Markov Chains

Introduction

Have you ever wanted to forecast weather? It's all about probability and statistics. In this assignment you will build a simple forecast model based of the probaility of transition from one weather type to another.

The Model

Assume we just have these four kinds of weather:

- Sunny (Su)
- Cloudy (C)
- Rainy (R)
- Snowy (Sn)

Assume further the following transistion probabilities:

- Su -> Su: 0.65 (that is, after a sunny day there is a probability of 65 % that the weather tomorrow is sunny too)
- Su -> C: 0.22
- Su -> R: 0.11
- Su -> Sn: 0.02

...

- C -> Su: 0.13
- C -> C: 0.55
- C -> R: 0.25
- C -> Sn: 0.07

...

- R -> Su: 0.08
- R -> C: 0.65
- R -> R: 0.25
- R -> Sn: 0.02

...

- Sn -> Su: 0.27
- Sn -> C: 0.55
- Sn -> R: 0.03

- Sn -> Sn: 0.15

(*Note: these probabilities are just taken by random and have nothing to do with reality.*)

The Forecast - Matrix Multiplication approach

We have two ways to make the prediction for the upcomming weather types: by performing calculations and by make a simulation. If the weather type today (day 0) is Su, then we know the probability for each type tomorrow (day 1). But the day after tomorrow (day 2), we need to take to account the probilities for each type on day 1. The probability for sun on day 2, given type Su today, $P(\mathrm{Su2})$ becomes

$$P(\mathrm{Su2}) = P(\mathrm{Su} \text{ -> } \mathrm{Su}) + P(\mathrm{C} \text{ -> } \mathrm{Su}) + P(\mathrm{R} \text{ -> } \mathrm{Su}) + P(\mathrm{Sn} \text{ -> } \mathrm{Su}) =$$

$$= 0.65 \cdot 0.65 + 0.22 \cdot 0.13 + 0.11 \cdot 0.08 + 0.02 \cdot 0.27 = 0.4653 = 46.53$$

,

We may continue to calculate the probability for the other types on day 2, and use these results in the calculation of the probability for a sunny day 3. And so on.

We summarize the probabilities in a matrix like

$$T = \begin{bmatrix} 0.65 & 0.13 & 0.08 & 0.27 \\ 0.22 & 0.55 & 0.65 & 0.55 \\ 0.11 & 0.25 & 0.25 & 0.03 \\ 0.02 & 0.07 & 0.02 & 0.15 \end{bmatrix}$$

where each row corresponds to the probability given the column of the current weather type in order Su, C, R, Sn. For example: given cloudy weather today (column 2) means 0.13 chance for sun tomorrow (row 1).

Well, the weather today is a fact, not a forecast. We can for sure say it's (for example) sunny today. The probability for each type of weather tomorrow is the given by

$$T_1 = T \cdot I = \begin{bmatrix} 0.65 & 0.13 & 0.08 & 0.27 \\ 0.22 & 0.55 & 0.65 & 0.55 \\ 0.11 & 0.25 & 0.25 & 0.03 \\ 0.02 & 0.07 & 0.02 & 0.15 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.65 \\ 0.22 \\ 0.11 \\ 0.02 \end{bmatrix}$$

And the probability for the weather the day after tomorrow is given by

$$T_2 = T \cdot T_1 = \begin{bmatrix} 0.65 & 0.13 & 0.08 & 0.27 \\ 0.22 & 0.55 & 0.65 & 0.55 \\ 0.11 & 0.25 & 0.25 & 0.03 \\ 0.02 & 0.07 & 0.02 & 0.15 \end{bmatrix} \begin{bmatrix} 0.65 \\ 0.22 \\ 0.11 \\ 0.02 \end{bmatrix} = \begin{bmatrix} 0.4653 \\ 0.3465 \\ 0.1546 \\ 0.0336 \end{bmatrix}$$

and so on. For example, we read in the first row in the resulting column vector, 0.4653. That is, the chance for sun on day 2 is 46.53 %, as calculated above. The sum of each column equals 1, meaning that's
a 100 % chance that the weather tomorrow is one the four possible kinds.

The system of the described random transistions from one state to another is called a **Markov Chain**. We can calculate the probability for a given end state as above, with repeated multiplication of stocastic matrices. A stocastic matrix is a matrix with elements representing probabilities, summed up to 1 in it's rows or columns.

The Forecast - Simulation approach

Another method to predict the weather by probability is to **simulate** the Markov Chain. `NumPy` has a method `random.choice()` (documentation here). It does what we can expect: returns a random element from a list. Without options, the the chance for each element to be chosen is equal. But there are options, one option is that we can **weight** the probability for each pick. See sample code below:

```
1 import numpy as np
2 np.random.seed(1000) # For reproducibility
3 weatherStates = np.array(['Sunny', 'Cloudy', 'Rainy', 'Snowy'])
4 weights = np.array([0.65, 0.22, 0.11, 0.02]) # The weighs in same
  order as elements
5 np.random.choice(weatherStates, p=weights) # outputs 'Cloudy' with
  given seed
```

This gives 65 % chance for 'Sunny' to be chosen, 22 % chance for 'Cloudy', 11 % chance for 'Rainy' and finally 2 % chance for 'Snowy'. This weight option is for us, and I will not tell more about the details of the simulation implementation.

The Tasks

Given the weather types and transistion probabilities above, calculate the probability for each kind of weather after

- 3 days
- 5 days

- 10 days
- 100 days

The calculation shall be done with

1. matrix multiplication, as well as
2. a simulation

Comment the result.