11 Predicting future states with Markov analysis

This chapter covers

- States, state probabilities, and transition matrices
- Vector of state probabilities
- Equilibrium conditions
- Absorbing states
- Matrix operations in Markov chains

Markov analysis is a mathematical technique used to predict future states of a system—such as a market, a machine, or a population—based on its current state and the probabilities of transitioning from one state to another. It is applicable specifically to systems that exhibit the Markov property, meaning future states depend only on the present state and not on the sequence of past events. This memoryless characteristic makes Markov analysis particularly useful for modeling dynamic processes where past influences can be effectively summarized by the present state.

The technique originated from the work of Andrey Markov, a Russian mathematician, in the early 20th century and has since been widely applied in various fields, including market share predictions and delinquency forecasting.

In market share analysis, businesses use Markov techniques to predict how market shares for different competitors will evolve over time based on current market conditions and consumer behavior patterns. For example, a company can estimate the likelihood of a customer switching from its brand to a competitor's brand within a given time period. Similarly, in forecasting delinquencies, financial institutions use Markov analysis to predict the probability that a borrower will move from one delinquency bucket to another, such as from being current on payments to being 30 days late, and then to 60 days late, and so on.

11.1 Understanding the mechanics of Markov analysis

Markov analysis works by defining a finite number of possible states and calculating the probability of transitioning from one state to another. These probabilities are organized in a transition probability matrix, which serves as the foundation for predicting future states. The analysis assumes that the probability of changing states remains constant over time, meaning the transition probabilities do not vary. Additionally, it is assumed that the size and makeup of the system do not change over time.

The key assumption behind all Markov analyses is that the future state of a system depends only on its current state and not on the sequence of events that preceded it. This property, known as the *Markov property*, allows for the prediction of any future state using the current state and the transition probability matrix. This matrix, a square matrix in which each element represents the probability of moving from one state to another, is central to Markov analysis.

Equilibrium conditions in Markov analysis refer to a state where the probabilities of being in each state remain constant over time. This occurs when the system has reached a *steady state*, meaning the distribution of states does not change as further transitions occur. Understanding equilibrium conditions is crucial for long-term predictions and stability analysis.

Absorbing states are special types of states in a Markov process: once entered, the system remains in that state permanently. These states are essential for analyzing processes that have terminal conditions, such as loan defaults and customer churn.

Throughout this chapter, we will demonstrate how to perform mathematical operations with matrices to conduct Markov analysis. This includes constructing and manipulating the transition probability matrix, calculating state probabilities, and determining equilibrium conditions. We will also explore how to identify and analyze absorbing states, providing a comprehensive understanding of how Markov analysis can be used to predict future states and inform decision-making in various practical applications. By the end of this chapter, you will have a solid understanding of the principles and techniques of Markov analysis, and you will be equipped with the skills to apply this powerful tool to a range of predictive modeling scenarios.

11.2 States and state probabilities

A *state* in Markov analysis represents a distinct condition or status that a process or system can be in at any given point in time. Identifying all possible states is the first step for understanding and predicting the behavior of the system. Each state provides a snapshot of the system's current condition, enabling analysts to model transitions and future states.

States are used to identify all possible conditions of a process or system. These conditions can be binary, representing two possible states. For instance, consider the engine of a classic automobile, which can either be running or not running. These two states encompass all possible conditions of the engine at any given moment.

States can also be more complex and involve multiple categories. For example, in the context of market share among four cellular carriers, the states could represent the percentage of the market held by each carrier. Here, the system's conditions are represented by the various market shares of the four carriers, providing a more detailed understanding of the market dynamics.

In Markov analysis, it is essential that the set of states be collectively exhaustive and mutually exclusive. *Collectively exhaustive* means all possible conditions of the system are represented within the set of states. No condition is left out. For instance, in the classic automobile example, the states "running" and "not running" cover all possible conditions of the engine. In the market share example, the states representing the market shares of all four cellular carriers must sum up to 100%, ensuring that all possible market conditions are included.

Mutually exclusive means no two states can occur simultaneously. Each state is distinct and non-overlapping. An automobile engine cannot be both running and not running at the same time. Similarly, in the market share example, a specific percentage of the market cannot be simultaneously attributed to two different carriers. Each percentage point of market share belongs to one and only one carrier.

By defining states that are both collectively exhaustive and mutually exclusive, analysts can accurately model the system and predict its future behavior. This clear delineation of states ensures a comprehensive and precise analysis, which is fundamental to the effectiveness of Markov analysis in predicting future states.

The second step is to determine the probabilities across states. These probabilities are stored in what is called a *vector of state probabilities*:

$$\begin{aligned} \pi\,(i) &= \text{vector of state probabilities for period } i \\ &= (\pi_1, \pi_2, \pi_3, \ldots, \pi_n) \end{aligned}$$

where

- n equals the number of states.
- $\pi_1, \pi_2, \pi_3, \ldots, \pi_n$ equals the probability of being in state 1, state 2,…, state n.

With respect to the classic automobile engine, which can be in only one of two states for any given period—running or not running—it's relatively straightforward to mathematically represent the vector of state probabilities:

$$\pi(0) = (1, 0)$$

where

- $\pi(0)$ equals the vector of states for the automobile engine in period 0. A period can represent any unit of time, such as a day, a week, a quarter, or a year. However, it is essential that historical and future periods are measured consistently. Periods typically start at 0, so that period 1 represents the first period following the initial transition.
- π_1 equal to 1 represents the probability of being in the first state, which, in this case, means the probability of the automobile engine being in running condition.
- π_2 equal to 0 represents the probability of being in the second state, which means the probability of the automobile engine not running.

This indicates that the probability of the automobile engine being in running condition (state 1) is 1, whereas the probability of it not being in running condition (state 2) is 0 for the first period. However, in most cases, we need to consider systems with multiple items or components.

11.2.1 Understanding the vector of state probabilities for multistate systems

Consider the market share distribution among four (fictional) US cellular carriers: Horizon, TeleConnect, AmeriCon, and USAMobile. Collectively, these carriers own 100% of the US market, and their market shares can be represented as state probabilities:

- Horizon has a 40% share of the US market. In market share analysis, the share of the market is equivalent to probability. So, state 1 is assigned to Horizon with a probability of 0.40.
- TeleConnect owns 35% of the market, which is assigned as state 2 with a probability of 0.35.
- AmeriCon owns 20% of the market, making it state 3 with a probability of 0.20.
- USAMobile owns 5% of the market, designated as state 4 with a probability of 0.05.

These market shares represent the probabilities of consumers having wireless subscriptions to one of these carriers in period 0 (before any transitions have occurred). Each state probability signifies the likelihood of a randomly selected consumer being a customer of a particular carrier at this point in time.

Understanding the vector of state probabilities in a multistate system is crucial for predicting how these probabilities might change over time. By analyzing the transition probabilities between these states, we can forecast shifts in market share and make informed decisions based on anticipated trends. This method is applicable not only to market share analysis but also to various other fields where the system can be modeled as transitioning between different states.

These probabilities can therefore be stored in a vector of state probabilities, such as

$$\pi(i) = (0.40, 0.35, 0.20, 0.05)$$

where

- $\pi(i)$ equals a vector of state probabilities for the four cellular carriers for period 0.
- π_1 is the probability that a consumer has a subscription to Horizon (state 1), which is equal to 40%.
- π_2 is the probability that a consumer has a subscription to TeleConnect (state 2), which is equal to 35%.
- π_3 is the probability that a consumer has a subscription to AmeriCon (state 3), which is equal to 20%.
- π_4 is the probability that a consumer has a subscription to USAMobile (state 4), which is equal to 5%.

The probabilities sum to 1, and the respective market shares add to 100%.

Churn is a significant factor in the wireless industry, where subscriptions are typically on a month-to-month basis. This flexibility allows consumers to switch carriers frequently, especially as carriers continually introduce new and attractive contract offers. Consumers have the option to terminate their subscription with one carrier and take their existing phone number to a different carrier, making the market highly dynamic and competitive.

Based on a study of recent historical figures, we observe the following transition probabilities for the four US cellular carriers (Horizon, TeleConnect, AmeriCon, and USAMobile). These figures illustrate how likely customers are to stay with their current carrier or switch to a competitor in the next period, which is the following quarter:

- Horizon
  - 85% of customers will stay with Horizon.
  - 6% will switch to TeleConnect.
  - 7% will switch to AmeriCon.
  - 2% will switch to USAMobile.
- TeleConnect
  - 80% of customers will stay with TeleConnect.
  - 8% will switch to Horizon.
  - 10% will switch to AmeriCon.
  - 2% will switch to USAMobile.

- AmeriCon
  - 75% of customers will stay with AmeriCon.
  - 10% will switch to Horizon.
  - 10% will switch to TeleConnect.
  - 5% will switch to USAMobile.
- USAMobile
  - 70% of customers will stay with USAMobile.
  - 12% will switch to Horizon.
  - 10% will switch to TeleConnect.
  - 8% will switch to AmeriCon.

These transition probabilities reflect the competitive nature of the wireless market, where each carrier must continuously strive to retain its customers while attracting those from its competitors. Understanding these probabilities is crucial for predicting market dynamics and making strategic decisions in the wireless industry.

It is important to note that the transition probabilities for each carrier from period 0 to period 1 sum to 100%. This means every customer is accounted for, ensuring that the model reflects the entire customer base without any loss or gain in total numbers. This comprehensive accounting allows for accurate predictions and analyses of market share dynamics.

For example, 85% of Horizon's customers are expected to remain with Horizon, whereas the remaining 15% will switch to other carriers (6% to TeleConnect, 7% to AmeriCon, and 2% to USAMobile). Similarly, the transition probabilities for TeleConnect, AmeriCon, and USAMobile customers also sum to 100%, fully accounting for all possible movements between carriers.

This complete accounting is essential for understanding and predicting market behaviors, as it ensures that all potential customer movements are included in the analysis. It provides a clear and accurate picture of how market shares are likely to evolve over time, based on current trends and consumer behaviors.

The churn dynamics result in quarter-over-quarter market share gains for AmeriCon and USAMobile at the expense of Horizon and TeleConnect:

- Horizon's share of the US market drops from 40% to 39.4%.
- TeleConnect's market share drops by more than two percentage points, from 35% to 32.9%.
- AmeriCon now owns 21.7% of the US market, versus just 20% from the previous period.
- USAMobile's share of the market increases from 5% to 6%.

Let's go step by step to demonstrate how the market share for one of these carriers, Horizon, changed between periods 0 and 1:

1. Horizon, which initially held 40% of the US cellular market, retains 85% of its customers. Consequently, Horizon's share of the market decreases to 34%,

which is the product of 0.40×0.85 when the decimal result is converted to a percentage.

2. Horizon acquired a small percentage of customers from each of the other carriers, including 8% from TeleConnect. Given that TeleConnect held 35% of the US market, this translates to $0.35 \times 0.08 = 0.028$, or 2.8% of additional market share.
3. Horizon also attracted 10% of AmeriCon's 20% market share, which translates to $0.20 \times 0.10 = 0.02$, or an additional 2% market share.
4. Horizon acquired 12% of USAMobile's 5% share of the market, which translates to $0.05 \times 0.12 = .06$, or 0.6% of new market share.

Horizon's new share of the US cellular market is derived by adding these four products together: 34% + 2.8% + 2% + 0.6% = 39.4%.

We could repeat these steps for the remaining three carriers, and as long as the system remains intact—without mergers, acquisitions, new entrants into the marketplace, and with constant probabilities for changing states—we could predict the market share distribution two quarters or even two years into the future. However, there are more efficient ways of applying Markov analysis to make these predictions.

11.2.2 Matrix of transition probabilities

The *matrix of transition probabilities* is a square matrix used in Markov analysis to describe the probabilities of transitioning from one state to another within a system over a specified period. Each element in the matrix represents the probability of moving from a specific initial state to a specific subsequent state. More precisely, it is a matrix of *conditional* probabilities, as the future state depends on the current state, so that

$$P_{ij} = \text{conditional probability of being in future state } j \text{ given current state } i$$

The matrix of transition probabilities is a fundamental component in Markov analysis, representing the likelihood of transitioning between states in a given system over a specified period. The matrix is denoted as P and is structured as a square matrix, where each element P_{ij} indicates the probability of moving from state i to state j. For illustrative purposes, consider the following matrix of transition probabilities:

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{1n} \\ P_{21} & P_{22} & P_{23} & P_{2n} \\ P_{31} & P_{32} & P_{33} & P_{3n} \\ P_{41} & P_{42} & P_{43} & P_{4n} \end{bmatrix}$$

For example, P_{13} represents the probability of transitioning from a current state of 1 to a future state of 3; or take P_{42}, which represents the probability of transitioning from state 4 to state 2.

Returning to the four US cellular providers, we established period-to-period probabilities for either staying with the current carrier or switching to a different carrier.

These probabilities are then incorporated into the following matrix of transition probabilities:

$$P = \begin{bmatrix} 0.85 & 0.06 & 0.07 & 0.02 \\ 0.08 & 0.80 & 0.10 & 0.02 \\ 0.10 & 0.10 & 0.75 & 0.05 \\ 0.12 & 0.10 & 0.08 & 0.70 \end{bmatrix}$$

The system under analysis is the US wireless market, where the states are defined as the four carriers that collectively hold 100% of the market share. In this system, Horizon represents state 1, TeleConnect is state 2, AmeriCon is state 3, and USAMobile is state 4. Thus, P_{13} represents the probability of transitioning from Horizon to AmeriCon, and P_{42} represents the probability of transitioning from USAMobile to TeleConnect. Overall, the entire matrix can be interpreted as follows:

- Row 1:
  - $P_{11} = 0.85$ = probability of being in state 1 given state 1 from the prior period
  - $P_{12} = 0.06$ = probability of being in state 2 given state 1 from the prior period
  - $P_{13} = 0.07$ = probability of being in state 3 given state 1 from the prior period
  - $P_{14} = 0.02$ = probability of being in state 4 given state 1 from the prior period
- Row 2:
  - $P_{21} = 0.08$ = probability of being in state 1 given state 2 from the prior period
  - $P_{22} = 0.80$ = probability of being in state 2 given state 2 from the prior period
  - $P_{23} = 0.10$ = probability of being in state 3 given state 2 from the prior period
  - $P_{24} = 0.02$ = probability of being in state 4 given state 2 from the prior period
- Row 3:
  - $P_{31} = 0.10$ = probability of being in state 1 given state 3 from the prior period
  - $P_{32} = 0.10$ = probability of being in state 2 given state 3 from the prior period
  - $P_{33} = 0.75$ = probability of being in state 3 given state 3 from the prior period
  - $P_{34} = 0.05$ = probability of being in state 4 given state 3 from the prior period
- Row 4:
  - $P_{41} = 0.12$ = probability of being in state 1 given state 4 from the prior period
  - $P_{42} = 0.10$ = probability of being in state 2 given state 4 from the prior period
  - $P_{43} = 0.08$ = probability of being in state 3 given state 4 from the prior period
  - $P_{44} = 0.70$ = probability of being in state 4 given state 4 from the prior period

The vector of state probabilities—that is, the respective market share for each carrier at period 0—combined with the matrix of transition probabilities allows us to predict future state probabilities. This allows us to model and forecast how market shares will evolve over subsequent periods based on the established probabilities of customers either staying with their current carrier or switching to a different one.

PREDICTING PERIOD 1 GIVEN PERIOD 0

The purpose of Markov analysis is to predict the future states of a system based on its current state and the probabilities of transitioning between states. In the context of the US cellular market, we can predict the distribution of market shares for the next period, such as period 1 or the start of the next quarter, given the market share at period 0. This prediction is achieved by multiplying the vector of state probabilities by the matrix of transition probabilities, which is mathematically expressed as follows:

$$\pi(1) = \pi(0)P$$

Thus, we can predict the respective market shares for Horizon, TeleConnect, AmeriCon, and USAMobile with the following matrix multiplication calculations:

$$\begin{aligned}
&= (0.40, 0.35, 0.20, 0.05)\begin{bmatrix} 0.85 & 0.06 & 0.07 & 0.02 \\ 0.08 & 0.80 & 0.10 & 0.02 \\ 0.10 & 0.10 & 0.75 & 0.05 \\ 0.12 & 0.10 & 0.08 & 0.70 \end{bmatrix} \\
&= [(0.40)(0.85) + (0.35)(0.08) + (0.20)(0.10) + (0.05)(0.12), \\
&\quad (0.40)(0.06) + (0.35)(0.80) + (0.20)(0.10) + (0.05)(0.10), \\
&\quad (0.40)(0.07) + (0.35)(0.10) + (0.20)(0.75) + (0.05)(0.08), \\
&\quad (0.40)(0.02) + (0.35)(0.02) + (0.20)(0.05) + (0.05)(0.70)] \\
&= (0.394, 0.329, 0.217, 0.06)
\end{aligned}$$

Matrix multiplication involves taking the dot product of each element in the initial vector with the corresponding row in the transition matrix. A *dot product* is an algebraic operation that takes two equal-length sequences of numbers and returns a single number, calculated as the sum of the products of corresponding elements. For example, to calculate the probability of being in the first state (Horizon) in the next period, we multiply each element in the initial vector by the corresponding element in the first column of the matrix and sum the results. The same process is repeated for the other states. This operation gives us a new vector that represents the predicted market shares for Horizon, TeleConnect, AmeriCon, and USAMobile in the next period.

PREDICTING PERIOD 2 GIVEN PERIOD 1

We have a new vector of state probabilities, but the matrix of transition probabilities remains the same. In fact, this matrix must remain constant for us to predict future states using Markov analysis. Given the new market shares, we can predict what the market share distribution will be in the next period by again multiplying the (new) vector of state probabilities by the matrix of transition probabilities. This iterative process allows us to forecast market dynamics over multiple periods accurately.

To predict period 2 from period 1, the following equation is used:

$$\pi(2) = \pi(1)P$$

This equation shows how we use the state probabilities from period 1 to project the market shares for period 2:

$$
\begin{aligned}
&= (0.394, 0.329, 0.217, 0.06)\begin{bmatrix} 0.85 & 0.06 & 0.07 & 0.02 \\ 0.08 & 0.80 & 0.10 & 0.02 \\ 0.10 & 0.10 & 0.75 & 0.05 \\ 0.12 & 0.10 & 0.08 & 0.70 \end{bmatrix} \\
&= [(0.394)(0.85) + (0.329)(0.08) + (0.217)(0.10) + (0.06)(0.12), \\
&\quad (0.394)(0.06) + (0.329)(0.80) + (0.217)(0.10) + (0.06)(0.10), \\
&\quad (0.394)(0.07) + (0.329)(0.10) + (0.217)(0.75) + (0.06)(0.08) \\
&\quad (0.394)(0.02) + (0.329)(0.02) + (0.217)(0.05) + (0.06)(0.70)] \\
&= (0.39012, 0.31454, 0.22803, 0.06731)
\end{aligned}
$$

We again observe a decline in market share for Horizon and TeleConnect, whereas AmeriCon and USAMobile experience increases. The magnitude of these losses and gains, however, is not as pronounced between periods 1 and 2 as it was between periods 0 and 1. This suggests that the market dynamics are stabilizing, with the initial significant shifts in market share giving way to more moderate changes in subsequent periods.

Predicting period 2 given period 0

Bypassing the need to predict period 1, it is possible to directly predict period 2 from period 0 using this equation:

$$\pi(2) = \pi(0)P^2$$

This method involves multiplying the original vector of state probabilities by the square of the transition probability matrix. The mathematical approach starts with squaring the transition probability matrix. This squared matrix encapsulates the probabilities of transitioning between states over two periods. Next, the initial vector of state probabilities is multiplied by P^2 to derive the state probabilities for period 2.

Multiplying a 4 × 4 matrix by itself is straightforward but messy and prone to errors when performed manually. To find each element in the resulting matrix, we take the dot product of the ith row of the first matrix with the jth column of the second matrix, like so:

- First row, first column:
  - $= (0.85)(0.85) + (0.06)(0.08) + (0.07)(0.10) + (0.02)(0.12)$
  - $= 0.7225 + 0.0048 + 0.0070 + 0.0024$
  - $= 0.7367$
- First row, second column:
  - $= (0.85)(0.06) + (0.06)(0.80) + (0.07)(0.10) + (0.02)(0.10)$
  - $= 0.0510 + 0.0480 + 0.0070 + 0.0020$
  - $= 0.1080$

- First row, third column:
  - = (0.85) (0.07) + (0.06) (0.10) + (0.07) (0.75) + (0.02) (0.08)
  - = 0.0595 + 0.0060 + 0.0525 + 0.0016
  - = 0.1196
- First row, fourth column:
  - = (0.12) (0.02) + (0.10) (0.02) + (0.08) (0.05) + (0.70) (0.70)
  - = 0.0024 + 0.0020 + 0.0040 + 0.4900
  - = 0.4984

Each of these 16 derived elements represents the probability of transitioning from one state to another over two periods and is placed in the corresponding position in the new matrix. Specifically, each element P_{ij}^2 is calculated as the sum of the products of the transition probabilities from state i to all intermediary states and from those intermediary states to state j. The placement of these elements follows a structured approach, moving from left to right within each row and from top to bottom through the rows. This ensures that the calculated probabilities are correctly aligned with their respective state transitions. The resulting matrix of transition probabilities at period 2, denoted as P^2, effectively captures the compounded effect of two transitions:

$$P = \begin{bmatrix} 0.7367 & 0.1080 & 0.1196 & 0.0357 \\ 0.1444 & 0.6568 & 0.1622 & 0.0366 \\ 0.1740 & 0.1660 & 0.5835 & 0.0765 \\ 0.2020 & 0.1652 & 0.1344 & 0.4984 \end{bmatrix}$$

We get the period 2 market share distribution by then multiplying P^2 by the original vector of state probabilities:

$$\begin{aligned}
&= (0.40, 0.35, 0.20, 0.05) \begin{bmatrix} 0.7367 & 0.1080 & 0.1196 & 0.0357 \\ 0.1444 & 0.6568 & 0.1622 & 0.0366 \\ 0.1740 & 0.1660 & 0.5835 & 0.0765 \\ 0.2020 & 0.1652 & 0.1344 & 0.4984 \end{bmatrix} \\
&= [(0.40)(0.7367) + (0.35)(0.1444) + (0.20)(0.1740) + (0.05)(0.2020), \\
&\quad (0.40)(0.1080) + (0.35)(0.6568) + (0.20)(0.1660) + (0.05)(0.1652), \\
&\quad 0.40)(0.1196) + (0.35)(0.1622) + (0.20)(0.5835) + (0.05)(0.1344), \\
&\quad (0.40)(0.0357) + (0.35)(0.0366) + (0.20)(0.0765) + (0.05)(0.4984) \\
&= \ (0.39012,\ 0.31454,\ 0.22803,\ 0.06731)]
\end{aligned}$$

We have now demonstrated how to predict period 2 directly from period 0 using Markov analysis. By squaring the matrix of transition probabilities and then multiplying this new matrix by the original vector of state probabilities, we can effectively forecast future state distributions without needing to calculate intermediate periods. This method highlights the power and efficiency of Markov analysis in modeling complex systems and predicting their behavior over time. By using these techniques, we gain valuable insights into the dynamics of the system and can make more informed decisions based on these projections.

Having illustrated the manual process of predicting future states using Markov analysis, we next demonstrate how to make similar predictions programmatically. By utilizing Python and specifically the NumPy library to perform matrix mathematical operations automatically, we can efficiently apply Markov analysis to model and forecast state probabilities, further enhancing our ability to analyze complex systems with greater precision and speed.

Making future state predictions programmatically

We now transition from the manual calculations of Markov analysis to a programmatic approach using Python. By utilizing the power of Python and the NumPy library, we can automate matrix mathematical operations to efficiently predict future state probabilities. This approach not only saves time but also ensures accuracy when dealing with complex systems and large data sets.

To illustrate this process, we will use the same US cellular market share data we worked with manually. By using Python and NumPy, we can seamlessly perform Markov analysis to predict changes in market share distribution over multiple periods.

We begin by defining the initial market shares for period 0 and the transition probability matrix. We then use the NumPy `dot()` method to calculate the market shares for period 1 by multiplying the initial state probabilities by the transition matrix. Next, we compute the market shares for period 2 in two ways: first by using the market shares from period 1, and second by directly multiplying the initial state probabilities by the squared transition matrix. This automated approach demonstrates the efficiency and precision of using Python for Markov analysis, allowing us to quickly and accurately predict future state probabilities.

First, we import the `numpy` library. Then we create a pair of `numpy` arrays: the vector of state probabilities representing the initial market shares for period 0, and the matrix of transition probabilities, which details the probabilities of transitioning from one state to another:

```
>>> import numpy as np

>>> pi_0 = np.array([0.40, 0.35, 0.20, 0.05])

>>> P = np.array([
>>>     [0.85, 0.06, 0.07, 0.02],
>>>     [0.08, 0.80, 0.10, 0.02],
>>>     [0.10, 0.10, 0.75, 0.05],
>>>     [0.12, 0.10, 0.08, 0.70]
>>> ])
```

The array assigned to pi_0 is a list of initial market share values for the four states: Horizon, TeleConnect, AmeriCon, and USAMobile, respectively. The market shares are equivalent to probabilities, and the values in the array must always sum to exactly 1.

The array assigned to P is the transition probability matrix, which represents the probabilities of transitioning from one state to another. Each element P_{ij} indicates the probability of moving from state i to state j.

Our next snippet of code calculates the market shares for period 1 by performing a dot product between the vector of state probabilities (pi_0) and the transition probability matrix (P) and then prints the results:

```
>>> pi_1 = np.dot(pi_0, P)
>>> print('Market shares for period 1:')
>>> print(pi_1)
Market shares for period 1:
[0.394 0.329 0.217 0.06 ]
```

The np.dot() method takes the vector of state probabilities and the matrix of transition probabilities and computes the dot product between them. The result is a new vector, pi_1, which represents the market shares for period 1.

Now that we have calculated the market share distribution for period 1, we can proceed to predict the distribution for period 2. By making another call to the np.dot() method, we pass the newly obtained state probabilities from period 1 (pi_1) along with the transition probability matrix (P) to forecast the market shares for the next period:

```
>>> pi_2_from_1 = np.dot(pi_1, P)
>>> print('Market shares for period 2 given period 1:')
>>> print(pi_2_from_1)
Market shares for period 2 given period 1:
[0.39012 0.31454 0.22803 0.06731]
```

Finally, we predict the market shares for period 2 directly from period 0. This is done by first squaring the matrix of transition probabilities and then using the np.dot() method to multiply the squared matrix with the original vector of state probabilities. This approach allows us to bypass the intermediate step of calculating period 1 market shares:

```
>>> P2 = np.dot(P, P)
>>> print(P2)
[[0.7367 0.108  0.1196 0.0357]
 [0.1444 0.6568 0.1622 0.0366]
 [0.174  0.166  0.5835 0.0765]
 [0.202  0.1652 0.1344 0.4984]]

>>> pi_2_direct = np.dot(pi_0, P2)
>>> print('Market shares for period 2 given period 0:')
>>> print(pi_2_direct)
Market shares for period 2 given period 0:
[0.39012 0.31454 0.22803 0.06731]
```

The `np.dot()` method performs a dot product between two parameters. To obtain the square of the transition probability matrix `P` (resulting in `P2`), we pass `P` twice to `np.dot()`. Then we pass `P2` and the original vector of state probabilities (`pi_0`) to `np.dot()` to calculate the market shares for period 2 directly from period 0, thereby bypassing period 1. Clearly, applying Markov analysis programmatically to predict future states is significantly easier and more efficient than manually performing matrix mathematical operations.

Regardless of the method used, we observe that the rate of change in market share distribution diminished between periods 1 and 2 compared to the change from period 0 to period 1. This suggests that instead of USAMobile or AmeriCon becoming the market leader in the future, a state of equilibrium is eventually established, where subsequent changes in future states are so small as to be immaterial. Next, we will explore this phenomenon further.

11.3 Equilibrium conditions

Based on recent historical data, a classic automobile engine has a 90% probability of being in running condition during the current week if it was in the same condition the previous week, which means there is a 10% probability of the engine not running in the current week if it had been running the week prior. Conversely, if the engine was not running the previous week, there is a 20% probability that it will self-correct during the current week and therefore an 80% probability it will remain in that state for another week. Let state 1 represent the engine running and state 2 represent the engine not running. The probabilities of these two states must sum to exactly 1, as they are mutually exclusive and collectively exhaustive. These probabilities reflect the engine's behavior over time and are essential for constructing an accurate matrix of transition probabilities:

$$P = \begin{bmatrix} 0.90 & 0.10 \\ 0.20 & 0.80 \end{bmatrix}$$

where

- Row 1:
  - $P_{11} = 0.90$ = probability of being in state 1 given state 1 from the prior period
  - $P_{12} = 0.10$ = probability of being in state 2 given state 1 from the prior period
- Row 2:
  - $P_{21} = 0.20$ = probability of being in state 1 given state 2 from the prior period
  - $P_{22} = 0.80$ = probability of being in state 2 given state 2 from the prior period

To mathematically derive the probability of the engine being in state 1 over the next week, we multiply the vector of state probabilities by the matrix of transition probabilities:

$$
\begin{aligned}
\pi(1) &= \pi(0)P \\
&= (1, 0)\begin{bmatrix} 0.90 & 0.10 \\ 0.20 & 0.80 \end{bmatrix} \\
&= [(1)(0.90) + (0)(0.20), (1)(0.10) + (0)(0.80)] \\
&= (0.90, 0.10)
\end{aligned}
$$

To predict states 1 and 2 for the week following, we multiply the new vector of state probabilities by the matrix of transition probabilities, which must remain constant:

$$
\begin{aligned}
\pi(2) &= \pi(1)P \\
&= (0.90, 0.10)\begin{bmatrix} 0.90 & 0.10 \\ 0.20 & 0.80 \end{bmatrix} \\
&= [(0.90)(0.90) + (0.10)(0.20), (0.90)(0.10) + (0.10)(0.80)] \\
&= (0.83, 0.17)
\end{aligned}
$$

This means in two weeks' time (or two periods from now) there is an 83% probability of the engine running and a 17% probability of it not running. This does not necessarily mean that in six weeks or so the engine will be permanently out of commission. Systems typically stabilize and eventually settle into a steady state of mutually exclusive and collectively exhaustive probabilities.

This steady-state condition, also known as the *equilibrium distribution*, is a fundamental concept in Markov chains. A *Markov chain* is a stochastic process in which the probability of transitioning to a future state depends solely on the present state, not on the sequence of past states. In the case of the automobile engine, as the transition process repeats over time, the probabilities of being in each state will converge to a long-run equilibrium. This equilibrium, or steady-state distribution, represents the expected proportion of time the system will spend in each state over an extended period, providing valuable insights into the long-term behavior of the system.

11.3.1 Predicting equilibrium conditions programmatically

Mathematically, equilibrium conditions, sometimes referred to as *steady-state probabilities*, are achieved when the vector of state probabilities remains constant over time. This occurs because the transition probability matrix is already fixed, ensuring that the probabilities of transitioning from one state to another do not change. In a discrete-time Markov chain, this steady-state distribution describes the long-term proportion of time the system will spend in each state, regardless of its initial condition.

Using Python, we can iteratively compute these equilibrium conditions by repeatedly applying the transition probability matrix to the initial vector of state probabilities over multiple periods until the state probabilities stabilize. Unlike methods that rely on random sampling to approximate distributions, this approach directly computes the steady-state probabilities using matrix operations. This allows us to efficiently analyze long-term system behavior and determine the steady-state distribution in a structured, deterministic manner.

The following snippet of Python code predicts the state probabilities over the next 20 periods, starting with an initial state vector and a transition probability matrix. This process involves repeatedly applying the transition matrix to the state vector and storing the results, ultimately allowing us to observe how the probabilities converge to their equilibrium values:

```
>>> pi_0 = np.array([1, 0])

>>> P = np.array([
>>>     [0.90, 0.10],
>>>     [0.20, 0.80]
>>> ])

>>> n_periods = 20

>>> state_probabilities = [pi_0]

>>> for _ in range(n_periods):
>>>     pi_next = np.dot(state_probabilities[-1], P)
>>>     state_probabilities.append(pi_next)

>>> for i, pi in enumerate(state_probabilities):
>>>     print(f'Period {i} = {pi}')
Period 0 = [1 0]
Period 1 = [0.9 0.1]
Period 2 = [0.83 0.17]
Period 3 = [0.781 0.219]
Period 4 = [0.7467 0.2533]
Period 5 = [0.72269 0.27731]
Period 6 = [0.705883 0.294117]
Period 7 = [0.6941181 0.3058819]
Period 8 = [0.68588267 0.31411733]
Period 9 = [0.68011787 0.31988213]
Period 10 = [0.67608251 0.32391749]
Period 11 = [0.67325776 0.32674224]
Period 12 = [0.67128043 0.32871957]
Period 13 = [0.6698963 0.3301037]
Period 14 = [0.66892741 0.33107259]
Period 15 = [0.66824919 0.33175081]
Period 16 = [0.66777443 0.33222557]
Period 17 = [0.6674421 0.3325579]
Period 18 = [0.66720947 0.33279053]
Period 19 = [0.66704663 0.33295337]
Period 20 = [0.66693264 0.33306736]
```

Creates an array that represents the original vector of state probabilities

Creates an array that represents the matrix of transition probabilities

Sets the number of periods to predict

Initializes a list with the vector of state probabilities and stores it for each period

Creates a loop over the specified number of periods, calculates a dot product between the most recent state probabilities and the transition matrix, and then appends the results to the state_probabilities list

Iterates over the same list and prints the state probabilities for each period

The results demonstrate how the probabilities of the engine being in a running condition (state 1) and not running (state 2) evolve over 20 periods. Starting with the engine fully in a running condition at period 0, the probabilities gradually shift until equilibrium conditions are achieved around period 10. At this point, the engine consistently has about a two-thirds probability of being in a running condition, or approximately 66.7%. This steady state continues with minimal fluctuations beyond period 10, indicating that the system has stabilized. This example highlights the effectiveness of Markov analysis in predicting long-term behavior and achieving equilibrium conditions within a system.

Perhaps even more fascinating is the observation that small adjustments in the matrix of transition probabilities can lead to significant changes in the steady-state probabilities. This sensitivity underscores the importance of accurately determining the transition probabilities, as even minor variations can dramatically alter the long-term behavior of the system.

To demonstrate this phenomenon, let's replace our original matrix of transition probabilities with the following matrix:

$$P = \begin{bmatrix} 0.80 & 0.20 \\ 0.10 & 0.90 \end{bmatrix}$$

This new matrix of transition probabilities shows a slight decrease in the probability of remaining in the same state and therefore an equal increase in the probability of transitioning to the other state compared to the original matrix.

We then run the same snippet of code, but with the new matrix of transition probabilities in lieu of the original matrix. Here are the results:

```
Period 0 = [1 0]
Period 1 = [0.8 0.2]
Period 2 = [0.66 0.34]
Period 3 = [0.562 0.438]
Period 4 = [0.4934 0.5066]
Period 5 = [0.44538 0.55462]
Period 6 = [0.411766 0.588234]
Period 7 = [0.3882362 0.6117638]
Period 8 = [0.37176534 0.62823466]
Period 9 = [0.36023574 0.63976426]
Period 10 = [0.35216502 0.64783498]
Period 11 = [0.34651551 0.65348449]
Period 12 = [0.34256086 0.65743914]
Period 13 = [0.3397926 0.6602074]
Period 14 = [0.33785482 0.66214518]
Period 15 = [0.33649837 0.66350163]
Period 16 = [0.33554886 0.66445114]
Period 17 = [0.3348842 0.6651158]
Period 18 = [0.33441894 0.66558106]
Period 19 = [0.33409326 0.66590674]
Period 20 = [0.33386528 0.66613472]
```

Comparing these results to the original results, we clearly see that small changes in the transition probabilities can lead to significant shifts in the steady-state probabilities.

With the new transition matrix, the steady-state probabilities drastically change, resulting in the engine having a higher likelihood of not running (approximately 67%) rather than running (approximately 33%) in the long term. This demonstrates the sensitivity of Markov analysis to the transition probabilities and highlights the importance of accurately estimating these probabilities to effectively predict the system's behavior. These observations on the sensitivity of transition probabilities naturally lead us to the concept of absorbing states and how their presence can further affect our analysis and predictions.

11.4 Absorbing states

In our previous analysis, we assumed that it is possible to transition from any state to any other state. However, this is not always the case. In some systems, certain states may act as end states: once such a state is entered, the system cannot transition to any other state. These are *absorbing states,* defined as states that, once reached, cannot be left. In other words, the probability of transitioning from an absorbing state to any other state is zero, and the probability of remaining in the absorbing state is one. Understanding absorbing states is crucial as they fundamentally alter the dynamics of a system and the way we predict future states.

Harmony Finance is a fictional bank that specializes in providing revolving credit to qualified customers. It issues credit cards to customers who meet specific eligibility criteria, allowing them to borrow money up to a set limit and repay it over time with interest. Most of Harmony Finance's customers manage their credit responsibly and pay off their balances in full every month. However, some customers become delinquent, meaning they miss their payment deadlines. Among these delinquent customers, a portion eventually have their accounts charged off, indicating that the bank considers the debt unlikely to be collected. Thus, an account can be (and must be) in only one of the following states in any given month:

- *State 1* (π_1)—The account is current, meaning there is no outstanding balance.
- *State 2* (π_2)—The account is charged off.
- *State 3* (π_3)—The account is in what's called early-stage delinquency, where two or three payments are due: the current payment plus one or two delinquent payments.
- *State 4* (π_4)—The account is in late-stage delinquency, where four to seven payments are due: the current payment plus three or more delinquent payments.

The first two states are absorbing states because it is impossible to transition from either of these states to any other state between consecutive months (or periods). Thus, the probability of an account being current this month and remaining in the same state next month is 100%, making the probability of it being in any other state zero. Similarly, if the account is charged off this month, it will remain in the charged-off state next month.

Based on trend analysis, Harmony Finance estimates that 80% of the accounts in early stage will become current by the next month, 10% will remain in early stage, and the remaining 10% will transition to late stage. It is impossible for an account to transition directly from early stage to charged-off status.

The analysis further estimates that 30% of the accounts now in late-stage delinquency will become current by the next month. Another 30% of these late-stage accounts are expected to transition back to early stage, and 30% will remain in late stage. Additionally, 10% of the late-stage accounts are projected to charge off.

Given the properties of absorbing states and the probabilities of transitioning between states, where possible, we can construct the following matrix of transition probabilities:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0.80 & 0 & 0.10 & 0.10 \\ 0.30 & 0.10 & 0.30 & 0.30 \end{bmatrix}$$

where

- Row 1:
  - $P_{11} = 1$ = probability of being in state 1 given state 1 from the prior period
  - $P_{12} = 0$ = probability of being in state 2 given state 1 from the prior period
  - $P_{13} = 0$ = probability of being in state 3 given state 1 from the prior period
  - $P_{14} = 0$ = probability of being in state 4 given state 1 from the prior period
- Row 2:
  - $P_{21} = 0$ = probability of being in state 1 given state 2 from the prior period
  - $P_{22} = 1$ = probability of being in state 2 given state 2 from the prior period
  - $P_{23} = 0$ = probability of being in state 3 given state 2 from the prior period
  - $P_{24} = 0$ = probability of being in state 4 given state 2 from the prior period
- Row 3:
  - $P_{31} = 0.80$ = probability of being in state 1 given state 3 from the prior period
  - $P_{32} = 0$ = probability of being in state 2 given state 3 from the prior period
  - $P_{33} = 0.10$ = probability of being in state 3 given state 3 from the prior period
  - $P_{34} = 0.10$ = probability of being in state 4 given state 3 from the prior period
- Row 4:
  - $P_{41} = 0.30$ = probability of being in state 1 given state 4 from the prior period
  - $P_{42} = 0.10$ = probability of being in state 2 given state 4 from the prior period
  - $P_{43} = 0.30$ = probability of being in state 3 given state 4 from the prior period
  - $P_{44} = 0.30$ = probability of being in state 4 given state 4 from the prior period

This matrix of transition probabilities can be multiplied by a vector of state probabilities to predict the percentage of accounts across the four possible states for any given period.

Eventually, the system will achieve a steady state where all accounts settle into one of two absorbing states: either current or charged off. It is critical for Harmony Finance to project the breakdown between these two absorbing states to effectively manage risk, allocate resources, and strategize for future financial stability. Such analysis requires the use of what is called a *fundamental matrix*.

11.4.1 Obtaining the fundamental matrix

Obtaining the fundamental matrix is a three-step process, and the mathematical complexity increases with each subsequent step. First, we partition the matrix of transition probabilities by extracting each quadrant of probabilities into its own matrix, like so:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.80 & 0 \\ 0.30 & 0.10 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.10 & 0.10 \\ 0.30 & 0.30 \end{bmatrix}$$

where

- I = an identify matrix with diagonal 1s and 0s.
- 0 = a matrix with just 0s.

The fundamental matrix (F) is obtained by taking the inverse of the difference between matrix B and matrix I; it is expressed by the following mathematical equation:

$$F = (I - B)^{-1}$$

We are therefore subtracting matrix B from matrix I as the second step. The superscript −1 indicates that we then take the inverse of the difference to obtain F so that

$$F = (I - B)^{-1}$$

or

$$F = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0.10 & 0.10 \\ 0.30 & 0.30 \end{bmatrix} \right)^{-1}$$

Subtracting matrix B from matrix I returns the following:

$$F = \begin{bmatrix} 0.90 & -0.10 \\ -0.30 & 0.70 \end{bmatrix}^{-1}$$

This is derived by subtracting each corresponding element of matrix B from matrix I:

$$I - B = \begin{bmatrix} 1 - 0.10 & 0 - 0.10 \\ 0 - 0.30 & 1 - 0.30 \end{bmatrix}$$

The third and final step is to get the inverse of this matrix. Fortunately, computing the inverse of a 2×2 matrix is relatively straightforward. However, the complexity increases exponentially as the number of rows and columns grows.

Generalizing the difference between matrix B and matrix I results in the following matrix:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The inverse of this equals

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}$$

or

$$\begin{bmatrix} {}^{d}/_{r} & -{}^{b}/_{r} \\ -{}^{c}/_{r} & {}^{a}/_{r} \end{bmatrix}$$

where $r = ad - bc$. Thus, we get r by subtracting the product of –0.10 and –0.30 from the product of 0.90 and 0.70. Therefore, $r = 0.63 - 0.03 = 0.60$.

The resulting matrix is derived by plugging in 0.60 for r and performing simple division:

$$F = \begin{bmatrix} 0.90 & -0.10 \\ -0.30 & 0.70 \end{bmatrix}^{-1}$$

or

$$F = \begin{bmatrix} {}^{0.70}/_{0.60} & -{}^{(-0.10)}/_{0.60} \\ -{}^{(-0.30)}/_{0.60} & {}^{0.90}/_{0.60} \end{bmatrix}$$

or

$$F = \begin{bmatrix} 1.167 & 0.167 \\ 0.500 & 1.500 \end{bmatrix}$$

From the fundamental matrix, Harmony Finance can accurately project the percentage of accounts that will ultimately settle into a current state and the percentage that will absorb into a charged-off state. This analysis allows Harmony Finance to forecast the long-term behavior of its accounts and make informed decisions based on the predicted distribution of account statuses.

11.4.2 Predicting absorbing states

The first step in making predictions is to multiply the fundamental matrix by matrix A:

$$FA = \begin{bmatrix} 1.167 & 0.167 \\ 0.500 & 1.500 \end{bmatrix} \times \begin{bmatrix} 0.80 & 0 \\ 0.30 & 0.10 \end{bmatrix}$$

or

$$FA = \begin{bmatrix} 0.9837 & 0.0167 \\ 0.8500 & 0.1500 \end{bmatrix}$$

This is derived by following matrix multiplication rules, so that

$$FA = \begin{bmatrix} (1.167)\,(0.80) + (0.167)\,(0.30) & (1.167)\,(0) + (0.167)(0.10) \\ (0.500)\,(0.80) + (1.500)(0.30) & (0.500)\,(0) + (1.500)(0.10) \end{bmatrix}$$

where

- $FA_{11} = (1.167)\,(0.80) + (0.167)\,(0.30) = 0.9336 + 0.0501 = 0.9837$
- $FA_{12} = (1.167)\,(0) + (0.167)\,(0.10) = 0 + 0.0167 = 0.0167$
- $FA_{13} = (0.500)\,(0.80) + (1.500)\,(0.30) = 0.400 + 0.450 = 0.8500$
- $FA_{14} = (0.500)\,(0) + (1.500)\,(0.10) = 0 + 0.150 = 0.1500$

Due to rounding errors, the elements in the top row of the matrix sum to 1.004 instead of 1. To correct this, we apply a manual adjustment by subtracting 0.004 from 0.9837, resulting in the following matrix:

$$FA = \begin{bmatrix} 0.9833 & 0.0167 \\ 0.8500 & 0.1500 \end{bmatrix}$$

Matrix *FA* indicates the probability of an account in one of the two non-absorbing states transitioning to one of the two absorbing states. More precisely, the figures in the top row indicate the probability of an account in early-stage delinquency transitioning to current (98.3%) or charging off (approximately 1.7%); the figures in the bottom row indicate the probability of a late-stage account transitioning to current (85%) versus charging off (15%), which underscores the criticality of Harmony

Finance collecting on accounts in early stage before they advance into late-stage delinquency.

11.4.3 Predicting absorbing states programmatically

Predicting absorbing states manually is a straightforward process, but it involves multiple steps that can be prone to human error. Despite its simplicity, the manual method requires careful attention to detail, which increases the risk of inaccuracies. Fortunately, we can achieve the same results more efficiently and accurately by using Python and specifically NumPy. With just a few lines of code, we can automate the calculations, thereby minimizing errors and saving time.

Most of the code should be familiar to you by now:

```
>>> P = np.array([
>>>     [1, 0, 0, 0],
>>>     [0, 1, 0, 0],
>>>     [0.80, 0, 0.10, 0.10],
>>>     [0.30, 0.10, 0.30, 0.30]
>>> ])

>>> I = np.array([
>>>     [1, 0],
>>>     [0, 1]
>>> ])

>>> A = np.array([
>>>     [0.80, 0],
>>>     [0.30, 0.10]
>>> ])

>>> B = np.array([
>>>     [0.10, 0.10],
>>>     [0.30, 0.30]
>>> ])

>>> I_minus_B = I - B
>>> F = np.linalg.inv(I_minus_B)

>>> FA = np.dot(F, A)

>>> print('The final matrix FA is:')
>>> print(FA)

The final matrix FA is:
[[0.98333333 0.01666667]
 [0.85       0.15      ]]
```

Creates an array that represents the matrix of transition probabilities

Creates an array that is the first submatrix of the transition probabilities matrix

Creates an array that is the second submatrix of the transition probabilities matrix

Creates an array that is the third submatrix of the transition probabilities matrix

Performs element-wise subtraction of matrix B from matrix I

Calculates the inverse of the matrix I_minus_B and assigns it to the variable F

Performs matrix multiplication between the fundamental matrix F and the matrix A, and assigns the resulting matrix to FA

Prints a descriptive message followed by the contents of the matrix FA

The preceding code defines the matrix of transition probabilities P; partitions it into three submatrices denoted I, A, and B; computes the fundamental matrix $F = (I - B)^{-1}$; and then multiplies F by A to get the final matrix FA. The resulting matrix FA is then printed, representing the probabilities of transitioning from non-absorbing states to absorbing states.

Markov analysis provides a powerful framework for predicting future states of a system based on current conditions and transition probabilities. By understanding and applying Markov techniques, we can model complex systems, forecast outcomes, and make informed decisions in various fields such as finance, market analysis, and operations. The ability to predict equilibrium conditions and identify absorbing states further enhances our strategic planning and resource allocation. Mastering these techniques equips us with essential tools to navigate uncertainty and optimize performance in dynamic environments. In the next chapter, we will demonstrate how to examine and test naturally occurring number sequences, which is crucial for identifying patterns and anomalies in data.

Summary

- Markov analysis is a statistical method used to predict the future states of a system based on its current state and the probabilities of transitioning from one state to another. It relies on the assumption that the future state depends only on the present state and not on the sequence of events that preceded it.
- The vector of state probabilities represents the probabilities of a system being in each possible state at a given period. It is used to describe the distribution of states in the system and is essential for predicting future states using Markov analysis.
- The matrix of transition probabilities is a square matrix where each element represents the probability of transitioning from one state to another in a given period. This matrix is fundamental in Markov analysis for determining the likelihood of moving between states and predicting future state distributions.
- To predict future states in Markov analysis, the vector of state probabilities and the transition probability matrix are used together through the mathematical operation known as the dot product. The vector of state probabilities represents the current distribution of the system across various states. By taking the dot product of this vector with the transition probability matrix, which contains the probabilities of moving from one state to another, we can calculate the vector of state probabilities for the next period. Repeatedly applying the dot product operation allows us to forecast the system's state distribution over multiple future periods, effectively predicting how the system will evolve over time.
- In Markov analysis, equilibrium conditions refer to a state where the system's probabilities stabilize and no longer change significantly over time. At equilibrium, the vector of state probabilities remains constant from one period to the next, indicating that the system has reached a steady state where the transitions between states balance out.
- Absorbing states are states that, once entered, cannot be left. When a system reaches an absorbing state, it remains in that state permanently. These states have a transition probability of 1 for remaining in the same state and 0 for transitioning to any other state.

- Markov analysis has diverse real-world applications across industries. It is used in finance for credit risk modeling and stock market analysis, healthcare for patient progression modeling, operations management for equipment maintenance and reliability analysis, and cybersecurity for detecting anomalous network behavior and predicting potential security threats. By applying Markov analysis to these domains, businesses and researchers can make data-driven decisions and optimize long-term outcomes.