

Markov Chain

Markov Property

The Markov property refers to the memoryless property of a stochastic process.

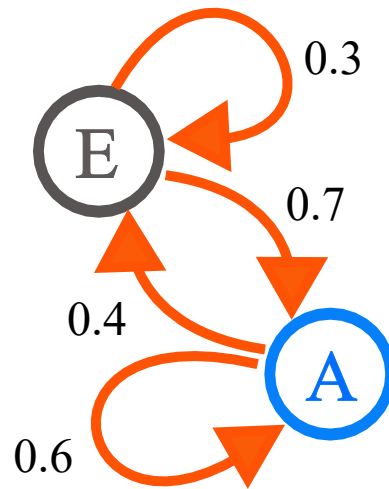
A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it.

In probability theory and related fields, a Markov process is a stochastic process that satisfies the Markov property (sometimes characterized as "memorylessness"). Roughly speaking, a process satisfies the Markov property if one can make predictions for the future of the process based solely on its present state just as well as one could knowing the process's full history, hence independently from such history, that is, conditional on the present state of the system, its future and past states are independent.

What is a Markov Chain

A Markov chain is a type of Markov process that has either a discrete state space or a discrete index set (often representing time), but the precise definition of a Markov chain varies. For example, it is common to define a Markov chain as a Markov process in either discrete or continuous time with a countable state space (thus regardless of the nature of time), but it is also common to define a Markov chain as having discrete time in either countable or continuous state space (thus regardless of the state space).

A state space is the set of all possible configurations of a system. For example, a system in queueing theory defining the number of customers in a line would have state space $\{0, 1, 2, 3, \dots\}$. The changes of state of the system are called transitions. The probabilities associated with various state changes are called transition probabilities. The process is characterized by a state space, a transition matrix describing the probabilities of particular transitions, and an initial state (or initial distribution) across the state space. By convention, we assume all possible states and transitions have been included in the definition of the process, so there is always a next state, and the process does not terminate. A discrete-time random process involves a system which is in a certain state at each step, with the



A diagram representing a two-state Markov process, with the states labelled E and A. Each number represents the probability of the Markov process changing from one state to another state, with the direction indicated by the arrow. For example, if the Markov process is in state A, then the probability it changes to state E is 0.4, while the probability it remains in state A is 0.6.

Examples

If one pops one hundred kernels of popcorn in an oven, each kernel popping at an independent exponentially-distributed time, then this would be a continuous-time Markov process. If X_t denotes the number of kernels which have popped up to time t , the problem can be defined as finding the number of kernels that will pop in some later time. The only thing one needs to know is the number of kernels that have popped prior to the time " t ". It is not necessary to know when they popped, so knowing X_t for previous times " t " is not relevant.

The process described here is an approximation of a Poisson point process – Poisson processes are also Markov processes.

Drunkard's Walk

A famous Markov chain is the so-called "drunkard's walk", a random walk on the number line where, at each step, the position may change by $+1$ or -1 with equal probability. From any position there are two possible transitions, to the next or previous integer. The transition probabilities depend only on the current position, not on the manner in which the position was reached. For example, the transition probabilities from 5 to 4 and 5 to 6 are both 0.5, and all other transition probabilities from 5 are 0. These probabilities are independent of whether the system was previously in 4 or 6.

Suppose that you start with 10 dollars, and you wager 1 dollar on an unending, fair, coin toss indefinitely, or until you lose all of your money. If X_n represents the number of dollars you have after " n " tosses, with $X_0 = 10$, then the sequence $\{X_n : n \in \mathbb{N}\}$ is a Markov process. If I know that you have 12 dollars now, then it would be expected that with even odds, you will either have 11 or 13 dollars after the next toss. This guess is not improved by the added knowledge that you started with 10 dollars, then went up to 11, down to 10, up to 11, and then to 12 dollars.

The process described here is a Markov chain on a countable state space.

Discrete Time Markov Chain

A discrete-time Markov chain is a sequence of random variables " X_1 ", " X_2 ", " X_3 ", ... with the Markov property, namely that the probability of moving to the next state depends only on the present state and not on the previous states:

$$P(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = P(X_{n+1} = x \mid X_n = x_n)$$

The possible values of " X_n " form a countable set " S " called the state space of the chain.

Markov chains are often described by a sequence of directed graphs, where the edges of graph " n " are labeled by the probabilities of going from one state at time " n " to the other states at time " n " + 1, $P(X_{n+1} = x \mid X_n = x_n)$. The same information is represented by the transition matrix from time " n " to time " n " + 1. However, Markov chains are frequently assumed to be time-homogeneous (see variations below), in which case the graph and matrix are independent of " n " and are thus not presented as sequences.

These descriptions highlight the structure of the Markov chain that is independent of the initial distribution $\Pr(X_1 = x_1)$. When time-homogeneous, the chain can be interpreted as a state machine assigning a probability of hopping from each vertex or state to an adjacent one. The probability $\Pr(X_n = x \mid X_1 = x_1)$ of the machine's state can be analyzed as the statistical behavior of the machine with an element x_1 of the state space as input, or as the behavior of the machine with the initial distribution $P(X_1 = y) = (x_1 = y)$ of states as input.

The fact that some sequences of states might have zero probability of occurring corresponds to a graph with multiple connected components, where we omit edges that would carry a zero transition probability. For example, if " a " has a nonzero probability of going to " b ", but " a " and " x " lie in different connected components of the graph, then $P(X_{n+1} = b \mid X_n = a)$ is defined, while $\Pr(X_{n+1} = b \mid X_1 = x, \dots, X_n = a)$ is not.

Variations

1. Time-homogeneous Markov chains (or stationary Markov chains) are processes where

$$\Pr(X_{n+1} = x \mid X_n = y) = \Pr(X_n = x \mid X_{n-1} = y)$$

Transition Matrix

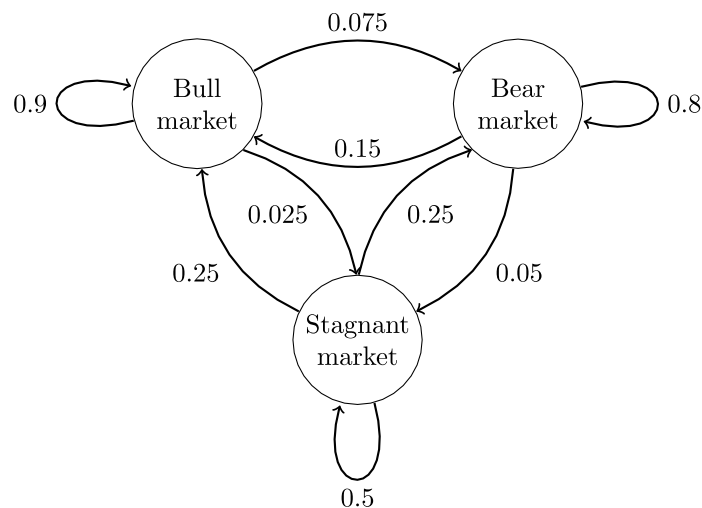
A transition matrix P_t for Markov chain $\{X\}$ at time t is a matrix containing information on the probability of transitioning between states. In particular, given an ordering of a matrix's rows and columns by the state space S , the (i, j) th element of the matrix is given by

$$P_t(ij) = P(X_{t+1} = j | X_t = i)$$

This means each row of the matrix is a probability vector, and the sum of its entries is 1.

Transition matrices have the property that the product of subsequent ones describes a transition along the time interval spanned by the transition matrices. That is to say, $P_0 P_1$ has in its (ij) th position the probability that $X_2 = j$ given that $X_0 = i$. And in general, the (ij) th position of $P_t P_{t+1} \dots P_{t+k}$ is the probability $P(X_{t+k+1} = j | X_t = i)$.

Building A Markov Chain



A state diagram for a simple example is shown in the figure on the right, using a directed graph to picture the state transitions. The states represent whether a hypothetical stock market is exhibiting a bull market, bear market, or stagnant market trend during a given week. According to the figure, a bull week is followed by another bull week 90% of the time, a bear week 7.5% of the time, and a stagnant week the other 2.5% of the time. Labelling the state space $\{1 = \text{bull}, 2 = \text{bear}, 3 = \text{stagnant}\}$ the transition matrix for this example is

$P =$

$$\begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

Using the transition matrix it is possible to calculate, for example, the long-term fraction of weeks during which the market is stagnant, or the average number of weeks it will take to go from a stagnant to a bull market. Using the transition probabilities, the steady-state probabilities indicate that 62.5% of weeks will be in a bull market, 31.25% of weeks will be in a bear market and 6.25% of weeks will be stagnant, since:

$\lim_{N \rightarrow \infty} P^N =$

$$\begin{bmatrix} 0.625 & 0.3125 & 0.0625 \\ 0.625 & 0.3125 & 0.0625 \\ 0.625 & 0.3125 & 0.0625 \end{bmatrix}$$

Probability of States with progressing steps

Consider a set of families. Every family falls into one of the following six states.

The states are ($S_1 = \{AA, AA\}$), ($S_2 = \{AA, Aa\}$), ($S_3 = \{AA, aa\}$), ($S_4 = \{Aa, Aa\}$), ($S_5 = \{Aa, aa\}$) and ($S_6 = \{aa, aa\}$).

The idea is that each pair of parents give birth to two children. For example, the parents ($S_2 = \{AA, Aa\}$) can give birth to $\{\{AA, AA\}, \{AA, Aa\}, \{Aa, Aa\}\}$. Such kind of rules results in the following state transition matrix.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/2 & 0 & 1/4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1/16 & 1/4 & 1/8 & 1/4 & 1/4 & 1/16 \\ 0 & 0 & 0 & 1/4 & 1/2 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The rows mean from which state you start, the columns are the states you can get to.

We will plot all the possible transitions ranging from step size 1 to 20 as the probabilities of reaching a state if we start from state S_3 .

In [15]:

```
import numpy as np
P = np.matrix([[1., 0., 0., 0., 0., 0.],
               [1./4, 1./2, 0., 1./4, 0., 0.],
               [0., 0., 0., 1., 0., 0.],
               [1./16, 1./4, 1./8, 1./4, 1./4, 1./16],
               [0., 0., 0., 1./4, 1./2, 1./4],
               [0., 0., 0., 0., 0., 1.]])

import numpy as np
from matplotlib import pyplot

P = np.matrix([[1., 0., 0., 0., 0., 0.],
               [1./4, 1./2, 0., 1./4, 0., 0.],
               [0., 0., 0., 1., 0., 0.],
               [1./16, 1./4, 1./8, 1./4, 1./4, 1./16],
               [0., 0., 0., 1./4, 1./2, 1./4],
               [0., 0., 0., 0., 0., 1.]])

v = np.matrix([[0, 0, 1, 0, 0, 0]])

# Get the data
plot_data = []
for step in range(20):
    result = v * P**step
    plot_data.append(np.array(result).flatten())

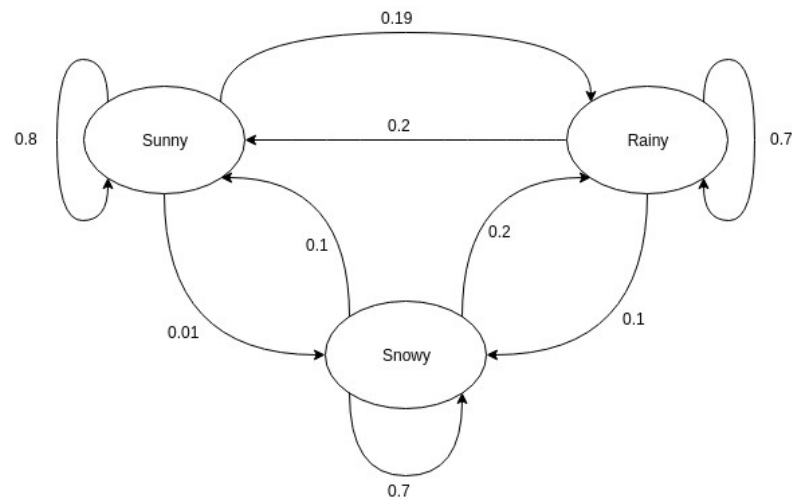
# Convert the data format
plot_data = np.array(plot_data)

# Create the plot
pyplot.figure(1)
pyplot.xlabel('Steps')
pyplot.ylabel('Probability')
lines = []
for i, shape in zip(range(6), ['x', 'h', 'H', 's', '8', 'r+']):
    line, = pyplot.plot(plot_data[:, i], shape, label="S%i" % (i+1))
    lines.append(line)
pyplot.legend(handles=lines, loc=1)
pyplot.show()
```

From the plot, we can observe that the probabilities of reaching states S1 and S6 initially increased and attained a maximum around 0.5 while the probabilities of all other states decreased and became 0. Therefore, if we start from state 3, and go on taking steps we will eventually end either in state S1 or state S6 with both having equal probabilities.

Implementation of Markov Chain Model

As a simple example, take a look at predicting the weather to understand this representation better. Consider that there are three possible states of the random variable Weather = {Sunny, Rainy, Snowy}.



In the above Markov chain, consider that the observed state of the current random variable is Sunny. Then, the probability that the random variable at the next time instance will also take the value Sunny is 0.8. It could also take the value Rainy with a probability of 0.19, or Snowy with a probability of 0.01. One thing to note here is that the sum of all the probability values on all the outward edges from any state should equal 1, since it's an exhaustive event.

In [2]:

```
import numpy as np

class MarkovChain(object):
    def __init__(self, transition_prob):
        """
        Initialize the MarkovChain instance.

        Parameters
        -----
        transition_prob: dict
            A dict object representing the transition
            probabilities in Markov Chain.
            Should be of the form:
            {'state1': {'state1': 0.1, 'state2': 0.4},
             'state2': {...}}
        """
        self.transition_prob = transition_prob
        self.states = list(transition_prob.keys())

    def next_state(self, current_state):
        """
        Returns the state of the random variable at the next time
        instance.

        Parameters
        -----
        current_state: str
            The current state of the system.
        """
        return np.random.choice(
            self.states,
            p=[self.transition_prob[current_state][next_state]
              for next_state in self.states]
        )

    def generate_states(self, current_state, no=10):
        """
        Generates the next states of the system.

        Parameters
        -----
        current_state: str
            The state of the current random variable.

        no: int
            The number of future states to generate.
        """
        future_states = []
        for i in range(no):
            next_state = self.next_state(current_state)
            future_states.append(next_state)
            current_state = next_state
        return future_states
```


In [8]:

```
transition_prob = {'Sunny': {'Sunny': 0.8, 'Rainy': 0.19,  
                             'Snowy': 0.01},  
                  'Rainy': {'Sunny': 0.2, 'Rainy': 0.7,  
                             'Snowy': 0.1},  
                  'Snowy': {'Sunny': 0.1, 'Rainy': 0.2,  
                             'Snowy': 0.7}}
```

```
weather_chain = MarkovChain(transition_prob=transition_prob)  
weather_chain.next_state(current_state='Sunny')  
weather_chain.generate_states(current_state='Snowy', no=10)
```

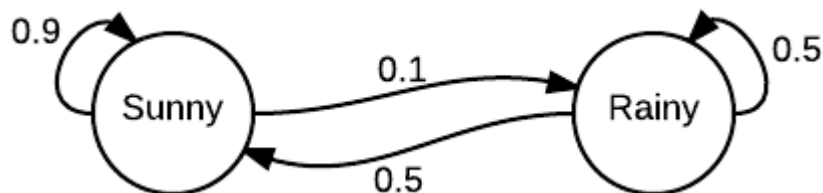
Out[8]:

```
['Snowy',  
 'Snowy',  
 'Snowy',  
 'Rainy',  
 'Rainy',  
 'Rainy',  
 'Rainy',  
 'Sunny',  
 'Sunny',  
 'Sunny']
```

Questionnaire

1. Suppose that there is a coin purse containing five quarters (each worth 25¢), five dimes (each worth 10¢), and five nickels (each worth 5¢), and one by one, coins are randomly drawn from the purse and are set on a table. If X_n represents the total value of the coins set on the table after n draws, with $X_0 = 0$, then is the sequence $\{X_n : n \in \mathbb{N}\}$ a Markov process?

2. Predicting the weather. The weather on day 0 (today) is known to be sunny. There are two kinds of weather - "sunny" "rainy". Based on this information predict tomorrow's and day after tomorrow's weather



Solutions (https://github.com/ebi-byte/kt/blob/master/supervised_ML/Markov%20Chain%20Solutions.ipynb)