TP Chaînes de Markov et Modèles de Markov Cachés

Pierre Chambet

November 24, 2025

1 Introduction

2 Markov modelling of rainy and dry periods

In this first part we model the alternation between dry and rainy days by a two–state Markov chain. The goal is to (i) make the link between the mathematical model and a concrete simulation, (ii) study the distribution of dry/rainy spell lengths, and (iii) compare the model with the measured data provided in `RR5MN.mat`.

2.1 Definition of the Markov chain (Questions 1–2)

We consider a discrete–time stochastic process $(X_t)_{t\geq 0}$ taking values in the state space

$$\mathcal{E} = \{E_0, E_1\} = \{\mathrm{dry}, \mathrm{rain}\}.$$

By assumption (X_t) is a homogeneous Markov chain:

$$\mathbb{P}(X_{t+1} = j \mid X_t = i, X_{t-1}, \dots, X_0) = \mathbb{P}(X_{t+1} = j \mid X_t = i) \quad \text{for all } i, j \in \{0, 1\}.$$

The model is parametrised by three probabilities:

- α: probability to remain in the same dry state, $\mathbb{P}(\mathrm{dry}_{t+1} \mid \mathrm{dry}_t) = \alpha$;
- β: probability to remain in the same rainy state, $\mathbb{P}(\mathrm{rain}_{t+1} \mid \mathrm{rain}_t) = \beta$;
- γ: initial probability of a dry day, $\mathbb{P}(X_0 = \mathrm{dry}) = \gamma$.

With this convention, the transition matrix A of the chain is

$$A = \begin{pmatrix} \mathbb{P}(E_0 \to E_0) & \mathbb{P}(E_0 \to E_1) \\ \mathbb{P}(E_1 \to E_0) & \mathbb{P}(E_1 \to E_1) \end{pmatrix} = \begin{pmatrix} \alpha & 1-\alpha \\ 1-\beta & \beta \end{pmatrix}.$$

The initial distribution is

$$\boldsymbol{\pi}_0 = \big(\mathbb{P}(X_0 = E_0), \mathbb{P}(X_0 = E_1)\big) = (\gamma,\ 1-\gamma).$$

In the graphical representation (Question 2), the chain can be drawn as two nodes “dry” and “rain” with four directed edges labelled by the transition probabilities:

$$\text{dry} \xrightarrow[\text{stay dry}]{\alpha} \text{dry}, \qquad \text{dry} \xrightarrow[\text{becomes rainy}]{1-\alpha} \text{rain},$$

$$\text{rain} \xrightarrow[\text{becomes dry}]{1-\beta} \text{dry}, \qquad \text{rain} \xrightarrow[\text{stay rainy}]{\beta} \text{rain}.$$

A zero entry in the matrix A would correspond to a *forbidden transition*: an arrow that does not exist in the graph.

In this first exercise we assume that the observation is equal to the state: on each day we observe $Y_t = X_t$, so no extra emission probabilities are needed.

2.2 Stationary probability of a rainy day (Question 3)

We are interested in the long–term probability of being in the rainy state, that is the second component of the stationary distribution $\boldsymbol{\pi}^\star = (\pi_0^\star, \pi_1^\star)$ satisfying

$$\boldsymbol{\pi}^\star = \boldsymbol{\pi}^\star A, \qquad \pi_0^\star + \pi_1^\star = 1.$$

Writing the first equation componentwise gives

$$\pi_0^\star = \pi_0^\star \alpha + \pi_1^\star (1-\beta), \qquad \pi_1^\star = \pi_0^\star (1-\alpha) + \pi_1^\star \beta.$$

Using the constraint $\pi_0^\star = 1 - \pi_1^\star$ and substituting into the second equation yields

$$\begin{aligned} \pi_1^\star &= (1-\pi_1^\star)(1-\alpha) + \pi_1^\star \beta \\ &= (1-\alpha) - (1-\alpha)\,\pi_1^\star + \beta \pi_1^\star. \end{aligned}$$

We regroup all terms containing $\pi_1^\star$ on the left:

$$\pi_1^\star + (1-\alpha)\pi_1^\star - \beta\pi_1^\star = 1-\alpha,$$

so that

$$\pi_1^\star\bigl(1 + 1 - \alpha - \beta\bigr) = 1 - \alpha \quad \Longrightarrow \quad \pi_1^\star = \frac{1-\alpha}{2-\alpha-\beta}.$$

If we instead parameterise the chain using the transition probabilities $p = \mathbb{P}(\text{dry} \to \text{rain})$ and $q = \mathbb{P}(\text{rain} \to \text{dry})$, that is

$$A = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix},$$

then the same computation gives the well–known formula

$$\mathbb{P}(X_t = \text{rain in stationarity}) = \pi_1^\star = \frac{p}{p+q}.$$

In the notebook we verify this numerically with a short piece of code:

```
alpha = 0.65
beta  = 0.02

A = np.array([[alpha, 1-alpha],
              [1-beta, beta]])
pi_star = np.array([1-alpha, 1-beta])
pi_star = pi_star / pi_star.sum()   # normalize, just to check

# Stationary distribution computed by formula
pi1_star_formula = (1.0 - alpha) / (2.0 - alpha - beta)

print("A␣=\n", A)
print("Stationary␣distribution␣(formula):", pi1_star_formula)
print("Check␣pi*␣A:", pi_star @ A)
```

2.3 Simulation of the chain (Question 4)

To check the theoretical value of the stationary rainfall probability, we simulate a long trajectory of the Markov chain and estimate the empirical frequency of rainy days.

We use a pure NumPy implementation of the Markov chain; because the observation is equal to the state, we only simulate (X_t).

```
def simulate_markov(T, start_prob, Nsamples, seed=None):
    """
␣␣␣␣Simulate␣a␣two-state␣Markov␣chain␣with␣transition␣matrix␣T
␣␣␣␣and␣initial␣distribution␣start_prob.
␣␣␣␣State␣0␣=␣dry,␣state␣1␣=␣rain.
␣␣␣␣"""
    if seed is not None:
        rng = np.random.default_rng(seed)
        choice = rng.choice
    else:
        choice = np.random.choice

    states = np.zeros(Nsamples, dtype=int)
    obs    = np.zeros(Nsamples, dtype=int)

    # initial state X_0
    states[0] = choice([0, 1], p=start_prob)
    obs[0]    = states[0]

    # evolution X_{t+1} ~ T[X_t, :]
    for t in range(1, Nsamples):
        prev = states[t-1]
        states[t] = choice([0, 1], p=T[prev])
        obs[t]    = states[t]

    return obs, states
```

```
# Parameters of the model
gamma = 0.95           # P(X0 = dry)
alpha = 0.65           # P(dry -> dry)
beta  = 0.02           # P(rain -> rain)
Nsamples = 10000

pi0 = np.array([gamma, 1-gamma])
T   = np.array([[alpha,     1.0-alpha],
                [1.0-beta,  beta]])

# Simulation
obs_sim, states_sim = simulate_markov(T, pi0, Nsamples)

# Empirical rainfall frequency
freq_rain_emp = np.mean(obs_sim == 1)
print("Empirical frequency of rain:", freq_rain_emp)
```

For a large number of samples (`Nsamples` in the order of 10^4–10^5), the empirical frequency of rainy days `freq_rain_emp` is very close to the theoretical stationary probability $\pi_1^\star$ computed in the previous subsection.

2.4 Distribution of dry and rainy spell lengths (Question 5)

A key advantage of the Markov model is that it gives a simple description of the *spell lengths*, that is, the duration of consecutive dry or rainy periods.

Let D_{rain} denote the length, in time steps, of a rainy spell. As soon as the chain enters state "rain", it will remain in that state until it transitions back to "dry". If we use the parametrisation

$$A = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix},$$

then a rainy spell ends when the transition rain→dry occurs, which happens with probability q at each step. It follows that D_{rain} has a geometric distribution:

$$\mathbb{P}(D_{\text{rain}} = k) = (1-q)^{k-1}\, q, \qquad k \in \{1, 2, 3, \dots\}.$$

Similarly the dry spell length D_{dry} is geometrically distributed with parameter p:

$$\mathbb{P}(D_{\text{dry}} = k) = (1-p)^{k-1}\, p.$$

The corresponding expectations are

$$\mathbb{E}[D_{\text{rain}}] = \frac{1}{q}, \qquad \mathbb{E}[D_{\text{dry}}] = \frac{1}{p}.$$

On the simulated sequence `obs_sim` we use the provided function `duree` to extract the durations of dry and rainy spells:

```
dSec, dRain, pdfSec, pdfRain, binsSec, binsRain = duree(obs_sim)

print("Number of dry spells (simulated):", len(dSec))
print("Number of rainy spells (simulated):", len(dRain))
print("Empirical mean rainy spell length:", dRain.mean())
```

We can compare the empirical mean and variance of D_{rain} and D_{dry} to the theoretical values $1/q$ and $1/p$. We can also compare the full empirical distributions to the geometric law on a semi–log plot.

To plot the empirical probability mass functions, the helper `duree` returns normalised histograms `pdfSec` and `pdfRain` together with the bin edges. Since the edges have one more element than the pdf, we use bin centres on the x–axis:

```
def centers_from_bins(bins):
    return (bins[:-1] + bins[1:]) / 2

centers_rain = centers_from_bins(binsRain)
plt.figure()
plt.bar(centers_rain, pdfRain, width=1, alpha=0.5,
        label="Simulated rainy spells")
plt.xlabel("Length of rainy periods")
plt.ylabel("Probability")
plt.title("Empirical distribution of rainy spell lengths")
plt.legend()
plt.show()
```

2.5 Comparison with measured data (Question 6)

Finally we repeat the spell–length analysis on the measured data contained in `RR5MN.mat`. The file provides a vector `Support` of zeros and ones indicating dry and rainy days. We load it and apply the same `duree` function:

```
mat = sio.loadmat("../data/RR5MN.mat")
ObsMesure = mat["Support"].astype(np.int8).squeeze() - 1
# 0=dry, 1=rain

dSecMes, dRainMes, pdfSecMes, pdfRainMes, binsSecMes, binsRainMes \
    = duree(ObsMesure)

print("Number of dry spells (measured):", len(dSecMes))
print("Number of rainy spells (measured):", len(dRainMes))
```

We then superimpose the simulated and measured spell–length distributions. Using bin centres for both histograms:

```
centers_rain_sim = centers_from_bins(binsRain)
centers_rain_mes = centers_from_bins(binsRainMes)

plt.figure()
```

```
plt.bar(centers_rain_sim, pdfRain, width=1, alpha=0.5,
        label="Rainy spells (simulated)")
plt.bar(centers_rain_mes, pdfRainMes, width=1, alpha=0.5,
        label="Rainy spells (measured)")
plt.xlabel("Length of rainy periods")
plt.ylabel("Probability")
plt.title("Durations of rainy periods: simulated vs. measured")
plt.legend()
plt.show()
```

On the plots we observe that the simple two–state Markov chain with fixed parameters does not perfectly reproduce the empirical distributions: for instance, in our experiments the model tends to produce longer rainy spells than those observed in the real data. This suggests that, while the Markov chain is a useful first approximation, a more refined model (for example with parameters estimated from the data, or with additional hidden states) would be needed for a more accurate fit.

3 Partie 2 : HMM à 3 états pour la pluie

4 Partie 3 : Reconnaissance de mots par HMM gaussiens

5 Conclusion