Published in Towards Data Science

Juan Nathaniel

May 6, 2021 · 4 min read ★ ·

Listen

Introduction to Markov Chain Programming

What are Markov Chains, why are they useful, and how to implement them

Imagine the following scenario: you want to know whether the weather tomorrow is going to be sunny or rainy. Now, you might have a natural intuition based on your experience and historical observation of the weather. If for the past 1 week the weather has

been sunny, then you have a 90% certainty that tomorrow will also be sunny. But if for the previous week or so the weather has been rainy, then the probability of tomorrow being sunny does not look too good: only at 50% chance. This scenario can be described as a ***Markov Chain*** process.

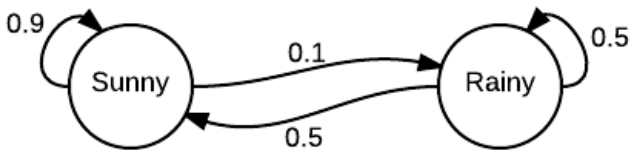
Photo by Shaojie on Unsplash

What is Markov Chain?

But what is a Markov Chain, formally? Markov Chain is a mathematical system that describes a collection of transitions from one state to the other according to certain stochastic or probabilistic rules.

Take for example our earlier scenario for predicting the next day's weather. If today's weather is sunny, then based on our (reliable) experience, the probability for the weather tomorrow transitioning to rainy is 10% and sunny 90%. On the other hand, if at present the weather is rainy, then the probability for tomorrow remaining rainy is 50% and sunny 50%.

> *These changes (or the lack thereof) between different states are called transitions while the variable of interest (ie. rainy or sunny) are called states.*



Weather transitions that follow Markov rule

For these transitions, however, to be qualified as Markov Chain, they must satisfy the **Markov Property**. The property states that the probability of transition is entirely dependent only on the current

state, and not on the preceding set of sequences. This characteristic allows Markov Chain to be **memory-less**.

Why Markov Chain?

Now that we have covered what is actually Markov Chain, let's discuss *why* is it worthwhile to get familiar with. Markov Chain has many applications in the real-world processes, such as in game theory, physics, economics, signal processing, information theory, and many more.

Furthermore, this seemingly simple process serves as the basis for many more complex stochastic simulation methods such as Markov Chain Monte Carlo (MCMC) or Hidden Markov Chain. Also, Markov Chain is a precursor for many of the modern data science techniques, such as being one of the building blocks for Bayesian statistics.

In all, Markov Chain will serve as a

good starting point for you to understand more advanced statistical modelling techniques in data science. It is imperative, therefore, to get your hands dirty by understanding the basics of Markov Process through a mathematical understanding and coding its algorithm implementation.

More About Markov Chain: Mathematical Definition

The Markov Chain model represents the probabilities for state transitions as a ***transition matrix***. If the system has **N** possible states (eg. **N=2** for our weather prediction case), then the transition matrix will have a **N x N** shaped transition matrix. Subsequently, the individual entry for the matrix, **N(i, j)**, will indicate the probability of transition between **state i and state j**.

For our weather prediction case, the transition matrix, ***T*** can be illustrated as:

Juan Nathaniel

537 Followers

Engineering @ Columbia University | Documenting and sharing my learning journey through AI, programming, and research

Follow

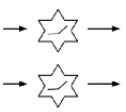
More from Medium

Jaco... in To...

Understanding The Derivative Of The...

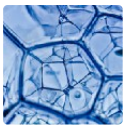
Jas... in MLe...

Two Steps Forward, Two Steps Back



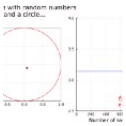
M... in Towar...

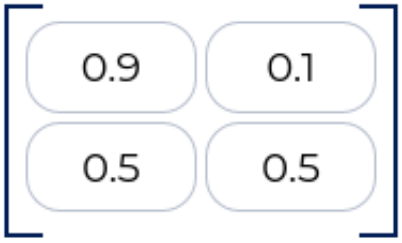
AlphaFold for Everybody



Be... in Tow...

Estimate pi using random numbers





Transition Matrix, **T**, for the weather prediction problem

> *What will happen if you want to determine the probability over multiple steps, say for the weather to rain over the next* M *days? You can simply raise the transition probability to the power of M.*

Programming Markov Chain

Let's try to code the above example in Python.

1. Import the necessary libraries

```
import numpy as np
import random as rm
```

2. Define the states and their probabilities (**NOTE**: ensure that the total probability in each row sums up to 1)

```
states = ["sunny", rainy"]
transitions = [["SS",
"SR"],["RS", "RR"]]

T = [[0.9, 0.1],[0.5, 0.5]]
```

3. Lets write the (rather tedious) Markov Chain function to predict the weather for the next **n** number of days! (**NOTE**: you should probably look for better Python library that abstracts the implementation of Markov Chain).

```
def weather_forecast(n_days, weath
    weather_list = [weather_today]
    n = 0
    prob = 1.0
    while n != n_days:
        if weather_today == "sunny
            change = np.random.cho
            if change == "SS":
                prob = prob * T[0]
                weather_list.appen
            else:
                prob = prob * T[0]
                weather_list.appen

        else:
            change = np.random.cho
            if change == "RS":
                prob = prob * T[1]
                weather_list.appen
            else:
                prob = prob * T[1]
                weather_list.appen
        n = n + 1
    return weather_list
```

markov_chain.py hosted with ❤ by GitHub view raw

4. Run the program, say, for the next 5 days.

```
future_weathers =
weather_forecast(n_days =
5)
```

Conclusion

We have discussed what is Markov Chain descriptively and formally, and why is it important to learn its basic principle. We have also coded a very basic Markov process from scratch. Now that you are familiar with how a Markov Chain works, you can deep dive into more complex stochastic modelling techniques, such as the Hidden Markov Process or MCMC. Stay tuned for more of these follow-up contents!

Do subscribe to my Email

newsletter: *https://tinyurl.com/2npw2fnz* ***where I regularly summarize AI research papers in plain English and beautiful visualization.***