



Review of Markov Chain and Its Applications in Telecommunication Systems

Amel Salem Omer and Dereje Hailemariam Woldegebreal^(✉)

School of Electrical and Computer Engineering, Addis Ababa University, Addis Ababa Institute of Technology, Addis Ababa, Ethiopia

amel.salem@yenepay.com, dereje.hailemariam@aait.edu.et

Abstract. Markov chain is a powerful mathematical tool that is used to predict future state of a random process based on its present state, for classical or first-order Markov chain, and past states for higher-order Markov chain. Markov chain has a wide range of applications in various fields of science and technology. To mention some in the area of telecommunication systems: Internet page ranking; Internet traffic modeling; language source modeling in natural language processing for text compression and text generation applications; speech recognition; wireless channel modeling; spectrum occupancy prediction for cognitive radio; user mobility modeling, handover management, and operation status monitoring in cellular mobile networks; network service and maintenance optimization; and in Markov chain Monte Carlo simulation methods. The main objective of this paper is to review fundamental concepts in Markov chain for discrete sources with emphasis on its application in telecommunication systems. It introduces terminologies, method to compute transition probabilities from real data, computing different states of the chain, and possible applications areas. The focus is given for both classical and higher-order Markov chain as well as Hidden Markov models. While acknowledging a related lecture note published in 2010, which we came to know lately, this review includes additional topics, such as higher-order Markov chain and Hidden Markov models, and tries to present core ideas in less mathematically rigorous but more practicable way. We hope that interested researchers who wish to apply Markov chain for various applications will benefit from this review.

Keywords: Hidden Markov model · Initial distribution · Markov chain · Modeling · Prediction · State · Steady state distribution · Transition probability

1 Background

Markov chain is a class of stochastic processes used in several fields of science and engineering to model real-world and randomly-varying systems or phenomena. Markov chain assumes that real life systems are made of discrete and countably finite *states*, which could be values of a random variable at different time t , situations of the phenomena or output of the system [1–7]. The set of states form a *state space*. The system is assumed to randomly transition (evolve or move) in *discrete time* from one state to another state

according to a certain *transition probability*. The collection of transition probabilities forms the transition *probability matrix*.

Markov chain is formulated to *predict*, in probabilistic sense, an immediate future state of the system based on the present state only, and not on past states. This type of Markov chain is sometimes referred as classical or first-order Markov chain, while the dependency on the current state only is characterized as “*memorylessness*” or *Markov property*. Higher-order Markov chain, on the other hand, includes the past states, in addition to the present state for the prediction of the future state. In natural language processing area, this dependency in the current and past states is called the *N*-gram approach, and the resulting model is called an *N*-gram model. The *N* in *N*-gram is the order of the Markov chain. The *N*-gram model improves the prediction accuracy compared with predication with one order dependency only [24, 26], and [28]. With knowledge of the future state, Markov chain iteratively estimates the states of the system after certain number of transitions in the future. In doing so, Markov chain is used to understand the dynamics of the system, as the system transitions from one state to another state.

In many real world problems, however, the states we are interested in are hidden or *non-observable*. In such cases, Hidden Markov models are used instead of Markov chain. In this modeling approach, the assumption is the presence of other observable parameters (states) that are related to the hidden (or non-observable) states with some probability function.

Nowadays Markov chain [1–8] and Hidden Markov chain [29, 30] are used to capture behavior of a wider range of real-world systems and phenomena. Application examples related to this review include: source modeling in natural language processing that includes text compression [28] and text generation [31–33]; speech recognition [29, 30]; data compression (like LZMA in 7zip); Internet traffic modeling [36]; Internet page ranking as in the PageRank used by Google search; spam filtering to determine legitimacy of text; and in stochastic simulation methods known as Markov chain Monte Carlo simulation methods. Moreover, Markov chains are used for channel modeling [23–27]; user mobility modeling in mobile systems [12–16]; handover management in mobile systems [17]; spectrum occupancy prediction for cognitive radio [18–22]; accessibility and retainability modeling in mobile networks [34, 35]; to implement proactive maintenance procedures [37] and the likes.

Navigating through literatures written in the area of Markov chain, one groups of literatures focus on fundamental concepts with less emphasis on real-world applications. The other groups are more on the application areas with lesser focus on fundamental concepts. An exception to this is the lecture note in [11] that discusses performance modeling of communication networks with Markov chain. However, the lecture note fails to address topic topics such as higher-order Markov chain and Hidden Markov models. The intention of this review paper is to bridge the gap between fundamental concepts and application areas. It introduces basic ideas behind Markov chain, both for first and higher order, and how they can be used to model real-world problems, specifically the once that we encounter in telecommunication systems.

The rest of the paper is organized as follows. Section 2 presents a couple of probability concepts relevant while explaining application of Markov chain. Markov chain for discrete sources is explained in Sect. 3. Basic terminologies that are covered in this section are: states transition probabilities and transition matrix; initial and steady-state probability distributions; computing transition probabilities from a data; verifying if a given systems fulfills the Markov property. Section 4 briefly presents Hidden Markov model. Section 5 summarizes possible application areas of Markov chain, with concluding remarks given in Sect. 6.

2 Review of Probability Theory

In this section, we review few concepts from probability theory relevant to understand Markov chain.

2.1 Law of Large Numbers for Markov Chain

In probability theory, the *law of large numbers* is a theorem that describes the result of performing the same experiment a large number of times [38]. According to the law, the average of the results obtained from a large number of trials repeated independently should be close to the expected value, and will tend to become closer to the expected value as more trials are performed. The law of large numbers is important because it guarantees stable long-term results for the averages of some random events.

The steady state or equilibrium state of Markov chain, which is the limit of observing the system over larger iterations, is analogous to performing the same experiment a large number of times. In a way, because of its steady state distribution, systems that obey Markov chain also comply with the law of large numbers.

2.2 Chain's Rule

In probability theory, the *Chain rule* (also called the general product rule) permits the calculation of any member of joint distributions of a set of random variables using only conditional probabilities [28]. The rule is useful in the study of N -gram modeling and Bayesian networks, which describe probability distribution in terms of conditional probabilities. Suppose that we have a set of random variables (observation) designated as S_1, S_2, \dots, S_h , then the probability of occurrence of the observation is:

$$\begin{aligned} P(S_1, S_2, \dots, S_h) &= P(S_1) \cdot P(S_2|S_1) P(S_3|S_2, S_1) \dots P(S_h|S_{h-1}, \dots, S_1) \\ &= \prod_{i=1}^h P(S_i|S_{i-1}, \dots, S_2, S_1) \end{aligned} \quad (1)$$

In higher-order Markov chain or N -gram Markov chain, smaller conditional probabilities make computation using Eq. (1) difficult. Hence, an approximation to the Chain rule in Eq. (1) is:

$$P(S_1, S_2, \dots, S_h) \approx \prod_{i=1}^h P(S_i|S_{i-1}, \dots, S_{i-N}) \quad (2)$$

In practice, frequency counting method is used to compute the conditional probabilities in Eqs. (1) and (2). For example in natural language processing, the random variable S_i could represent letters or words in a language. To compute the probability of each S_i and a combinations of it, a large corpora i.e., a collection of texts written with the language's alphabet and containing all the possible letters or words sequences is organized to count needed probabilities [28]. Using this count method, the conditional probability in Eq. (2) can be calculated for the N -gram model as:

$$P(S_1, S_2, \dots, S_h) = \frac{\text{count}(S_1, S_2, \dots, S_h)}{\sum_i \text{count}(S_1, S_2, \dots, S_h)} \quad (3)$$

The $\text{count}(S_1, S_2, \dots, S_h)$ in the numerator is count of the desired h letters/words while count in the denominator is all counts of possible combinations of sequences of h letters/words in the corpora. In the discussion of higher-order Markov chain in SubSect. 3.4, we will see variant of Eq. (2).

3 Markov Chain for Discrete Sources

Andrey Markov first introduced Markov chain in the year 1906 [1]. He explained Markov chain as special classes of *stochastic process/system with random variables designating the states or outputs of the system, such that the probability the system transitions from its current state to a future state depends only on the current state*, and it is independent of the series of states that preceded it [1, 11]. This is unlike most studies in stochastic process that are based on the independent and identically distributed (i.i.d.) assumption, which is like saying the future state is independent of any other prior states, or the process every time “resets” itself into fresh. Hence, Markov chain is a sort of compromise between systems whose states have complete or full dependency, which is difficult to model, and i.i.d. systems.

Over the years, the dependency of future state on present state only has been modified to model sources where the future state depends on $N - 1$ past and present states. This gives rise to the higher-order Markov chain or an N -gram model. Whatever the order, Markov chain is used to predict, in probability sense, about the future states of the random variable (or stochastic process), based on current and previous states. This section sections briefly explains the Markov chain and higher-order Markov chain.

3.1 First-Order Markov Chain

Markov chain is a kind of stochastic model used to model randomly varying systems, like information sources, whose future output has dependency only on current state only. This dependency on the current state only is sometimes characterized as “*memorylessness*”. Key issues in applying Markov chain for a certain application require identifying and computing state, state space, initial (state) distribution (or initial state vector), transition probability, transition probability matrix, the state probability after certain iterations, and equilibrium or steady state distribution. These terms are briefly explained below.

To better explain Markov chain using an example from a mobile system, consider a mobile system with three basestations and a single user, as shown in Fig. 1 (a) below.

Let BS_i for $i \in \{1, 2, 3\}$ designate the three basestations and let the current location of the user is recorded when a user receives a call. Observing the record over a larger time interval every time a new call arrives provides information about mobility pattern of the user, which are the user is likely to stay in the same cell or move to either of the two cells.

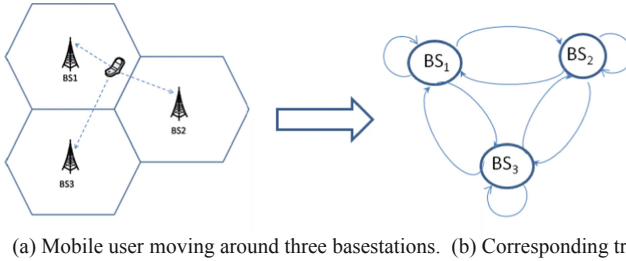


Fig. 1. Mobility of a mobile user in a cellular system.

State in Markov Chain: Markov chain assumes that real life systems are made of *states*. The states could also be outputs or values of the random process. The state space, i.e., the set of all possible states, is discrete (usually) and countable (finite). In discrete Markov chain the system randomly transitions in discrete time from one state to another state according to certain *transition probability*.

- In the mobility example shown in Fig. 1, state could be the basestation the user is in when a call arrives. The current state is the basestation in which the user is connected when a call arrives. Accordingly, the future state is the basestation the user will visit (or will be in) when the next call arrives.
- If we take information source modeling for the purpose of text compression as a second example, each letter (or group of letters) generated from an information source can be viewed as a state, and the state space contains all possible letters (or group of letters) generated from the source [28]. Every time the information source generates a letter, we can think of the chain is moving/transitioning in discrete time and randomly from one state (in this case a letter) to another state to make meaningful words and sentences.

Transition Probability: indicates the probability or rate with which the chain transitions in discrete time from one state to another state. The transitions capture hidden patterns in the system, e.g., in the mobility example explained above the transition indicates the rate or frequency with which the user moves from one basestation (area) to another basestation (area). There are multiple factors dictating the user's mobility. For the text generation example, transition from one letter to the next letter in a text is used as an approximation of the true underlying nature of the language considered.

To have a generalized formulation of the transition probability, suppose that we have a discrete-time and random source having N states and the state space designated as

$\{a_0, a_1, \dots, a_{N-1}\}$. Moreover, let the sequence of states generated in time are given as $S_1, S_2, S_3, \dots, S_n, \dots$, where $S_n \in \{a_0, a_1, \dots, a_{N-1}\}$ and n in S_n indicates the discrete time index. For Markov chain fulfilling the memoryless assumption, the transition probability is expressed as:

$$\begin{aligned} P(S_{n+1} = a_j | S_n = a_i, S_{n-1} = a_h, S_{n-2} = a_g, \dots) \\ = P(S_{n+1} = a_j | S_n = a_i) \end{aligned} \quad (4)$$

We learn from the Markov property that only the most recent state (information) matters to predict the next or future state. Alternatively, the past and the future are conditionally independent given the present. From now on, the transition probability from state a_i to state a_j is designated as:

$$P_{ij} = P(S_{n+1} = a_j | S_n = a_i) \quad (5)$$

Transition Probability Matrix: collection of the transition probabilities P_{ij} forms the *probability transition matrix*, P . The transition probability matrix shows all possible state transitions. Each element of the matrix represents the probability that the system switches its state to or remains in the same state. Every state in the state space is included once as a row and again as a column [11]. P is a square matrix whose order is the same as the number of states.

Reconsidering the mobility modeling discussed in Fig. 1, and assuming the following observations are made after recording the mobility patter of the user.

- If the user is in BS_2 , it is less likely to stay in the same cell when the next call arrives; moreover, it is just as likely for the user to be in BS_1 as in BS_2 .
- If the user is either in BS_1 or BS_3 , it has an even chance of staying in the same basestation when the next call arrives.
- If there is change from BS_1 or BS_3 , only half of the time is this a change to BS_2 .

With this information, the transition matrix is formed as follows:

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{pmatrix} = \begin{pmatrix} 0.50 & 0.25 & 0.25 \\ 0.50 & 0.00 & 0.50 \\ 0.25 & 0.25 & 0.50 \end{pmatrix} \quad (6)$$

The entries in the first row of the matrix P in Eq. (6) represent the probabilities for the mobility of the user following that it is in BS_1 . Similarly, the entries in the second and third rows represent the probabilities for the user location following that it is in BS_2 and BS_3 , respectively. In general, an $n \times n$ transition probability matrix P must satisfy the following properties [5].

- Each transition probability is non-negative, i.e., $P_{ij} \geq 0$ for all i and j .
- For all i and j , the summation of all transition probabilities in a row must be equal to one, i.e.,

$$\sum_{j=1}^n p_{ij} = 1 \quad (7)$$

Transition Diagram: The transition probability matrix could also be represented using a transition diagram. Transition diagrams are used to represent all transitions in the system. Figure 1. (b) shows the transition diagram for the mobility example. Each node represents the state of the Markov chain and a directed arrow is used to represent the presence of transition from one state to another. The edge represents the current state and the arrow indicates the next state [29].

Initial (Probability or State) Distribution: The initial state distribution is usually expressed as a probability distribution vector, whose entries indicate probability that the system will be in a given state at a given *initial* time. Let U be a $1 \times n$ dimension vector, given in Eq. (8) designating the initial distribution. Each entry of the vector is non-negative and represents the probability that the chain or system is in that state (or starts in that state) at a given (initial) time instant.

$$U = [P_1, P_2, \dots, P_n] \quad (8)$$

All elements of the vector should sum to one. In the absence of clear knowledge on the initial distribution, the system can be assumed to be in one state with certainty, so that all entries of the vector except the entry for that state will have zero values. Alternatively, we can assume that all states are equally likely so that $P_i = 1/n$ for $i \in \{1, 2, \dots, n\}$.

Steady State Distribution – One of the interesting things about systems that obey Markov chain is that, from which ever initial states the chain starts from, the chain will converge to *steady state, stationary, equilibrium or static* distribution after sufficiently large iterations/transitions. Steady state indicates there exists a condition where regardless of the current state, the probability for the next state is always the same. In steady state, however, does not mean that the chain stop from transitioning; rather, from a given state the probability of outward transitions and inward transitions will be the same (or is in equilibrium) so that the probability distribution stays same.

With knowledge of the transition matrix P and the initial probability vector U as defined in Eq. (6) and Eq. (8), respectively, the probability distribution of the chain after k transitions in the future is given by [2].

$$U^{(k)} = UP^k \quad (9)$$

P^k is the result of multiplying P k times by itself. Each element of $U^{(k)}$, designated as $P_{ij}^{(k)}$, is the probability of going from state i to state j in k iterations. As we keep iterating through state transitions by applying P^k , the probability vectors $U^{(k)}$ converge to some fixed value, say $\pi_{(k)}$. That is called the *steady state distribution*.

$$\lim_{k \rightarrow \infty} U^{(k)} = UP^k = \pi_{(k)} \quad (10)$$

We note from Eq. (10) that based on knowledge of the state space, initial distribution and transition matrix, Markov chain can predict/estimate, in probabilistic sense, the future state of the system after certain transitions in time.

Regular Markov Chain: is one type of Markov chain, where after long transitions, the states will become independent of the initial states. For this type of chain, it is true that long-range predictions are independent of the starting state. Considering again the mobility example, the transition matrix after the second and six transitions are given in Eq. (11) below. Note that after the sixth transition, the matrix will remain the same, indicating the resulting state vector will be the same.

$$\begin{aligned}
 P^{(2)} &= \begin{pmatrix} 0.438 & 0.188 & 0.375 \\ 0.375 & 0.250 & 0.375 \\ 0.375 & 0.188 & 0.438 \end{pmatrix} \\
 \dots & \\
 P^{(6)} &= \begin{pmatrix} 0.400 & 0.200 & 0.400 \\ 0.400 & 0.200 & 0.400 \\ 0.400 & 0.200 & 0.400 \end{pmatrix}
 \end{aligned} \tag{11}$$

Note that whatever what the initial distribution vector U is, as an example $U = [1, 0, 0]$ so that the user is in BS_1 initially, the probabilities that the user will be in BS_1 , BS_2 and BS_3 are 0.40, 0.20, and 0.04. This distribution will be the same no matter where the chain started or what the initial distribution is.

3.2 Additional Properties of Markov Chain

Communicating States: Two states are called communicating states, if state i is accessible from state j and state j is accessible from state i . In other words, it is possible to travel from state i to state j with some probability P_{ij} greater than zero, and it is possible also to travel from state j to state i with some probability P_{ji} with value greater than zero [1–8].

Absorbing State: is a state where once the system entered this state, it cannot leave the state. The system will loop and stay at the state [32]. In case of an insurance system modeled as a Markov chain, once people withdrawn their insurance or if the insured is dead, the next state will stay withdrawn and cannot be active again [32].

Transient State: a state is said to be a transient state if, upon entering this state, the process might never return to this state again.

Recurrent States: is a state if, upon entering this state, the process definitely will return to this state again. Therefore, recurrent state is a state that is not transient [46].

Irreducibility: The Markov chain is called irreducible if it is possible to move from any state i to any state j regardless of the number of transitions it takes. Or we call the Markov chain irreducible if all the states of the chain are in one communication class.

Aperiodic: The Markov chain is called aperiodic if, after departing from state i , the system can return to it through following different paths. Otherwise, it is called periodic, and the system returns to its initial state after some multiplication of period m , where m is greater than 1. All states in a same class could be either periodic or aperiodic and will have the same period.

Positive Recurrent: The Markov chain is called positive recurrent if there is a non-zero probability that after departing from state i , the process will return to it in a finite time.

Ergodicity: happens if all states are aperiodic and positive recurrent.

Reversibility: Markov Chain is said to be reversible if the Markov process changes from one state to another in one direction same number of times as in the reverse direction.

3.3 Computing Transition Probabilities from Data

Estimation of future states via Markov chain typically needs starting values for model parameters, i.e., initial probability distribution and transition matrix [29]. For the initial distribution if the system under consideration is small and simple, any random starting values will usually do. With large and complex systems, good starting values are needed for finding the optimal solution in a reasonable time.

The transition matrix, on the other hand, can be estimated by a Maximum Likelihood approach [29], [43]. Assuming that we have a system having N states designated as $\{S_1, S_2, \dots, S_N\}$. Let n_{ij} is the number of transitions from state S_i to state S_j and n_i is the total number of transitions from state S_i . The Maximum-likelihood estimates of the transition probability P_{ij} is

$$P_{ij} = \frac{n_{ij}}{n_i} \quad (12)$$

Moreover,

$$\sum_{j=1}^N n_{ij} = n_i \quad (13)$$

And

$$\sum_{j=1}^N n_{ij} = n_i \quad (14)$$

Table 1, below how the transition probabilities may be represented by the following table.

3.4 Higher-Order Markov Chain

As mentioned in previous sections, there are sources where the future state depends on $N - 1$ past and the present states. This leads to the name *higher-order Markov chain* or *N-gram model* [28]. As an example in natural languages, because of hidden structures in the language and grammatical rules, the likelihood of a letter generated from a source is dependent on previously generated $N - 1$ contiguous sequence of letters. [In most natural language processing studies, the N -gram indicates sequence of meaningful words instead of letters.]

Table 1. Table for transition probabilities computation.

State	1	2	...	N	Row Sum
1	n_{11}	n_{12}		n_{1m}	n_1
2	n_{21}	n_{22}		n_{2m}	n_2
.
.
N	n_{m1}	n_{m2}		n_{mm}	n_m

We can think of N -gram as sequence of N letters, and a statistical model assigns probabilities to the sequences of letters. Common values of N have special names: a 1-g (or unigram) is a one-letter sequence of letters; a 2-g (or bigram) is a two-letter sequence of letters; and a 3-g (or trigram) is a three-letter sequence of letter and the likes. The N -gram model assigns probabilities to letters and sequence of letters, considering the dependency among the letters. As an example, if we have three sequence of letters S_1, S_2, S_3 where $S_i \in \{a_0, a_1, \dots, a_{N-1}\}$ drawn from the alphabet \mathcal{A} . If we have a sequence containing three letters, the N -gram model assigns probability to the sequence as:

$$P(S_1 S_2 S_3) = P(S_1)P(S_2|S_1)P(S_3|S_1 S_2) \quad (15)$$

Unigram Model: writes the likelihood in Eq. (15) in terms of the probabilities of the three letters as:

$$P_{uni}(S_1 S_2 S_3) = P(S_1)P(S_2)P(S_3) \quad (16)$$

One way to estimate the probability in (16) is through the relative frequency count approach mentioned in Sect. 2.2 above, by taking a substantially large corpus. If M is the total number of three-letter sequences count in the corpus, the maximum likelihood estimate of the sequence $S_1 S_2 S_3$ is computed from the number of counts as:

$$P_{uni}(S_1 S_2 S_3) = \frac{\text{count}(S_1 S_2 S_3)}{M} \quad (17)$$

N-gram Model: in a case that the sequence has large number of letters, getting a good estimate for the probability of the sequence is a bit difficult, as the training corpora may not be sufficient enough to bring out all the possible combinations of letter sequence in the source [4]. One approach to simplify the probability estimate is to combine chain rule and the N -gram assumption, so that probability of observing sequence of letters S_1, \dots, S_h is approximated as [28]:

$$P(S_1, \dots, S_h) = \prod_{i=1}^h P(S_i|S_1, \dots, S_{i-1}) \approx \prod_{i=1}^h P(S_i|S_{i-(N-1)}, \dots, S_{i-1}) \quad (18)$$

Given a large corpora, the maximum likelihood estimation of the conditional probabilities in Eq. (18) are computed via the frequency count as:

$$P(S_i|S_{i-(N-1)}, \dots, S_{i-1}) = \frac{\text{count}(S_{i-(N-1)}, \dots, S_{i-1}, S_i)}{\text{count}(S_{i-(N-1)}, \dots, S_{i-1})} \quad (19)$$

4 Hidden Markov Models

One inherent assumption in the Markov chain includes: the random variables, resulting from state transitions, are observable and transition probabilities can be computed directly from the observation. In many real world problems, however, the states we are interested in are hidden or non-observable [9, 10].

Hidden Markov Models are probabilistic models that are applied when states are not directly observable or when the states are hidden; the model assumes the presence of other observable parameters (emissions) that are related to the hidden states with some probability function. The observable parameters or symbols are also called emissions [29, 30]. In the Hidden Markov Model, the hidden states of the system being modeled are assumed to be of a Markovian process. While parameters of interest in observable Markov chain are transition matrix, P , and initial-state probabilities, U , Hidden Markov model requires emission probabilities, say E , as a third and additional parameter.

Application areas for a Hidden Markov model include bio-informatics, linguistic analysis, mobility analysis, healthcare informatics, and other sequence and pattern recognitions. In telecom related areas, communication channel classifications and analysis, network traffic identification are observed.

To better illustrate the difference between Markov chain and Hidden Markov model, let us consider an example from [29]. Let annual temperature, which can be expressed as “hot state” or as “cold state”, is our parameter of interest. In the case of Markov chain, direct measurement of the temperature is required to predict how the temperature will look like in upcoming days or years.

In the case of Hidden Markov model, say if size of a tree growth (tree ring sizes) is only observed (measured), so that temperature is hidden state. Assuming correlation between the size of tree growth and temperature, Hidden Markov model predicts the temperature based on the observable tree ring sizes. The observable evidences are tree ring sizes as small, medium and large. Therefore, here the states are hidden but there is other correlated observation, the size of tree growth.

Considering another example, where a mobile network operator monitors status of its network. Assume that the operator labels the network status as in “Good” state or “Bad” state, based on some key performance indicator (KPI) values. As in the case of Markov chain, the initial distribution and transition matrix can be directly calculated from observed transitions between the states, i.e., count the observed status, “Good” or “Bad” and calculate the transition probabilities using Maximum Likelihood estimation approach mentioned in Sect. 3.3.

Now the question is, what if the KPI values that are not measurable directly? Or what if the available information is the form of other observations, say peoples’ attempt to

connect disconnect per unit time, average connected time and/or number of connected users per cell? The network status is somehow correlated with the probabilities of these observations. We can use all the three other observations, or just one of them, to make inference about the network status. In such a case where we do not have the direct status or KPI information, we can use Hidden Markov model.

Let us take the first one which is people’s attempt to connect disconnect per unit time. Let us also have prior knowledge that if there is higher attempt to connect disconnect per unit time, there is a high probability for the network to be in “Bad” status and vice versa. The network status can also be somehow at an average performance in between the states. Now we can quantize the attempt to connect disconnect per unit time in to three parts and assign probabilities. Let the quantized attempts per unit time be 10, 5 and 2. Based on available evidences, the emission probabilities for each symbol from the Good state is 0.2 for 10, 0.3 for 5 and 0.5 for 2. And the emission probability for each symbol from the Bad state is 0.5 for 10, 0.3 for 5 and 0.2 for 2.

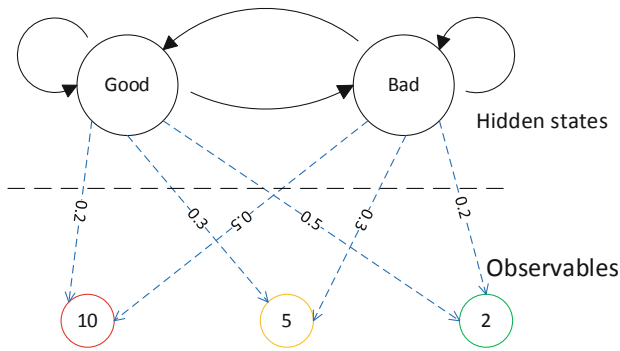


Fig. 2. Hidden Markov model for the mobile network example.

In summary:

- 10, 5, 2 are the observation sequences, O , that can occur randomly in any state;
- E is the Hidden Markov model parameter that relates the observation symbols with the hidden states. E is developed from the available evidence as E_0 .
- Therefore, E_0 will be initialized as;

$$\text{Bad} = \{10: 0.5, 5: 0.3, 2:0.2\}$$

$$\text{Good} = \{10: 0.2, 5: 0.3, 2:0.5\}$$

In Fig. 2 the emission probabilities are those symbolized by the downward blue arrows.

- U is the initial-state probabilities and will be fairly initialized, usually equally distributed as $U_0 = [0.5, 0.5]$.
- The transition matrix P is also fairly initialized, usually equally distributed as $P_0 = ([0.5, 0.5], [0.5, 0.5])$

- Once we determine and initialize those parameters, the Hidden Markov model will be trained with a list of observation datasets. This dataset contains the observation symbols. And in this example, the observation datasets will be different combinations of 10, 5, 2.

Additional considerations regarding Hidden Markov is, since the model tries to capture more information, the model parameters should be optimized. Otherwise we might end up in a computationally costly model. One major area to do the optimization is the emission probability matrix E . Its size is the number of states by the number of observations. Therefore, for a given number of states, we need to limit our observations to only the important ones.

5 Applications of Markov Chain

The purpose of this section is to provide summary of potential application areas of the Markov chain and Hidden Markov model in the context of telecom systems.

5.1 Markov Chain for Natural Language

Probabilistic Language Source Model: A language source is a source that randomly generates sequence of letters drawn from the language's alphabet, which is finite in number. Application areas that require language modeling include text compression and automated text generation. Each letter has a certain probability of occurrence, which depends on the language's structure, context and other factors [28]. Group of letters from the sequence form meaningful word, where group of words form sentences and paragraphs.

A probabilistic or statistical language source model assigns a probability distribution to letters of the alphabet, words of the language, or some sequence of letters which may not be meaningful. Given such a sequence of letters, say of length m , it assigns a probability $P(l_1, l_2, \dots, l_m)$ to the whole sequence.

A Markov language source is a language source whose underlying dynamics are governed by finite Markov chain i.e., having a finite alphabet, dependency of a given letter on a sequence is limited to previous N letters, and a stationary probability distribution for possible sequence of letters.

In Markovian language sources, letters, words of sequence of letters are equivalent to states of the Markov chain. The source is assumed to emit sequence of letters or group of letters, as in text compression application [28] or sequence of words as in text generation application [31-32] to create words, sentences and paragraphs. Every emission can be considered as if the chain is transitioning from one to another state.

The usual assumption in classical Markov chain is the dependency among sequence of letters (or words) is in the first degree, i.e., the likelihood of a letter (or a word) is dependent on the likelihood of a letter (or a word) generated one time step earlier only. In natural languages, because of hidden structures in languages and grammatical rules, the likelihood of the letter (or the word) is dependent on previously generated N sequence of co-occurring letters (or words); leading to the name N -gram language model. Hence,

the N -gram model is a type of probabilistic language model for predicting the likelihood of the next letter (or word) from a sequence of $(N - 1)$ previously generated letters (or words). Note that $N - 1$ is the order of the Markov chain. Common values of N have special names: “*unigram*” for $N = 1$, “*bigram*” for $N = 2$, “*trigram*” for $N = 3$, “*four-gram*” for $N = 4$, “*five-gram*” for $N = 5$, and so on. Note that in the N -gram model, the number of states in the Markov chain will be N^2 unlike N in *unigram* model.

The N -gram model probabilities are computed from frequency counts of all possible letters (or words) from a document. However, increasing N is conceptually increasing the alphabet size of the source. A large training corpora/text is needed to bring out all possible combination of message sequences in a language [28]. This approach has a problem when confronted with any N -grams that have not been explicitly seen before. Instead, some form of smoothing is necessary, assigning some of the total probability mass to unseen words or N -grams. Various methods are used, from simple “add-one” smoothing (assign a count of 1 to unseen N -grams, as an uninformative prior) to more sophisticated models, such as Good-Turing discounting or back-off models [28].

5.2 Markov Chain for Automatic Text Generation

The use of computer programs to generate texts of a language, say English, is an active research topic. Automated text generation system generate a predictive text based on previously typed term(s) or word(s). A typical example of text generation is Google’s autocomplete system when we type in the search tab of Google when composing emails [31–33]. Other applications of text generation include: to creating artificial texts or scripts to generate artificial news, create a better movie synopsis, or create poems [31–33]. Text generation is also used in Chatbots for answering frequently asked question) and document summarization (extracting important info based on the model), Language translation.

Markov chain is used for text generation, and as compared to other text generation schemes has the advantages of being simple and easy to implement, and lower computation time [46]. However, the generated text is as good as the input corpus used for training, i.e., generating the probability distribution matrix and the order of the N -gram Markov Chains (high order model) used to capture the context [31–33].

5.3 Spectrum Occupancy Prediction for Cognitive Radio

The radio spectrum is a key and scarce natural resource used to provide a wide variety of wireless services. Traditionally, radio spectrum is exclusively allocated to users and strictly regulated to prevent interference among users using same or adjacent frequency bands. Emergence of new wireless technologies and innovative services has raised the demand for more radio spectrum bands.

One way to avail more radio spectrum is to apply spectrum sharing concepts to harvest underutilized bands of the spectrum. Dynamic Spectrum Access is an optimal spectrum sharing solution to opportunistically allow secondary users (or unlicensed) to shares bands that are allocated for primary (or licensed) users in a non-interfering manner [22–24]. To avoid the interference the secondary users, among others, should have spectrum sensing capability (i.e., to detect the presence of the primary user that

uses the spectrum), decide if the spectrum is in usage, use the spectrum when free and evacuate the licensed spectrum if needed by the primary user (called spectrum mobility).

Spectrum prediction is one spectrum sensing approach in which the future state of the spectrum is forecasted on the bases of the historical information available about the spectrum. Secondary users within the cognitive radio network, collect sensing information, and utilize statistical correlation, to infer possible future states of the primary user usage patterns [18]. Parameters studied by channel prediction broadly target channel availability i.e., prediction of the channel status as idle, or busy from the perspective of secondary users, as well as, duty cycle i.e., prediction of the average fraction of time the primary user is occupying the channel [19].

In the Markov chain-based spectrum occupancy predication, the spectrum shared by the primary and the secondary users is modeled as a two state (hidden) Markov chain; in which the states are the spectrum being “occupied/busy” and spectrum is “free/idle” [18–22]. As the traffic demand by the primary user is dynamic and random, there is a transition for the channel from “idle” to “busy” state; a similar state transition occurs “busy” to “idle” occurs when the primary user stops using the spectrum. Sometimes “0” and “1” are used to indicate the idle and busy states, respectively [24]. Hidden Markov Model based spectrum prediction assume that the actual spectrum state is assumed to be hidden from secondary users [23, 24]. With $X = \{“0”, “1”\}$ denoting the actual/hidden state space, while $Y = \{“0”, “1”\}$ denoting the observation/sensing state space, Fig. 3. below shows the transition diagram for the Hidden Markov Model. In the diagram a_{ij} for $i, j \in X$ designate transition probabilities in the hidden state while b_{ij} for $i, j \in Y$ are the emission probabilities.

To improve prediction accuracy and considering availability of historic measurement data higher-order Markov model is proposed in [20]. The high-order Markov model provides better predication accuracy than first-order Markov model as it takes into account more historical information; however, increase of order increases the complexity exponentially grows, which leads to the increase of the prediction delay of the model [20].

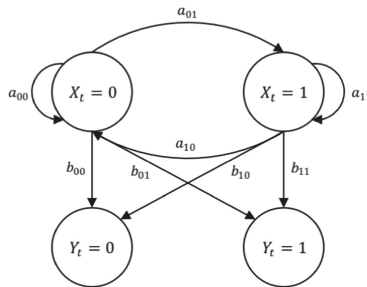


Fig. 3. Hidden Markov Model (HMM) [20].

5.4 Channel Modeling for Wireless Communication

Wireless channel is a channel whose gain is random and continuously varying in time. Widely used stochastic channels models include: Gaussian, Rician and Rayleigh [23]. In a typical wireless environment with no dominant line of sight propagation path between a transmitter and a receiver, Rayleigh fading is most applicable to capture the statistical variation of the channel. On the other hand, Rician model considers the presence of a dominant line-of-sight, in addition to reflection paths.

Though in reality the wireless channel gain continuously varies in time, what usually the case is that the gain remains constant or correlated for a certain time interval called the *coherent time* of the channel. Such a wireless channel is termed to be a *block fading* channel, and most researches that involve the wireless channel are based on this block fading assumption. In subsequent coherence time intervals, the channel gain either remains the same (i.e., the underlying factors that contribute for channel change are the same) or the channel gain changes to different levels. If we further assume that the channel gains are *quantized*, i.e., to have finite-values, and have finite discrete steps, then the Markov chain concept can be applied to model the channel with the following analogy [23–27]:

- Each level of the channel gain corresponds to states in Markov chain. Depending on the channel gain values, a two state channel model, where one classified as “good” and the other one classified as “bad” or a four state model with different types of statistical channels are investigated [23].
- Coherence time is related with the time with which the chain stays within a state.
- Changes in channel gains correspond to state transitions in Markov chain.
- The rate with which the channel changes states is a function of the underlying factors that change the channel and form the transition probabilities.
- Classical Markov chain or higher-order Markov chain models can be used to study the channel, depending on degree of correlation among successive states.

Knowing the possible channels gains that can be present in a transmission channel and applying Markov chains, it is possible to implement a more realistic channel [23]. Markov chain is also used in Switched Channel Model i.e., the channel switching between Rayleigh and Rice fading (the later with different K-factor), is investigated in [24–27]. The results indicate that the more state of the Markov chain, the better of the prediction of the channel performance.

5.5 Mobility Prediction in Cellular Networks

The mobility or motion pattern of a mobile user often exhibit some degree of regularity so that whereabouts of the user are predictable. Mobility prediction is widely used to assist handoff management, resource reservation and provide location-based services [12–16]. Markov chain-based mobility prediction approaches are widely proposed in the literature for their ability to yield better accuracy than most other predictors with lower complexity [12].

Handover in Cellular Networks: is a process of transferring users from a serving radio network (cell or basestation) to another radio network because of users' mobility. Seamless handover guarantees service continuity for mobile users, by reserving communication *resources* (e.g., bandwidth) on all the cells that the mobile users will probably visit during its session [17]. In order to achieve a seamless handover, the *handover latency* needs to be reduced, as resource allocation takes a lot of time.

By predicting the mobility pattern of users, in terms of future probably visited cells, the resource allocation can be performed prior to the actual handover, thus can reduce delays in resource allocation and also the handover latency [12–16].

Markov chain comes in handy for the users' location (at a cell level) prediction. In the Markov context:

- A user connected to a given cell over a time interval of interest would form the “state”;
- As the user moves from one cell to another cell, the user is declared to be moving and, hence, *state transition* occurs.
- The possible movement of the user within the coverage area would form the *transition matrix*. If we track and record the cells visited by an individual user over a certain period of time, the user's mobility history generates the Markov chain transition matrix.
- The initial distribution can be derived from the user' initial state.

The Markov chain approach can be used to predict the position (or mobility pattern) of the user over future time steps, which contains temporal information. Higher-order Markov model can also be used to improve mobility prediction accuracy [14]. One challenge to apply Markov chain based approach is availability of historical data as mobile operators may not be allowed by the customers to use their historical data due to privacy concerns. Moreover, extraction and processing complexity of historical records is another challenge for the implementation.

5.6 Markov Chain for Network Accessibility and Retainability Prediction

To deliver reliable services to customers, mobile network operators monitor performance of their radio access network and core network using different key performance indicators, such as accessibility, retainability, mobility, integrity and availability. Moreover, taking corrective actions based on monitored parameters is a reactive approach and contributes to network and service degradations until corrective actions are taken. With huge investment mobile network operators implement network management systems that monitor historic network performance data, and with of upgrade can analyze possible trends and patters about the network and services. So a systematic way of monitoring and predicting the network performance status is required to guaranty service quality.

The papers in [34, 35] use discrete time Markov chain model to predict accessibility performance on Long Term Evolution mobile system based on KPI parameters data gathered from LTE network for eNode B for a period of one month. Based on the KPI's value the network condition could be categorized into three states and after identifying the state sequence, a transition probability matrix is used to predict future accessibility state of LTE network.

5.7 Network Services and Maintenance Optimization

Telecommunication network equipment's and links fail intermittently for reasons related to rehabilitation of network infrastructure, physical damage of transmission cables, power problems etc. In addition, network congestion (as a result of growth in attractive telecom services and subscribers' growth), complexity in telecom equipment's and other factors increases stress to telecom infrastructure and to maintenance activity. These issues have to be solved timely in order to maintain the quality of service. Telecom operators, like any other industrial company, implement corrective maintenance, preventive maintenance and predictive maintenance strategies.

- Corrective maintenance is a reactive maintenance that, intervention to repair and restore normal operation, is done once failures occur. In the past few decades, telecom operators have used corrective maintenance based strategy. Though simpler to implement, this strategy cannot satisfy the need of customers as service outage and frequent failures of devices affect QoS.
- Preventive maintenance is a proactive approach carried out at predetermined intervals of time or according to prescribed criteria. It is intended to reduce probability of failure, degradation of the operation of an item and/or degradation of a service.
- Predictive maintenance is carried out by monitoring indicative or significant/critical parameters of equipments and services, and extrapolating/predicting based on historic data potential failures. The prediction is on the probability of the equipment/service will become faulty in the future and/or fault indication (symptom) of degradation conditions (events) of devices. Predicting the performance of the network of infrastructures is relevant to determining the optimal maintenance activity for a network of infrastructures.

Different types of Markov chain model can be used for root causes analysis of existing problems, predicting the performance of the network, optimizing operation and maintenance, and performance and reliability modeling [37]. Markov chain models have the ability to capture the time-dependence and uncertainty of the deterioration process.

6 Conclusion

A Markov Chain is one simpler but powerful tool that is used to model dynamical systems that changes their states or outputs over time. Classical Markov chain formulation assumes that the evolution of a state depends only on the current state, not on the past states. This is the Markov property. So far Markov chain is applied in wide range of disciplines, where the telecommunication engineering is one area. In order to improve the prediction accuracy and based on availability of sufficient data, higher-order Markov chain is used for some applications. Moreover, in cases where parameters of interest are not directly observable, Hidden Markov models are finding widespread applications. We hope that the review provides good insight on basic concepts and terminologies used in the use of Markov chain.

References

1. Tolver A.: An Introduction to Markov Chain, Lecture Notes for Stochastic Process. Department of Mathematical Sciences, University of Copenhagen (2016)
2. Seneta, E.: Markov and the creation of Markov Chains. School of Mathematics and Statistics, University of Sydney (2006)
3. Baum, L., Petrie, T.: Statistical inference for probabilistic functions of finite state Markov chains. *Ann. Math. Stat.* **37**(6), 1554–1563 (1966)
4. A brief Introduction to Markov Chains with Examples (2021). <https://www.edureka.co/blog/introduction-to-markov-chains/#What%20Is%20A%20Markov%20Chain>. Accessed 13 Mar 13
5. Markov Chain (2021). https://en.wikipedia.org/wiki/Markov_chain. Accessed 13 Mar 2021
6. Powell, V.: Markov Chains Explained Visually (2021). <https://setosa.io/ev/markov-chains/>. Accessed 9 May 2021
7. Privault, N.: Understanding Markov Chains. Springer, Singapore (2013). <https://doi.org/10.1007/978-981-13-0659-4>
8. Bickenbach, F., Bode, E.: Markov or Not Markov – This Should Be a Question. Kiel Working Paper No. 1086, December 2001. <https://www.files.ethz.ch/isn/124233/kap1086.pdf>. Accessed 26 Aug 2021
9. Stamp, M.: A Revealing Introduction to Hidden Markov Models. Department of Computer Science, San Jose State University (2021)
10. Hassan, R., Nath, B.: Stock market forecasting using hidden Markov model: a new approach (2005)
11. Mo, J.: Performance modeling of communication networks with Markov chains. *Synth. Lect. Commun. Netw.* **3**, 1–90 (2010)
12. Fazio, P., Marano, S: Mobility prediction and resource reservation in cellular networks with distributed Markov chains. In: 8th International Wireless Communications and Mobile Computing Conference (IWCMC), Limassol, Cyprus (2012)
13. Amirrudin, N., Ariffin, S., Malik, N., Ghazali, E.: Mobility prediction via Markov model in LTE Femtocell. *Int. J. Comput. Appl.* (0975–8887), **65**(18), (2013)
14. Zaidi, S., Manalastas, M., Farooq, H., Imran, A.: Mobility management in emerging ultra-dense cellular networks: a survey, outlook, and future research directions. *IEEE Access* **8**, 183505–183533 (2020)
15. Bellahsene, S., Kloul, L.: A new Markov-based mobility prediction algorithm for mobile networks. In: Aldini, A., Bernardo, M., Bononi, L., Cortellessa, V. (eds.) *Computer Performance Engineering*, vol. 6342, pp. 37–50. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15784-4_3
16. Hadachi, A., Batrashev, O., Lind, A., Singer, G., Vainikko, E.: Cell phone subscribers mobility prediction using enhanced Markov chain algorithm. In: *IEEE Intelligent Vehicles Symposium (IV)*, 8–11 June 2014, Dearborn, Michigan, USA (2014)
17. Lahby, M., Essouiri, A., Sekkaki, A.: A novel modeling approach for vertical handover based on dynamic k-partite graph in heterogeneous networks. *Digit. Commun. Netw.* **5**(4), 297–307 (2019)
18. Wu, J., Li, Y.: A survey of spectrum prediction methods in cognitive radio networks. In: *AIP Conference Proceedings*, vol. 1834, p. 020018 (2017)
19. Song, C., Chen, D., Zhang, Q.: Understand the predictability of wireless spectrum: a large-scale empirical study. In: 2010 IEEE International Conference on Communications (2010). <https://doi.org/10.1109/ICC.2010.5502054>
20. Oluwaranti, A., Okegbile, S.: Two state Markov chain based predictive model for cognitive radio spectrum availability: a conceptual approach. In: 2016 Future Technologies Conference (FTC), pp. 179–186 (2016)

21. Eltom, H., Kandeepan, S., Moran B., Evans, R.: Spectrum occupancy prediction using a hidden Markov model. In: 9th International Conference on Signal Processing and Communication Systems (ICSPCS) (2015)
22. Hong, Z., Zhao1, Y. Luo, Y., Wang G., Pu, L: Advanced high-order hidden bivariate Markov model based spectrum prediction. *EAI Endorsed Trans. Wirel. Spectr.* **3**(10) (2017)
23. Liu, X., Liu, C. Liu, W., Zeng, X.: Wireless channel modeling and performance analysis based on Markov chain. In: 29th Chinese Control and Decision Conference (CCDC), pp. 2256–2260 (2017)
24. Castanet, L.: Channel modelling based on N-state Markov chains for SatCom systems simulation. In: 12th International Conference on Antennas and Propagation (ICAP 2003) (2003). <https://doi.org/10.1049/cp:20030030>
25. Sánchez-Salas, D., Cuevas-Ruíz, J., González-Mendoza, M.: Wireless channel model with Markov Chains using MATLAB (2012). <https://doi.org/10.5772/46475>
26. Sanchez-Salas, D., Cuevas-Ruiz, J.: N-states channel model using Markov chains. In: Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007), pp. 342–347 (2007). <https://doi.org/10.1109/CERMA.2007.4367710>
27. Zhang, S., Wang, H., Zhang, X.: Estimation of Channel State Transition Probabilities Based on Markov Chains in Cognitive Radio
28. Terefe, T., Hailemariam, D.: Entropy estimation and entropy-based encoding of written Amharic language for efficient transmission in telecom networks. In: Proceeding of IEEE AFRICON, Cape Town, South Africa, pp. 238–244 (2017)
29. Jurafsky, D., Martin, J.: Hidden Markov Models. *Speech and Language Processing*, 30 December 2020
30. Rabiner, R. *Hidden Markov Models and Selected Applications in Speech Recognition* (1989)
31. How Google autocomplete works in Search. <https://www.blog.google/products/search/how-google-autocomplete-works-search/>. Accessed 29 Aug 2021
32. Text Generation With Markov Chains. <https://algotech.netlify.app/blog/text-generating-with-markov-chains/>. Accessed 29 Aug 2021
33. Hands-On Guide To Markov Chain For Text Generation. <https://analyticsindiamag.com/hands-on-guide-to-markov-chain-for-text-generation/>. Accessed 29 Aug 2021
34. Hendrawan: RRC success rate accessibility prediction on SAE/LTE network using Markov chain model. In: 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), pp. 1–5 (2017)
35. Hendrawan: Accessibility degradation prediction on LTE/SAE Network using discrete time Markov chain (DTMC) model. *ITB J.* **13**(1), 18 (2019)
36. Kim, J., Hwang, J. Kim, K.: High-performance internet traffic classification using a Markov Model and Kullback-Leibler divergence. *Mobile Inf. Syst.* **2016** (2016)
37. Frankel, E.: Testing for Markov properties. In: *Systems Reliability and Risk Analysis. Engineering Applications of Systems Reliability and Risk Analysis*, vol. 1. Springer, Dordrecht (1984). https://doi.org/10.1007/978-94-009-6920-9_8
38. Markov, A.A.: Extension of the law of large numbers to dependent quantities. *IzvestiiaFiz.-Matem. Obsch. Kazan Univ. (2nd Ser.)* **15**, 135–156 (1906). (in Russian)