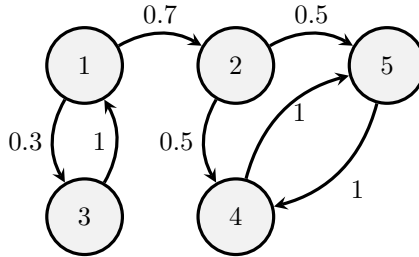


Figure 2.2: Communication classes

here the classes are $\{1, 3\}, \{2\}, \{4, 5\}$

Definition 2.4.3 (Irreducible Markov chain). A Markov chain is said to be irreducible if it has only one communicating class. That is, every state communicate with each other.

That is, for any states i, j there is some positive integer n such that the (i, j) entry of P^n is positive.

A Markov chain that is not irreducible called reducible.

For any state i and j define f_{ij}^n to be the probability that, starting from i , the first transition into j occurs at n time.

i.e.

$$f_{ij}^n = \mathbf{P}(X_n = j, X_k \neq j, k = 1, 2, \dots, n-1 | X_0 = i).$$

Let

$$f_{ij}^* = \sum_{n=0}^{\infty} f_{ij}^n$$

Then, f_{ij}^* denote the probability of ever making a transition into step j when start from state i . If j is not accessible from i then f_{ij}^* will be zero.

Definition 2.4.4 (Recurrent and Transient state). A state j of a Markov chain is said to be *recurrent* if $f_{ii}^* = 1$ and *transient* if $f_{ii}^* < 1$.

In other word, if a Markov chain start in a recurrent state, there is a guarantee that it will visit that state again in the future (eventually return to that state with probability 1).

In contrast, a transient state in a Markov chain is a state where, once you reach it, there is a positive probability that you will never return to that state. i.e. if you begin in a transient state, there's a chance you won't return there.

Corollary 2.4.0.1. *state i is recurrent if and only if $\sum_{n=0}^{\infty} p_{ii}^n = \infty$*

Proof. The state i is recurrent if and only if, starting in state i , the expected number of time periods that the process is in state i is infinite.

Let

$$I_n = \begin{cases} 1, & \text{if } X_n = i \\ 0, & \text{if } X_n \neq i \end{cases}$$

we then have $\sum_{n=0}^{\infty} I_n$ represent the number of periods that the process is in state i Then,

$$\begin{aligned} E \left[\sum_{n=0}^{\infty} I_n | X_0 = i \right] &= \sum_{n=0}^{\infty} E[I_n | X_0 = i] \\ &= \sum_{n=0}^{\infty} \mathbf{P}(X_n = i | X_0 = i) \\ &= \sum_{n=0}^{\infty} p_{ii}^n \end{aligned}$$

Hence the result. □

Theorem 2.4.1. *If i is recurrent and $i \longleftrightarrow j$, then j is also recurrent.*

Proof. Let m and k be such that $p_{ij}^k > 0, p_{ji}^m > 0$. Now for any $n \geq 0$

$$p_{jj}^{m+n+k} \geq p_{ji}^m p_{ii}^k p_{ij}^n$$

This follows because the left-hand side of the preceding equation is the probability of going $j \rightarrow j$ in $m + n + k$ steps, whereas the right-hand side is the probability of going $j \rightarrow j$ in $m + n + k$ steps via a path that goes from $j \rightarrow i$ in m steps, then $i \rightarrow i$ in n additional steps, then $i \rightarrow j$ in k additional steps. By summing the preceding over n , we obtain

$$\sum_{n=0}^{\infty} p_{jj}^{m+n+k} \geq p_{ji}^m p_{ii}^k \sum_{n=0}^{\infty} p_{ij}^n = \infty$$

where $\sum_{n=0}^{\infty} p_{ij}^n = \infty$ because state i is recurrent. Thus, j is also recurrent. □

Proposition 2.4.2. *In an irreducible Markov chain with a finite state space, all states are recurrent.*

Definition 2.4.5 (Absorbing State). A state i of Markov chain is called absorbing if $p_{ii} = 1$ that is, it is impossible to leave the state.

A Markov Chain is absorbing if it has at least one absorbing state.

For example consider the Markov chain shown in Fig. 2.3

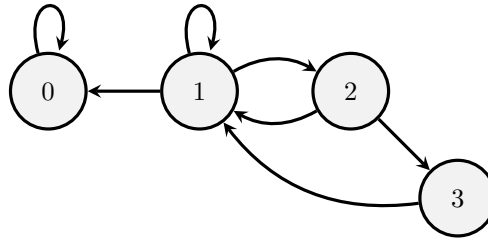


Figure 2.3: Absorbing Markov Chain

here $p_{00} = 1$ hence, State 0 is Absorbing State in this case.

Chapter 3

Stationary Probability

Consider a two-state Markov chain with transition probability matrix

$$P = \begin{bmatrix} 0.35 & 0.65 \\ 0.89 & 0.11 \end{bmatrix}$$

Then 2-step transition matrix will be

$$P^2 = \begin{bmatrix} 0.7010 & 0.2990 \\ 0.4094 & 0.5906 \end{bmatrix}$$

Then 8-step transition matrix will be

$$P^8 = \begin{bmatrix} 0.5810 & 0.4190 \\ 0.5737 & 0.4263 \end{bmatrix}$$

Once again 12-state transition matrix is,

$$P^{12} = \begin{bmatrix} 0.5782 & 0.4218 \\ 0.5776 & 0.4224 \end{bmatrix}$$

Note that The 8-step transition matrix is almost identical to 12-step transition matrix. So it seems that p_{ij}^n is converging to some value as $n \rightarrow \infty$ that is same for all i . In other words, In other word it seems to exist a limiting probability that the process will be in state j after a large number of transitions, and this value is independent of initial state. This limiting probability is known as stationary probability and the distribution of X_j^n as $n \rightarrow \infty$ is known as stationary distribution.

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*. We denote period of i by $d(i)$.

If state i is recurrent, then it is said to be positive recurrent if, starting in state the expected time until the process returns to state i is finite. While there exist recurrent states that are not positive recurrent (such states are called null recurrent). Positive recurrent, aperiodic states are called *ergodic*.

Definition 3.0.1 (Stationary Distribution). A probability distribution $\{p_j, j \geq 0\}$ is called stationary for the Markov chain if

$$p_j = \sum_{i=0}^{\infty} p_i p_{ij}, \quad j \geq 0 \tag{3.1}$$

i.e. If $\mathbf{t} = (p_1, p_2, \dots, p_j, \dots)$ is a probability distribution vector and P is transition matrix, Then

$$\mathbf{t} = \mathbf{t}P. \quad (3.2)$$

From eq. (3.2) we see that 1 is a eigenvalue of transition matrix P and \mathbf{t} is eigenvector corresponding to 1. Since in transition matrix such that $\sum_{j=0}^{\infty} p_{ij} = 1 \forall i$ i.e. sum of all elements of row is 1, 1 must be a eigenvalue.

Now the question arises is eigenvector corresponding to 1 is a probability vector (sum of all component in 1) or not.

Theorem 3.0.1 (Existence and Uniqueness of Stationary Distribution). *An irreducible aperiodic Markov chain belongs to one of the following two classes:*

1. *Either the state are all transient or all null recurrent; in this case $p_{ij}^n \rightarrow 0$ as $n \rightarrow \infty \forall i, j$ and there exist no stationary distribution.*
2. *Or else, all states are positive recurrent, i.e.*

$$\pi_j = \lim_{n \rightarrow \infty} p_{ij}^n \geq 0. \quad (3.3)$$

In this case, $\{\pi_j, j = 0, 1, \dots\}$ is stationary distribution and there exist no other stationary distribution.

Proof. we will first proof (2), we have,

$$\sum_{j=0}^M p_{ij}^n \leq \sum_{j=0}^{\infty} p_{ij}^n = 1 \forall M.$$

Letting $n \rightarrow \infty$ we get,

$$\begin{aligned} \sum_{j=0}^M \pi_j &\leq 1 \forall M. \\ \sum_{j=0}^{\infty} \pi_j &\leq 1. \end{aligned}$$

Now,

$$p_{ij}^{n+1} = \sum_{k=0}^{\infty} p_{ik}^n p_{kj} \geq \sum_{k=0}^M p_{ik}^n p_{kj} \forall M.$$

Letting $n \rightarrow \infty$ gives us,

$$\pi_j \geq \sum_{k=0}^M \pi_k p_{kj} \forall M,$$

i.e.

$$\pi_j \geq \sum_{k=0}^{\infty} \pi_k p_{kj}, \quad \forall j \geq 0.$$

To show that the above is actually an equality, suppose the inequality is strict for some j . Then we obtain,

$$\sum_{j=0}^{\infty} \pi_j > \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \pi_k p_{kj} = \sum_{k=0}^{\infty} \pi_k \sum_{j=0}^{\infty} p_{kj} = \sum_{k=0}^{\infty} \pi_k.$$

which is contradiction. Therefore,

$$\pi_j = \sum_{k=0}^{\infty} \pi_k p_{kj}, \quad \forall j, \dots$$

Putting $p_j = \pi_j / \sum_0^\infty \pi_k$, we see that $\{p_j, j = 0, 1, 2, \dots\}$ is stationary distribution, and hence at least one stationary distribution exist. Now let $\{p_j, j = 0, 1, 2, \dots\}$ be any stationary distribution. Then if $\{p_j, j = 0, 1, 2, \dots\}$ is the probability distribution of X_0 then

$$\begin{aligned} p_j &= \mathbf{P}(X_n = j) \\ &= \sum_{i=0}^{\infty} \mathbf{P}(X_n = j | X_0 = i) \mathbf{P}(X_0 = i) \\ &= \sum_{i=0}^{\infty} p_{ij}^n p_i \end{aligned} \quad (3.4)$$

Then, from eq. (3.4) we see,

$$p_j \geq \sum_{i=0}^M p_{ij}^n p_i \quad \forall M.$$

Letting n and then M approach ∞ we get,

$$p_j \geq \sum_{i=0}^{\infty} \pi_j p_i = \pi_j.$$

Now we show that $p_j \leq \pi_j$ use eq. (3.4) and that fact that $p_{ij}^n \leq 1$ to obtain

$$p_j \leq \sum_{i=0}^M p_{ij}^n p_i + \sum_{i=M+1}^{\infty} p_i \quad \forall M.$$

and letting $n \rightarrow \infty$ gives,

$$p_j \leq \sum_{i=0}^M \pi_j p_i + \sum_{i=M+1}^{\infty} p_i \quad \forall M.$$

Since $\sum_0^\infty = 1$, we obtain upon letting $M \rightarrow \infty$ that

$$p_j \leq \sum_{i=0}^{\infty} \pi_j p_i = \pi_j. \quad (3.5)$$

If the state are transient or null recurrent and $\{p_j, j = 0, 1, 2, \dots\}$ is a stationary distribution, then eq. (3.4) hold and $p_{ij}^n \rightarrow 0$, which is clearly impossible. Thus, for case (1), no stationary distribution exists and the proof is complete. \square

Remark 1. It is quite intuitive that if the process is started with the limiting probabilities, then the resultant Markov chain is stationary. For in this case the Markov chain at time 0 is equivalent to an independent Markov chain with the same P matrix at time ∞ . Hence the original chain at time t is equivalent to the second one at time $\infty + t = \infty$ therefore stationary.

From eq. (3.3) we see that for stationary distribution vector π is row vector and

$$P^n \rightarrow \begin{bmatrix} \pi \\ \pi \\ \pi \\ \vdots \\ \vdots \end{bmatrix} \quad \text{as } n \rightarrow \infty$$

3.1 Reversibility

Consider a stationary Markov Chain having transition probability p_{ij} and the stationary probabilities π_i suppose that we construct a sequence of states going backwards in time. That is, starting

at time n consider the sequence of states $X_n, X_{n-1}, X_{n-2}, \dots$. It is also a Markov Chain with probabilities p_{ij}^* define by,

$$\begin{aligned} p_{ij}^* &= P(X_m = j | X_{m+1} = i) \\ &= \frac{P(X_{m+1} = i | X_m = j)}{P(X_{m+1} = i)} \\ &= \frac{\pi_j p_{ji}}{\pi_i} \text{ as it is in stationary state.} \end{aligned} \quad (3.6)$$

From eq. (3.6) we observe that probability of present state X_{m+1} given past X_m , $m > 0$ is independent of future. Thus the reversed process is also a Markov Chain.

Definition 3.1.1 (Time Reversible Markov Chain). If for any Markov chain $p_{ij}^* = p_{ij}$, $\forall i, j$ then the Markov chain is called time Reversible.

Hence the condition for time Reversibility is

$$\pi_i p_{ij} = \pi_j p_{ji} \quad \forall i, j$$

Proposition 3.1.1 (Reversible implies stationary). Suppose that $P = (q_{ij})$ is a transition matrix of a Markov chain that is reversible with respect to a non-negative vector $\mathbf{s} = (s_1, \dots, s_M)$ whose components sum to 1. Then s is a stationary distribution of the chain.

Proof. We have

$$\sum_{i=0}^{\infty} s_i p_{ij} = \sum_{i=0}^{\infty} s_j p_{ji} = s_j \sum_{i=1}^{\infty} p_{ji} = s_j,$$

So, \mathbf{s} is stationary. □

Consider a graph having a positive number w_{ij} associate with each edge (i, j) and suppose that a particle moves vertex to vertex in the following manner if the particle is presently at a vertex i then it will move to next vertex j with probability

$$p_{ij} = \frac{w_{ij}}{\sum_j w_{ij}} \quad (3.7)$$

where $w_{ij} = 0$ if (i, j) is not an edge of the graph. This Markov chain is called Random Walk on an edge weighted graph.

An irreducible positive recurrence Markov Chain is stationary if the initial state is chosen according to the stationary probabilities. Then we say that such chain is in steady state.

Proposition 3.1.2. Consider a random walk on an edge weighted graph with a finite number of vertices if this Markov chain is irreducible then it is in steady state, time reversible with stationary probability given by

$$\pi_i = \frac{\sum_j w_{ij}}{\sum_i \sum_j w_{ij}}$$

Proof. The time Reversibility equation $\pi_i p_{ij} = \pi_j p_{ji}$ and from eq. (3.7) we get,

$$\frac{\pi_i w_{ij}}{\sum_k w_{ik}} = \frac{\pi_j w_{ji}}{\sum_k w_{jk}} = c$$

or $\pi_i = c \sum_k w_{ik}$ doing sum over all j we get the desire result. □

Example 3.1.1. What is the average number of steps it would take for a randomly moving knight returns to its starting point.

Solution: If a knight start from any 4(b1 , g1, b8, g8) of its starting position the process will be same as all positions are symmetric. Let us start from b1. From b1 position knight has equal 1/3 probability to going to a1, b3, d2. Then we consider the chess board as a graph where each node connected to another if a knight has a legal move between them. Then total number of node 64 and total number of edge $3 \times 8 + 2 \times 4 + 4 \times 20 + 6 \times 16 + 8 \times 16 = 336$. Then stationary distribution for b1 node is

$$\pi_{b1} = \frac{\sum_j w_{ij}}{\sum_i \sum_j w_{ij}} = \frac{3}{336} = \frac{1}{112}.$$

Now after very long time of randomly hopping around the board the knight has probability $\frac{1}{112}$ that it will be on the node b1. Then knight will take 112 average hopping to return to b1.

Chapter 4

Some Examples of Markov Chain

4.1 The Simple Random Walk

The markov chain whose state space is the set of all integers \mathbb{Z} and has transition probabilities

$$p_{i,i+1} = p = 1 - p_{i,i-1}, \quad i \in \mathbb{Z}.$$

where $0 < p < 1$, is called the simple random walk.

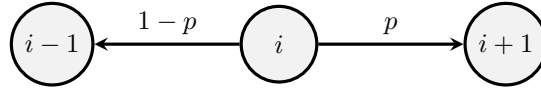


Figure 4.1: Simple Random Walk

One interpretation is that it represents the winnings of a gambler who on each play of the game either wins or loses one dollar.

Since all states clearly communicate it follows that from theorem 2.4.1 they are either all transient or all recurrent. So let us consider state 0 and attempt to determine if $\sum_{n=0}^{\infty} p_{00}^n$ is finite or not.

Since it is impossible to be back at the initial state after an odd number of transitions, we must have,

$$p_{00}^{2n+1} = 0 \quad \forall n \geq 0.$$

On the other hand, the gambler would be even after $2n$ trials if and only if he wins n games and lost n of games. It is like coin toss with probability of getting head is p and tail is $1-p$. Then the desired probability is those binomial probabilities. Then we have,

$$p_{00}^{2n} = \binom{2n}{n} p^n (1-p)^n = \frac{2n!}{n!n!} (p(1-p))^n, \quad n = 1, 2, 3, \dots$$

by using Stirling's approximation, i.e.

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

we obtain

$$p_{00}^{2n} \sim \frac{[4p(1-p)]^n}{\sqrt{n\pi}}$$

we know if $a_n \sim b_n$, then $\sum_n a_n < \infty$ iff $\sum_n b_n < \infty$. Consequently $\sum_n p_{00}^n < \infty$ if and only if

$$\sum_{n=0}^{\infty} \frac{[4p(1-p)]^n}{\sqrt{n\pi}} < \infty$$

However, $4p(1-p) \leq 1$ with equality holding if and only if $p = \frac{1}{2}$. Hence $\sum_{n=1}^{\infty} p_{00}^n = \infty$ if and only if $p = \frac{1}{2}$. Thus, the chain is recurrent with $p = \frac{1}{2}$ and transient when $p \neq \frac{1}{2}$.

Hence, if $p = 1/2$ then the process random walk visit state 0 again and again infinitely many time and if $p \neq 1/2$ it may or may not visit state 0.

If we consider random walk has $N + 1$ step i.e. $\{0, 1, \dots, N\}$ is state space of random walk. Then, The transition matrix will be.

$$P = \begin{bmatrix} 1-p & p & 0 & 0 & \dots \\ 1-p & 0 & p & 0 & \dots \\ 0 & 1-p & 0 & p & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}_{(N+1) \times (N+1)}$$

considering $p_{00} = 1 - p$ and $p_{NN} = p$ as those are last state.

Example 4.1.1. Consider a random walk with 4 state and $p = 0.6$. Then, the transition probability will be,

$$P = \begin{bmatrix} 0.4 & 0.6 & 0 & 0 \\ 0.4 & 0 & 0.6 & 0 \\ 0 & 0.4 & 0 & 0.6 \\ 0 & 0 & 0.4 & 0.6 \end{bmatrix}$$

then 10000-step transition probability will be

$$P^{10000} = \begin{bmatrix} 0.1231 & 0.1846 & 0.2769 & 0.4154 \\ 0.1231 & 0.1846 & 0.2769 & 0.4154 \\ 0.1231 & 0.1846 & 0.2769 & 0.4154 \\ 0.1231 & 0.1846 & 0.2769 & 0.4154 \end{bmatrix}$$

hence, Stationary distribution will be

$$\pi = [0.1231 \quad 0.1846 \quad 0.2769 \quad 0.4154]$$

As, $\pi = \pi P$

4.1.1 Random Walk in Simulation

Using python we can simulate a Random Walk. And observing the output we find same result as above. If $p = 0.5$ the Random walk will visit state 0 over and over again. But it is not the case for $p = 0.6$ and $p = 0.45$.

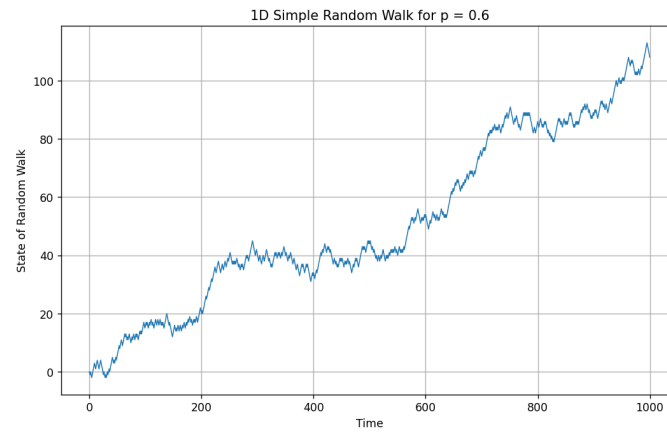


Figure 4.2: Random Walk for $p = 0.6$

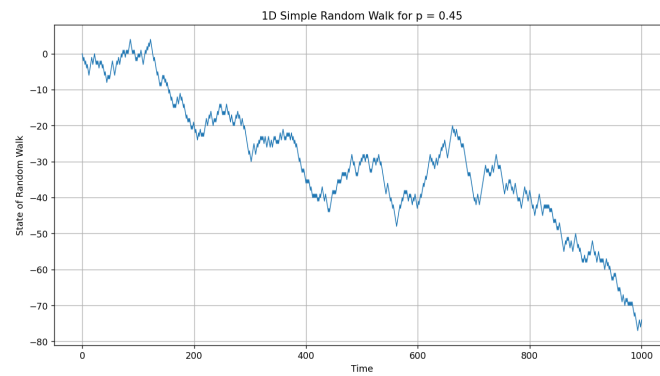


Figure 4.3: Random Walk for $p = 0.45$

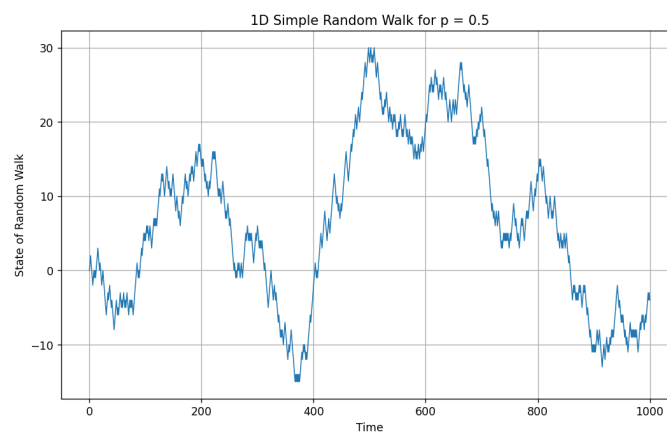


Figure 4.4: Random Walk for $p = 0.5$

4.2 The Gambler's Ruin Problem

Consider a gambler who starts with an initial fortune and then on each successive gamble either wins \$1 or loses \$1 independent of the past with probabilities p and $q = 1 - p$ respectively. The gambler's objective is to reach a total fortune of $\$N$, without first getting ruined (running out of money). If the gambler succeeds, then the gambler is said to win the game. In any case, the gambler stops playing after winning or getting ruined, whichever happens first. Let the gambler starts with $\$i$ where $0 < i < N$.

Let X_n denote the total fortune after the n th gamble. Then $X_0 = i$, Then the process $\{X_n, n = 0, 1, 2, \dots, N\}$ is a Markov chain With transition probabilities

$$\begin{aligned} p_{00} &= p_{NN} = 1, \\ p_{i,i+1} &= p = 1 - p_{i,i-1} \quad i = 1, 2, \dots, N-1 \end{aligned}$$

This Markov chain has three classes, classes are $\{0\}$, $\{1, 2, \dots, N-1\}$ & $\{N\}$ the first and third class being recurrent and second transient. Since all each transient state is only visited finitely often, it follows that after some finite amount of time, the gambler will either wins $\$N$ or go broke.

Since the game stops when either $X_n = 0$ or $X_n = N$, let

$$\tau_i = \min \{n \geq 0 : X_n \in \{0, N\} | X_0 = i\}$$

denote the time at which the game stops when $X_0 = i$. If $X_{\tau_i} = N$, then the gambler wins of $X_{\tau_i} = 0$ gambler is ruin.

Let $P_i = \mathbf{P}(X_{\tau_i} = N)$ denotes the probability that the gambler wins when $X_0 = i$ clearly $P_0 = 0$ and $P_N = 1$ by definition, (Here $p_{00} = 1$ and $p_{NN} = 1$ then state 0 and N is absorbing state.) we next proceed to compute P_i , $1 \leq i \leq N-1$. By conditioning on the outcome of the initial play of the game, we obtain,

$$P_i = pP_{i+1} + qP_{i-1}, \quad i = 1, 2, \dots, N-1,$$

Since $p + q = 1$ we get,

$$\begin{aligned} pP_i + qP_i &= pP_{i+1} + qP_{i-1} \\ P_{i+1} - P_i &= \frac{q}{p}(P_i - P_{i-1}), \quad 0 < i < N, \end{aligned}$$

In particular, $P_2 - P_1 = (q/p)(P_1 - P_0) = (q/p)P_1$ (since $P_0 = 0$) So that, $P_3 - P_2 = (q/p)(P_2 - P_1) = (q/p)^2 P_1$, And hence,

$$P_{i+1} - P_i = \left(\frac{q}{p}\right)^i P_1 \quad 0 < i < N.$$

$$\begin{aligned} P_{i+1} - P_1 &= \sum_{k=1}^i (P_{k+1} - P_k) \\ P_{i+1} - P_1 &= \sum_{k=1}^i \left(\frac{q}{p}\right)^k P_1 \\ P_{i+1} &= P_1 + \sum_{k=1}^i \left(\frac{q}{p}\right)^k P_1 = P_1 \sum_{k=0}^i \left(\frac{q}{p}\right)^k \end{aligned}$$

Then we get,

$$P_{i+1} = \begin{cases} P_1 \frac{1 - (\frac{q}{p})^{i+1}}{1 - (\frac{q}{p})} & \text{if } p \neq q; \\ P_1(i+1) & \text{if } p = q = 0.5 \end{cases} \quad (4.1)$$

Choosing $i = N - 1$ we get and using $P_N = 1$ we get,

$$1 = P_N = \begin{cases} P_1 \frac{1 - (\frac{q}{p})^N}{1 - (\frac{q}{p})} & \text{if } p \neq q; \\ P_1 N & \text{if } p = q = 0.5 \end{cases}$$

Hence we get, $P_1 = 1/N$ when $p = q = 0.5$ and $P_1 = \frac{1 - (q/p)^N}{1 - (q/p)}$ Then,

$$P_i = \begin{cases} \frac{1 - (\frac{q}{p})^i}{1 - (\frac{q}{p})^N}, & \text{if } p \neq q; \\ \frac{i}{N} & \text{if } p = q = 0.5 \end{cases} \quad (4.2)$$

4.2.1 Becoming infinitely rich or getting ruined

If $p > 0.5$, Then $\frac{q}{p} < 1$ from eq. (4.2) we get,

$$\lim_{N \rightarrow \infty} P_i = 1 - \left(\frac{q}{p}\right)^i > 0$$

i.e. if $p > 0.5$, there is a positive probability that the gambler will never get ruined but instead will become infinitely rich as $N \rightarrow \infty$.

If $p \leq 0.5$ then $\frac{q}{p} \geq 1$ thus from eq. (4.2)

$$\lim_{N \rightarrow \infty} P_i = 0, \quad p \leq 0.5$$

i.e. if $p \leq 0.5$ (each gamble is not in his favor), then with probability one the gambler will get ruined $N \rightarrow \infty$.

Example 4.2.1 (Random walk hitting probabilities). Ellen bought a share of stock for \$10, and it is believed that the stock price moves (day by day) as a simple random walk with $p = 0.55$. What is the probability that Ellen's stock reaches the high value of \$15 before the low value of \$5?

What is the probability that Ellen will become infinitely rich?

Solution: Let us define, $P(a) = \mathbf{P}(X_n \text{ hits } a \text{ before hitting } b)$

We want to calculate the probability that the stock goes up by 5 before going down by 5.

By letting $a = 15$ and $b = 5$ and $N = a - b = 10$, we can imagine a gambler who started with \$5 and want to win \$10 before going broke. So we can compute $P(a)$ by casting the problem into the framework of the gamblers ruin problem: $P(a) = P_5$ and $N = 10$. Then,

$$P(5) = P_5 = \frac{1 - \left(\frac{0.45}{0.55}\right)^5}{1 - \left(\frac{0.45}{0.55}\right)^{10}} = \frac{1 - (0.82)^5}{1 - (0.85)^{10}} = 0.73$$

To see Ellen will ever be infinity rich we make $N \rightarrow \infty$ and started gamblers ruin problem at \$10 then,

$$\lim_{N \rightarrow \infty} P_{10} = 1 - (q/p)^{10} = 1 - (0.82)^{10} = 0.86$$

4.2.2 Gambler's Ruin Problem Simulation

We can nicely make a simulation of Gambler's ruin problem And observe the output of it.

Observation:

- When $p = 0.45$ The Gambler will be ruined after 384 games.
- When $p = 0.51$ The Gambler will reach his fortune after 4412 games.

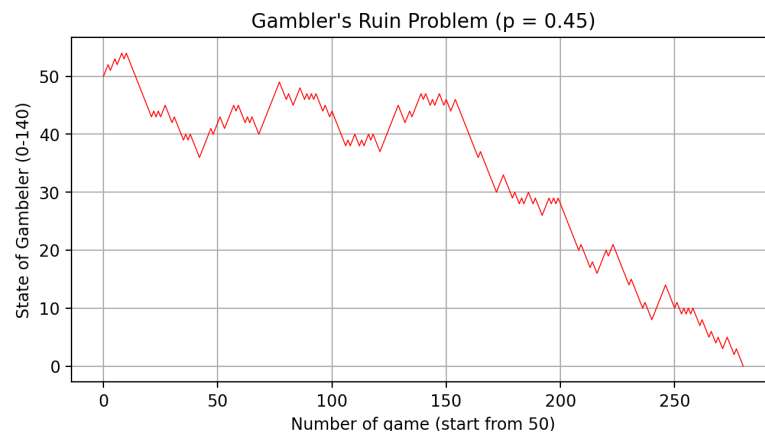


Figure 4.5: Gambler's Ruin Problem with ($p = 0.45$)

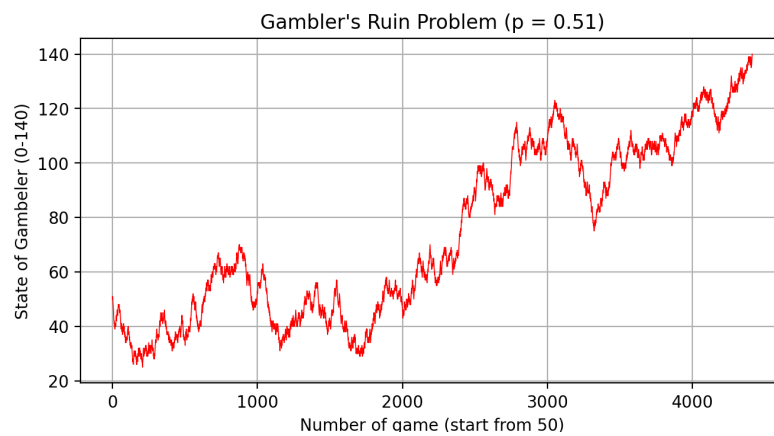


Figure 4.6: Gambler's Ruin Problem with ($p = 0.51$)

Chapter 5

Application of Markov Chain

Markov chain Has numerous application in real life. Let us explore house in this chapter.

5.1 Google PageRank

Before google surfing in the web is very hard, in search engines page was rank by primly on keyword-based algorithms to rank web pages. One popular early search engine, called AltaVista, used a ranking algorithm that focused on keyword matching. It would examine factors like the frequency of the keyword in the page's content, the placement of the keyword (e.g., in titles or headings), and the number of times the keyword appeared in the page's metadata. Keyword-based algorithm is very easy to misuse. A spam page could boost its ranking just by including a long list of words repeated over and over again.

In other hand taking the link structure into account led to dramatic improvements in search engines. As a first attempt, one could rank a page based on how many other pages link to it. That is, if Page A links to Page B, we consider it a “vote” for B, and we rank pages based on how many votes they have. But this is again very open to abuse: a spam page could boost its ranking by creating thousands of other spam pages linking to it. And though it may seem democratic for each page to have equal voting power, an incoming link from a reliable page is more meaningful than a link from an uninformative page.

To solve this problem Sergey Brin and Larry Page develop a algorithm to rank page in their search engine Google and named it after Larry Page Google PageRank, which was introduced in 1998. Google PageRank ranks the importance of a page not only by how many pages link to it, but also by the importance of those pages.

To understand Google PageRank we consider a smaller version of World Wide Web contain only 4 pages connected as shown in fig. 5.1.

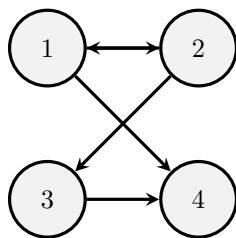


Figure 5.1: A small Web Containing 4 page

Imagine someone randomly surfing the web, starting at some page and then randomly click to get from one page to the next (with equal probabilities for all links on the current page). The idea

of PageRank is to measure the importance of a page by the long-run fraction of time spent at that page.

Of course, some pages may have no outgoing links at all, such as page 4 above (**dangling node**). When the web surfer encounters such a page, rather than despairing he or she opens up a new browser window and visits a uniformly random page. Thus a page with no links is converted to a page that links to every page, including itself. For the example above, the resulting transition matrix is

$$P = \begin{bmatrix} 0 & 1/2 & 0 & 1/2 \\ 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

If we start with probability vector $\mathbf{t} = [1/4 \quad 1/4 \quad 1/4 \quad 1/4]$.
i.e. every page has equal probability to visit in first state. Then the 2nd step probability vector will be $\mathbf{t}_1 = [0.1875 \quad 0.1875 \quad 0.1875 \quad 0.4375]$.
Now doing $\mathbf{t}_2 P$ we get the 3rd step probability vector \mathbf{t}_3 . After repeating this process we get.

$$\mathbf{t}_{100} \approx \mathbf{t}_{101} = [0.2 \quad 0.2 \quad 0.2 \quad 0.4]$$

Hence it is the stationary probability.

In general let M be the number of pages on the web, let P be the $M \times M$ transition matrix of the chain described above, and let π be the stationary distribution. Think of π_j as a measure of how important Page j is. Hence the equation

$$\pi_j = \sum_{i=1}^M \pi_i p_{ij}$$

says that the score of Page j should be based not only on how many other pages link to it, but on their scores.

It is not clear that a unique stationary distribution exists for this chain, since it may not be irreducible and aperiodic. Even if it is irreducible and aperiodic, convergence to the stationary distribution could be very slow since the web is so immense. To address these issues, suppose that before each move, the web surfer flips a coin with probability α of Heads. If Head, the web surfer click to a random link from the current page; if Tails, the web surfer teleports to uniformly random page. The resulting chain has the *Google transition matrix*

$$G = \alpha P + (1 - \alpha) \frac{J}{M}, \tag{5.1}$$

where J is the $M \times M$ matrix of all entries are 1. Note that the row sums of G are 1 and that all are positive, so G is valid transition matrix for an irreducible, aperiodic Markov Chain. Which means that it has unique stationary distribution π , called *PageRank* and the chain converge to it.

The choice of α is an important consideration; choosing α close to 1 makes sense to respect the structure of the web as much as possible, but there is a tradeoff since it turns out that smaller values of α make the chain converge much faster. As a compromise, the original recommendation of Brin and Page was $\alpha = 0.85$.

PageRank is conceptually nice, but computing it sounds extremely difficult, considering that that $\pi G = \pi$ could be a system of 100 equations in 100 billion unknowns. Instead of thinking of this as a massive algebra problem, we can use the Markov chain interpretation: for any starting

distribution \mathbf{t} , $\mathbf{t}G^n \rightarrow \pi$ as $n \rightarrow \infty$. And $\mathbf{t}G$ is easier to compute than it seems at first:

$$\mathbf{t}G = \alpha(\mathbf{t}P) + \frac{1-\alpha}{M}(\mathbf{t}J),$$

where computing the first term is easy to compute as most element P is 0 and computing the second term is very easy $\mathbf{t}J$ is a vector of all 1. To compute $\mathbf{t}G^n$ by $((\mathbf{t}G)G) \dots$. After a very long iteration of n if it appears to converge then we say it is the approximation of PageRank.

For the above example

$$\mathbf{t}_{10000} \approx \mathbf{t}_{10001} = \begin{bmatrix} 0.2004008 & 0.2004008 & 0.2004008 & 0.3987976 \end{bmatrix}$$

Hence it is approximate Stationary Distribution.

5.2 Text Generator

Markov Chain is used in language modelling, text generation, and speech recognition. Google search, Google's Smart Compose on Gmail are just a few examples. Using the probabilistic properties of Markov chains to generate new text based on patterns observed in a given training text.

Here a basic outline of how to build a text generator using the Markov chain:

- **Data Processing:** Start by preparing your training text data. Remove any unnecessary characters or formatting and split the text into individual nodes.
- **Building the Transition Matrix:** Using the given data construct the transition matrix that represents the probability to go from one node to another node.
- **Generating New Text:** To generate random text start with the initial node and using the transition matrix select the next node. Repeating this we are able to generate a text.

Using the complete work William Shakespeare as a sample I am able to make a text generator in python using the concept of Markov chain. Here is some output:

COMMERCIALLY. PROHIBITED COMMERCIAL DISTRIBUTION INCLUDES BY ANY SERVICE THAT CHARGES FOR DOWNLOAD TIME OR USED COMMERCIALLY. PROHIBITED COMMERCIAL DISTRIBUTION INCLUDES BY

Stones have seen More free thee without more bondage happy, my cousin's hands, and is but it will come; for pulling

Dead shepherd, tell you live. But cannot for fashion of more turn their ends Of the worst of thunder heard

5.3 Other Applications

Markov chain has a very wide range of usefulness. It has use cases in many fields like Data Science, AI (Artificial Intelligence), Finance etcetera.

1. **Image and Signal Processing:** Markov Random Field a type of Markov chain, are used in image segmentation, denoising, and object recognition. MRF models capture the spatial dependencies between neighbouring pixels or image regions.

2. **Finance and Economics:** Markov chains are used in modelling financial markets and analysing economic systems. it can be used to model stock prices, interest rates, and exchange rates. Markov models are also used for portfolio optimization and risk analysis.
3. **Queueing Theory:** Markov chains are extensively used in the study of queueing systems. it helps analyse the behaviour of waiting lines, arrival and service processes, and queue lengths. Applications include traffic engineering, telecommunications, and service operations.
4. **Artificial Intelligence and Machine Learning:** Markov chains are utilized in reinforcement learning algorithms, where an agent learns to make decisions based on a sequence of states and actions. Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs) are used in planning and decision-making problems.
5. **Genetics and Molecular Biology:** Markov chains are used to model and analyse biological sequences, such as DNA and protein sequences. it is used in sequence alignment algorithms, predicting protein structures, and identifying hidden patterns in genomic data.

Bibliography

- [1] Jessica Hwang Josepk K. Blitzstein. *INtroduction to Probability*. CRC Press, 2 edition, 2015.
- [2] Fariha Mahfuz. Markov chains and their applications. https://scholarworks.uttyler.edu/cgi/viewcontent.cgi?article=1010&context=math_grad, 2020.
- [3] Shaldon Ross. *A first Course in Probability*. Pearson, 8 edition, 2010.
- [4] Shaldon M. Ross. *Probability Models for Computer Science*. Academic Press, 3 edition, 2006.
- [5] Sheldon M. Ross. *Stochastic Process*. Jhon Wily & Sons, 2 edition, 1996.
- [6] Eugene Seneta. Markov and the creation of markov chains. <https://www.maths.usyd.edu.au/u/eseneta/senetamcfinal.pdf>.
- [7] wikipedia. https://en.wikipedia.org/wiki/Markov_chain.

Appendix A

Random Walk in Python

```
import numpy as np
import matplotlib.pyplot as plt

# define number of steps and starting point

NumberOfStape = 1000
start = 0
p = 0.45 # probability of going up.
Prob = [ p, 1 - p ]
Position = [ start ]
Step = [ 1, -1 ]
for i in range(NumberOfStape):
    rr = np.random.random()
    if rr < Prob[0]:
        position = Position[-1] + 1
    else:
        position = Position[-1] -1
    Position.append(position)

# for plotting Random Walk

fig = plt.figure(figsize=(10,6), dpi=125)
plt.plot(Position, linewidth=0.9)
plt.xlabel("Time")
plt.ylabel("State of Random Walk")
plt.title("1D Simple Random Walk for p = 0.45")
plt.grid(True)
plt.show()
```

Figure A.1: Random Walk python script

Appendix B

The Gambler's Ruin Problem

```
import numpy as np
import matplotlib.pyplot as plt

S_AMOUNT = 50
W_AMOUNT = 140

I_Amount = S_AMOUNT
NumberOfGame = 100000

Amount = [ I_Amount ]
p = 0.45 # probability of winn
Prob = [p,1-p]

t = 0
while NumberOfGame > 0:
    t += 1
    rr = np.random.random()
    if rr < Prob[0]:
        I_Amount += 1
    else:
        I_Amount -= 1

    Amount.append(I_Amount)

    if I_Amount == 0:
        break
    if I_Amount == W_AMOUNT:
        break
    NumberOfGame -= 1

print(f"Number of game: {t}")
print(f"Total Gain: {I_Amount}")

# Graph Plot,
fig = plt.figure(figsize=(8,5),dpi=200)
plt.plot(Amount, color='red', linewidth=0.6)
plt.xlabel(f"Number of game (start from {S_AMOUNT})")
plt.ylabel(f"State of Gambeler (0-{W_AMOUNT})")
plt.title(f"Gambler's Ruin Problem (p = {p})")
plt.grid(True)
plt.show()
```

Figure B.1: Gambler Ruin problem python script

Appendix C

Google PageRank

```
ALPHA = 0.85

import numpy as np

def PageRank(TransitionMat, InitVector, NumbOfIt, M):
    ti = InitVector
    J = np.full((M,M),1)
    aJ = np.multiply((1-ALPHA)/M,J)
    aTransitionMat = np.multiply(ALPHA,TransitionMat)
    G = np.add(aTransitionMat,aJ) # Google transition matrix.
    for _ in range(NumbOfIt):
        ti = np.matmul(ti,G)
    return ti

P = np.matrix([[0, 1/2, 0, 1/2], [1/2, 0, 1/2, 0], [0, 0, 0, 1], [1/4, 1/4,
    1/4, 1/4]])
t0 = np.matrix([1/4,1/4,1/4,1/4])

t10000 = PageRank(P, t0, 10000, 4)
t10001 = PageRank(P, t0, 10001, 4)
t10002 = PageRank(P, t0, 10002, 4)
t10003 = PageRank(P, t0, 10003, 4)
t10004 = PageRank(P, t0, 10004, 4)

print(f' t_10000 = {t10000}')
print(f' t_10001 = {t10001}')
print(f' t_10002 = {t10002}')
print(f' t_10003 = {t10003}')
print(f' t_10004 = {t10004}')
```

Figure C.1: Google PageRank Small simulation

```
t_10000 = [[0.20618557 0.20618557 0.20618557 0.3814433 ]]
t_10001 = [[0.20618557 0.20618557 0.20618557 0.3814433 ]]
t_10002 = [[0.20618557 0.20618557 0.20618557 0.3814433 ]]
t_10003 = [[0.20618557 0.20618557 0.20618557 0.3814433 ]]
t_10004 = [[0.20618557 0.20618557 0.20618557 0.3814433 ]]
```

Figure C.2: OutPut of Google PageRank Small simulation

Appendix D

Text Generator

```
import numpy as np
import json

def make_pair(list):
    for i in range(len(list) - 1):
        yield (list[i], list[i+1])

text = open('D:/coddin_g_and_programming/Python/program/markov_chain/
    shakespeare_mod.txt', 'r').read()
text_words = text.split()

pairs = make_pair(text_words)

words_dist = {}
for word_1, word_2 in pairs:
    if word_1 in words_dist.keys():
        words_dist[word_1].append(word_2)
    else:
        words_dist[word_1] = [word_2]

# with open('D:/coddin_g_and_programming/Python/program/markov_chain/dist.
    json', 'w') as f:
#     json.dump(words_dist, f)

first_word = np.random.choice(text_words)

while first_word.islower():
    first_word = np.random.choice(text_words)

chain = [first_word]

n_word = 20

for i in range(n_word):
    chain.append(np.random.choice(words_dist[chain[-1]]))

print(' '.join(chain))
```

Figure D.1: Text Generator using markov chain