MASTER TRIED - RMCMC

MARKOV CHAINS AND HIDDEN MARKOV MODELS

CARL ROBINSON & NELSON FERNANDEZ-PINTO 15 February 2018

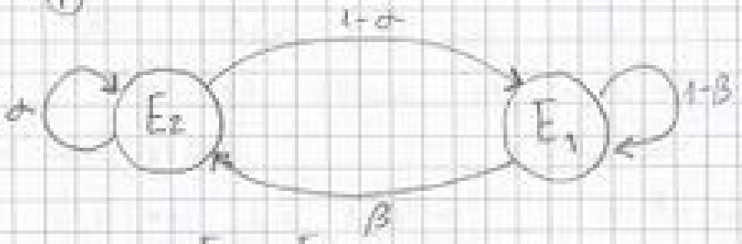
Exercise 1.

EXO 1 : Pleut-il ?

$$E = \{ E_1 , E_2 \}$$

E_1: Not raining, E_2: raining

①

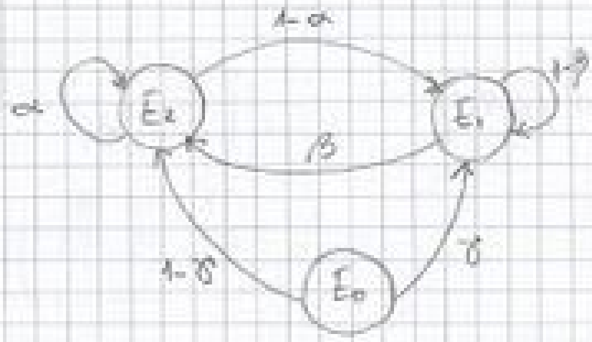


$$\lambda = \{ E, A, \pi_0 \}$$

$$A = \begin{array}{c|cc} & E_1 & E_2 \\ \hline E_1 & 1-\beta & \beta \\ E_2 & 1-\alpha & \alpha \end{array}$$

$$\pi_0 = (\gamma, 1-\gamma)$$

1.3



$$A = \begin{array}{c|ccc} & E_0 & E_1 & E_2 \\ \hline E_0 & 0 & \gamma & 1-\gamma \\ E_1 & 0 & 1-\beta & \beta \\ E_2 & 0 & 1-\alpha & \alpha \end{array}$$

$$\pi_0 = (1, 0, 0)$$

1.5

$$P(E_2) = P(E_2/E_2)P(E_2) + P(E_2/E_1)P(E_1)$$

$$P(E_2) = P(E_2/E_2)P(E_2) + P(E_2/E_1)(1 - P(E_2))$$

$$P(E_2) = P(E_2/E_2)P(E_2) + P(E_2/E_1) - P(E_2/E_1)P(E_2)$$

$$P(E_2)[1 - P(E_2/E_2) + P(E_2/E_1)] = P(E_2/E_1)$$

$$P(E_2) = \frac{P(E_2/E_1)}{[1 - P(E_2/E_2) + P(E_2/E_1)]} = \frac{\beta}{1+\beta-\alpha}$$

1.6 $P(D_{pluie} = D)$ Proof by induction

$$P(D_{pluie} = 2) = P(E_2/E_2)\ P(E_2/E_1) = \alpha \cdot (1-\alpha)$$

$$P(D_{pluie} = 3) = P(E_2/E_2) \cdot P(E_2/E_2) \cdot P(E_2/E_1) = \alpha^2(1-$$

$$P(D_{pluie} = D) = (1-\alpha) \cdot \alpha^{D-1} \quad \vdots$$

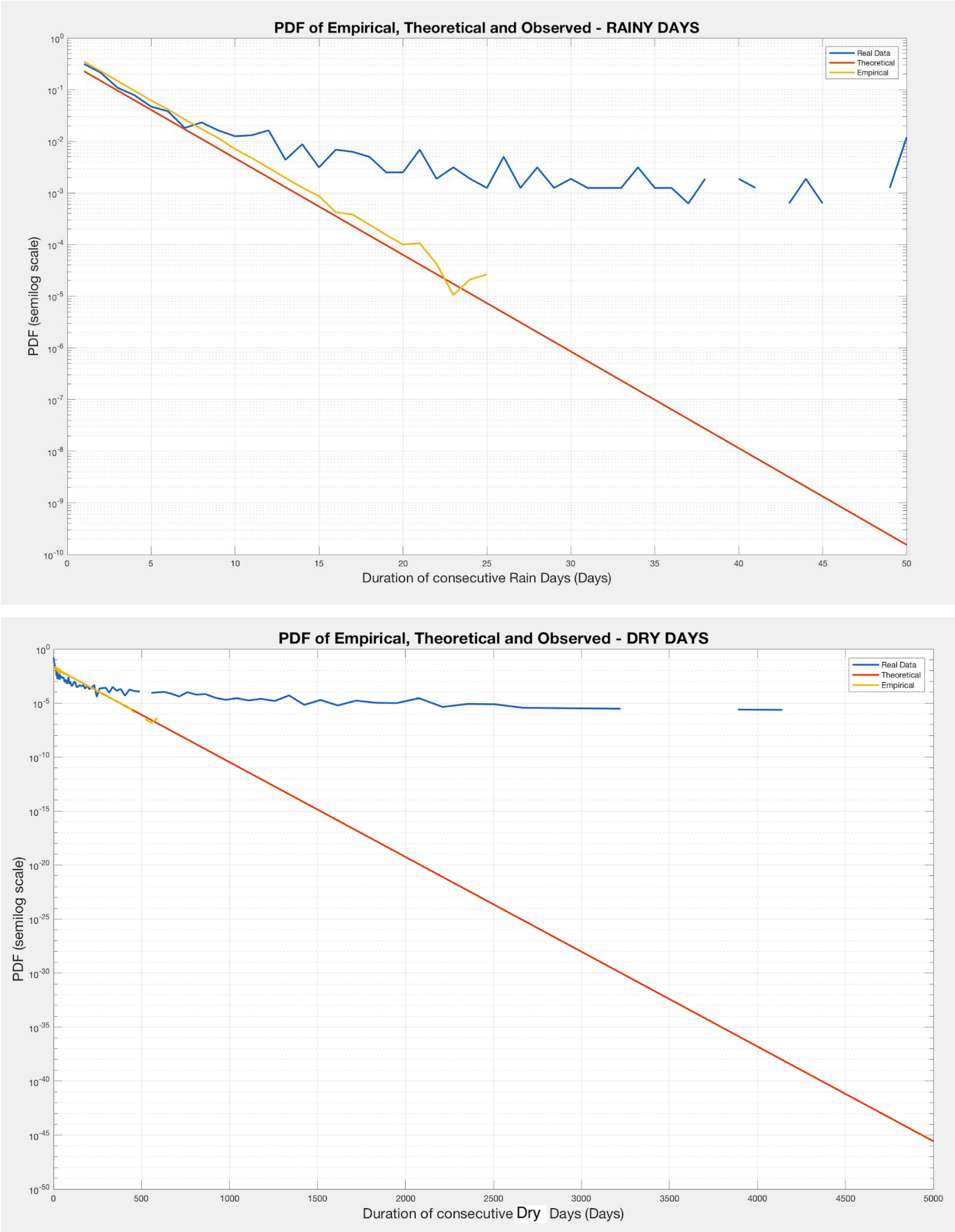


Figure 1. Empirical, theoretical and real distributions by duration Markov Chain.

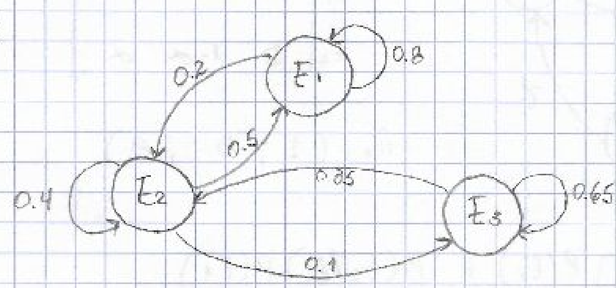
(1.9) the empyrical distribution obtained for rain state only match the theoretical in durations lower than 30 mins. After important discrepancies appear, the model underestimates long rainy events. In the case of dry state, the behaviour is worse, the model does not match the observed probability density function in any duration. Finally, we conclude that a two - states markov chain model is not appropiate to modelise the observed sequences.

Exercise 2.

Exo 2 Hidden Markov Model (HMM):

$$S = \{ \overset{\text{no rain}}{S_1}, \overset{\text{rain}}{S_2} \}$$

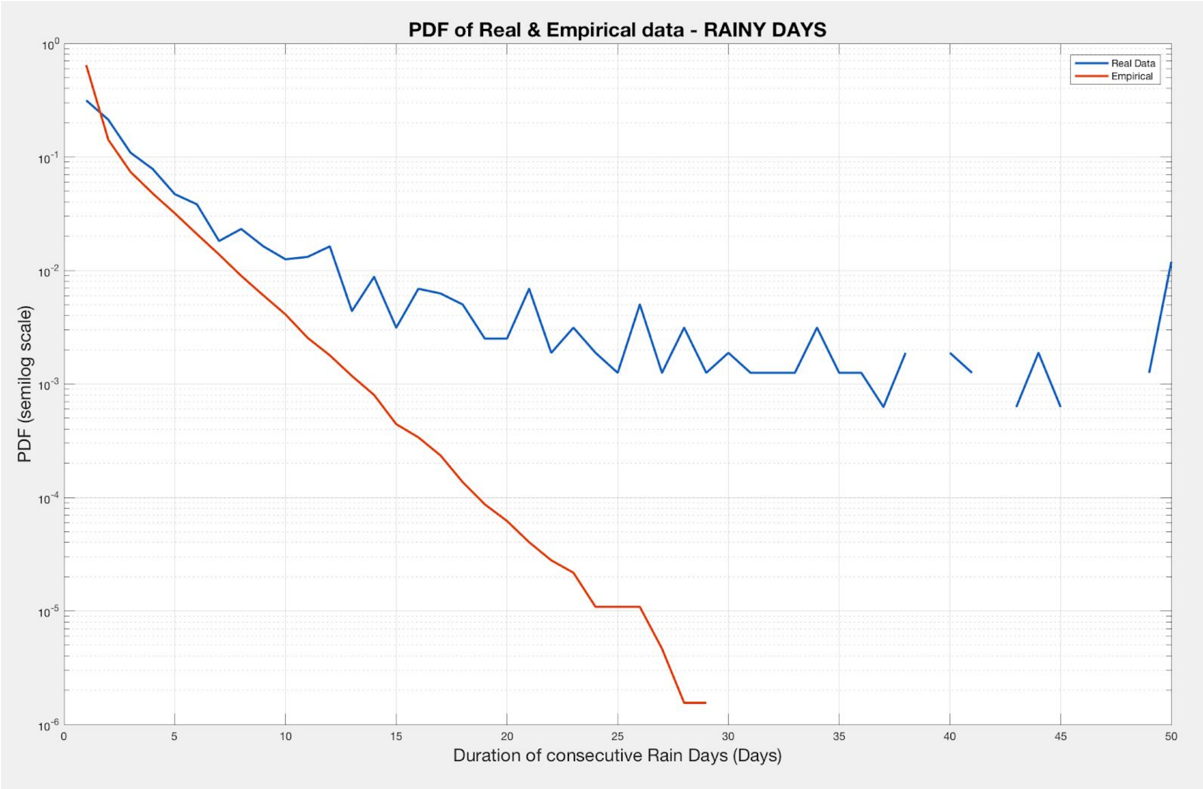
(1)



$$A = \begin{pmatrix} 0.8 & 0.2 & 0 \\ 0.5 & 0.4 & 0.1 \\ 0 & 0.35 & 0.65 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 & 0 \\ 0.8 & 0.2 \\ 0 & 1 \end{pmatrix}$$

(2) the proportions obtained using the simulated sequence are 87,34 % dry and 12,14 % rain. With real data we obtain 95,86 % of dry and 4,19% rain. then, the simulated sequence maintains an acceptable proportion among rain and dry states. However, when we look at the pdf, important differences appear, the model does not match the duration of the states. this is logical as we are using a-priori probabilities in A and B.



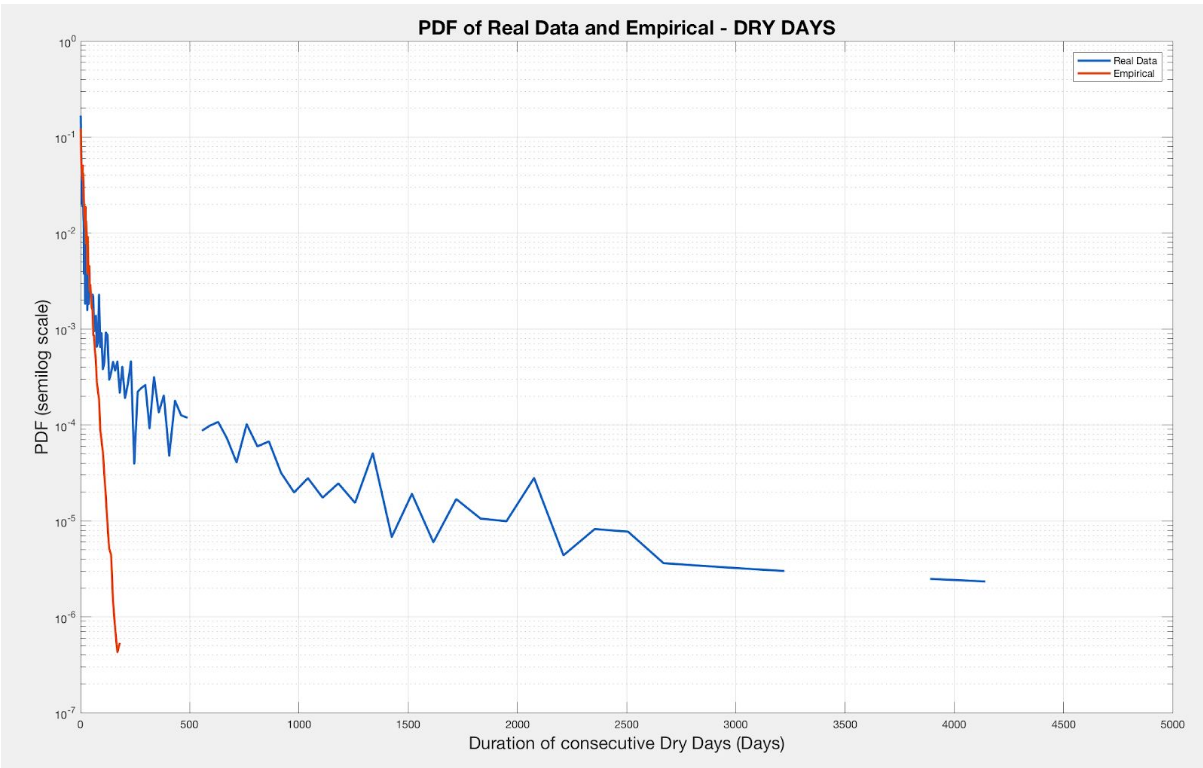


Figure 2. Empirical and real distributions by duration using arbitrary transition and emission probabilities (HMM)

(3) We now use the algorithm of Baum Welch to estimate the transition and emission matrix coefficients. To do that, we use a sequence of real observations and provide the optimal coefficients of the model:

Estimated coefficients

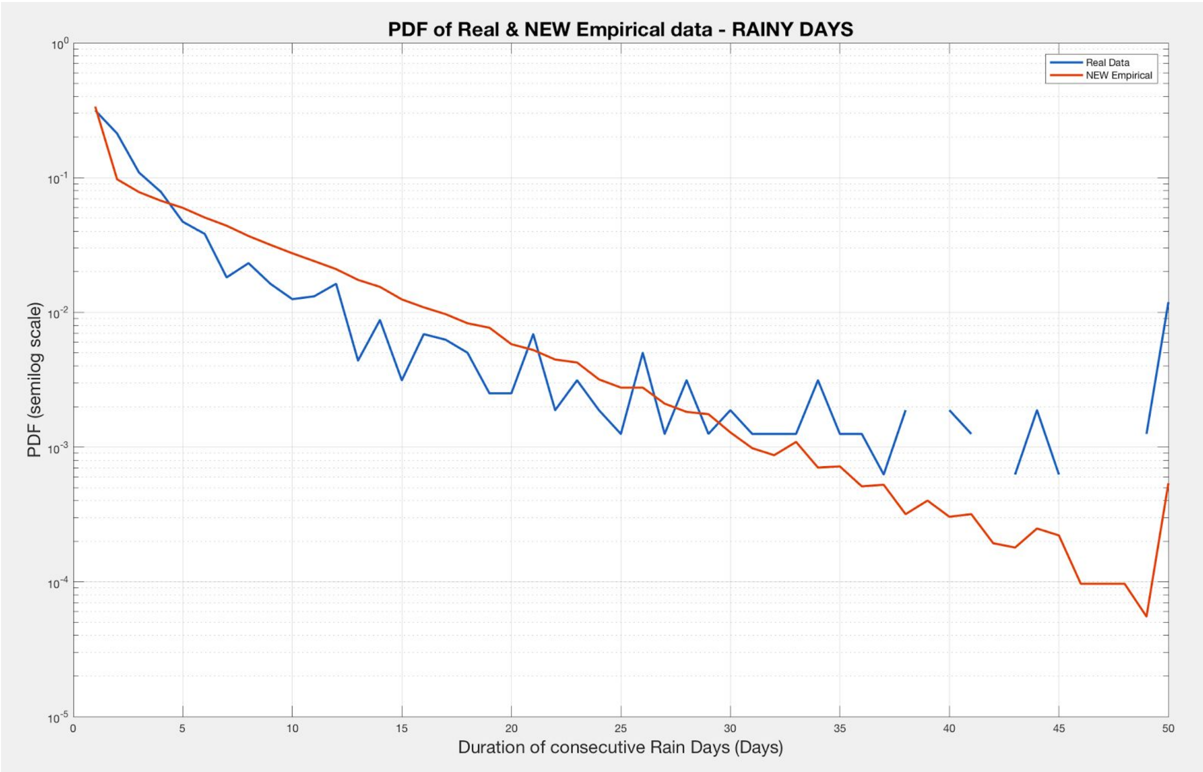
$$A = \begin{array}{c} \\ \\ \\ \\ \end{array}\begin{array}{ccc} E_1 & E_2 & E_3 \\ \text{clear} & \text{cloudy} & \text{very cloudy} \end{array}$$

$$A = \begin{pmatrix} 0,9968 & 0,0032 & 0 \\ 0,0414 & 0.8777 & 0,0810 \\ 0 & 0.1403 & 0,8597 \end{pmatrix} \begin{array}{l} \\ = 1 \\ = 1 \end{array} ; \quad B = \begin{array}{cc} \text{Not rain} & \text{rain} \end{array} \begin{pmatrix} 1 & 0 \\ 0,9686 & 0,0314 \\ 0 & 1 \end{pmatrix}$$

A priori coefficients

$$A = \begin{pmatrix} 0.8 & 0.2 & 0 \\ 0.5 & 0.4 & 0.1 \\ 0 & 0.35 & 0.65 \end{pmatrix} ; \quad B = \begin{array}{c} E_1 \\ E_2 \\ E_3 \end{array} \begin{pmatrix} 1 & 0 \\ 0.8 & 0.2 \\ 0 & 1 \end{pmatrix}$$

- the probability to pass from clear to cloudy got lowered from 0.2 to 0,032. The probability to stay in the state 'clear' augmented to 0.996.
- the probability to remain in cloudy state increased from 0.4 to 0,877. The probability to clear decreased importantly.
- the transition from very cloudy to cloudy decreased from 0.35 to 0.14 and the probability to stay in very cloudy state increased
- Regarding the emission matrix the probability of not rain in a cloudy day increased to 0.968. The others remained almost equal.
- In general the states are not easy to determine as they are not measurables directly. Therefore we are not sure if each state corresponds to a clear, cloudy and very cloudy day that's generally true in HMM models



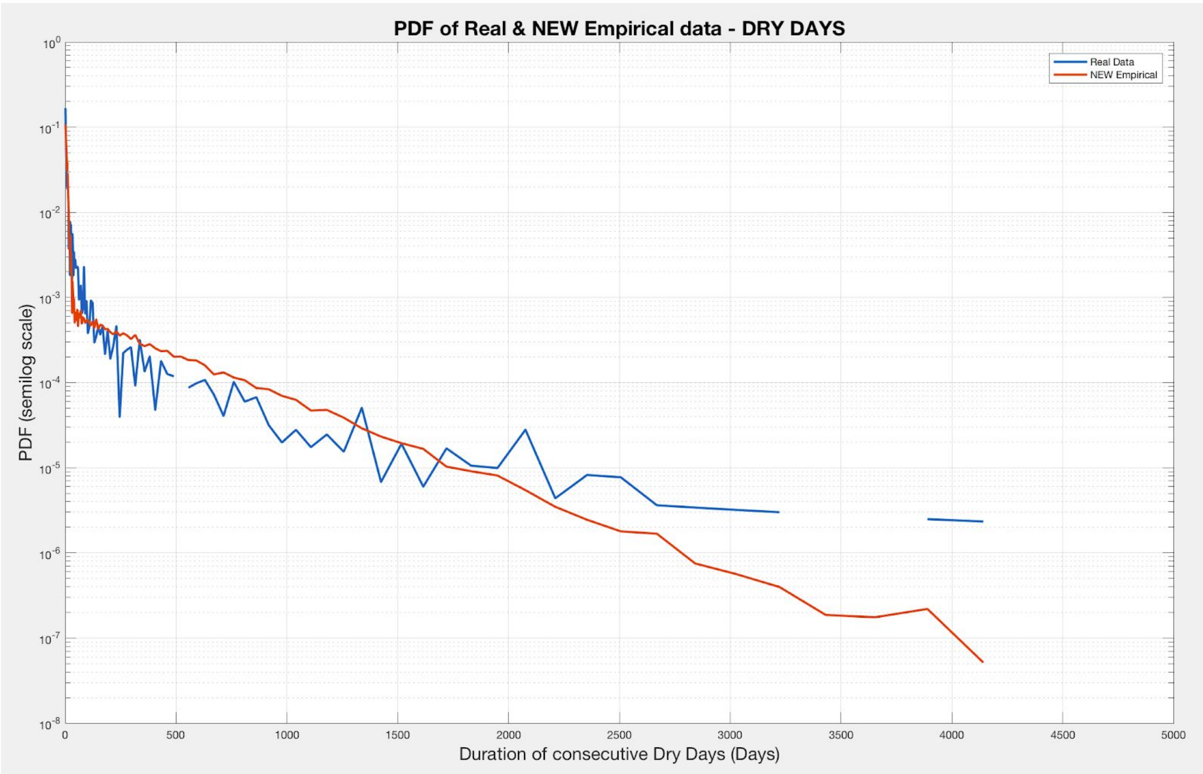


Figure 3. Empirical and real distributions by duration using trained model on rainy and dry days.

• We also remark that emission probabilities of state E_1 and E_2 are fairly similar (only 4% of difference) we propose to reduce the number of states to two. the results of this experiments shows using a priori probabilities an amelioration on the modelization of rainy days. After adjustemens, the model does not fit as well as the three states model, as does not present the 'elbow' at the beginning of the figure that improves the adjustement.

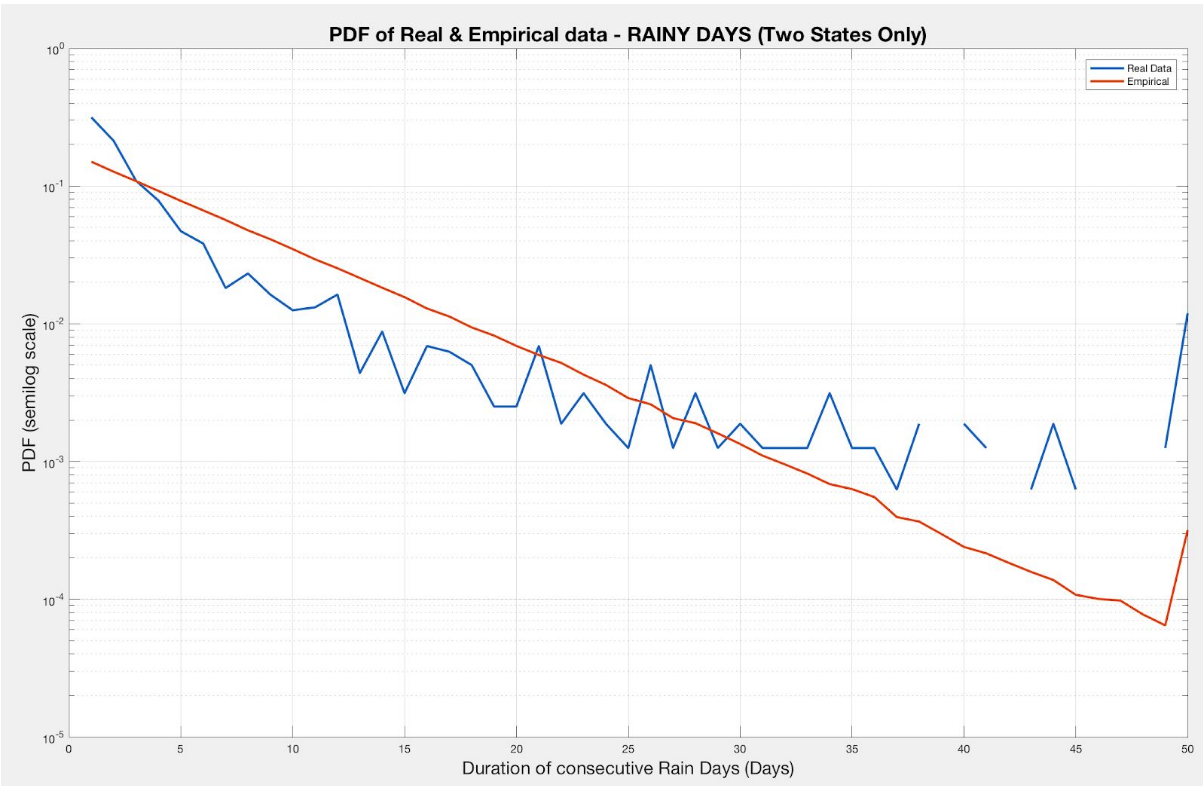


Figure 4. Empirical and real distributions by duration using trained model on rainy days (TWO STATES).

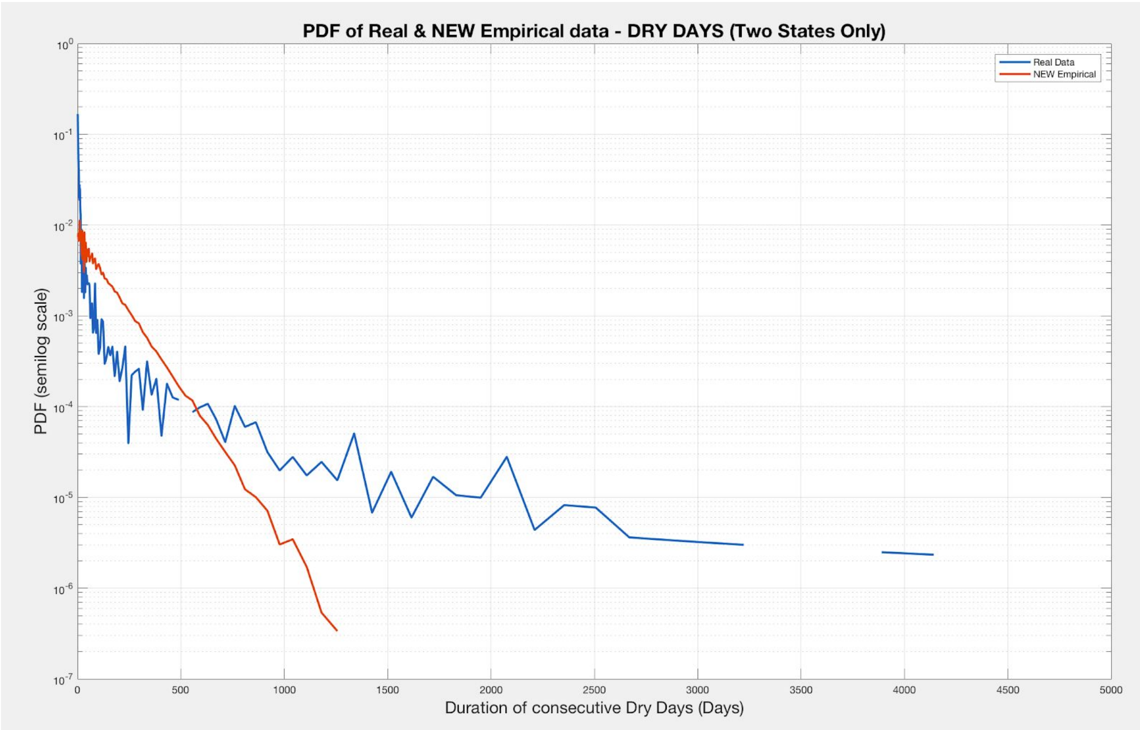
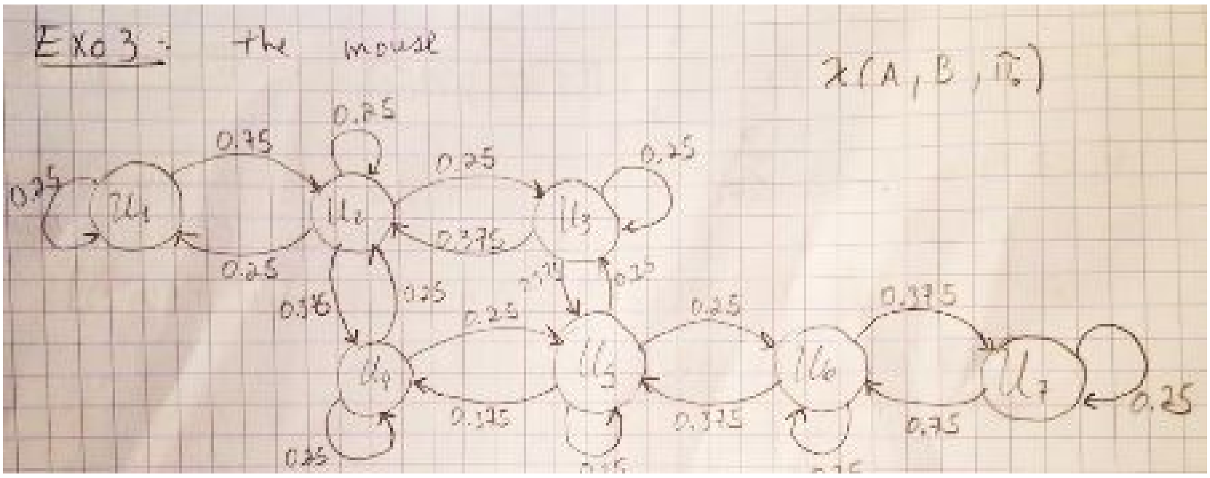


Figure 5. Empirical and real distributions by duration using trained model on dry days (TWO STATES).

2.4 In conclusion a modification of three states using a hidden markov model outperformed the two-states Markov chain. By adding some complexity and taking into account emission probability allows to better modelise the distribution. We found an 'elbow' of non linearity in low durations that allows to obtain a better fit. this is valid both in rainy and dry events. However, the model is far from being perfect, we observe important differences in the tail of the distributions. Also the fact we use 3 states instead of only two (even with fairly similar emission probabilities) helped to improve the results.

Exercise 3.



$$\pi_0 = (1, 0, 0, 0, 0, 0, 0)$$

$$A = \begin{bmatrix} 0.25 & 0.75 & 0 & 0 & 0 & 0 & 0 \\ 0.25 & 0.25 & 0.25 & 0.25 & 0 & 0 & 0 \\ 0 & 0.375 & 0.25 & 0 & 0.375 & 0 & 0 \\ 0 & 0.375 & 0 & 0.25 & 0.375 & 0 & 0 \\ 0 & 0 & 0.25 & 0.25 & 0.25 & 0.25 & 0 \\ 0 & 0 & 0 & 0 & 0.375 & 0.25 & 0.375 \\ 0 & 0 & 0 & 0 & 0 & 0.75 & 0.25 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.7 & 0.1 & 0.1 & 0 & 0 & 0 \\ 0 & 0.15 & 0.7 & 0 & 0.15 & 0 & 0 \\ 0 & 0.15 & 0 & 0.7 & 0.15 & 0 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0.7 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.15 & 0.7 & 0.15 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \end{bmatrix}$$

3.4 After simulating a sequence of 10.000 we obtain a percentage of correct predictions of 69.08%. When increasing to 1.000.000 samples we obtain 70,0414%. that corresponds almost exactly to the detection rate of the detector.

3.5) $P(D \geq 2) = \sum_{i=1}^{7} \left(P(E_i|E_i) \cdot P(E_i) \right)$ $\quad 0.25$

$P(D \geq 2) = 0.25 \sum_{i=1}^{7} P(E_i) = 0.25$

$P(D \geq 2) = 1 - P(D=1) = 1 - 0.75 = 0.25$ $\quad 0.1875$

$P(D \geq 3) = 1 - P(\leq 2) = 1 - [P(D=1) + P(D=2)]$ $\quad 0.75 \quad (0.25)\cdot(0.75)$

$P(D \geq 3) = 0{,}0625$

3.6) After obtaining the histogram of states of the generated sequence we can estimate the probabilities as:

$$\boxed{P(E_i) = \frac{\text{count } E_i}{\text{total}}}$$

We obtain the following results

$P(E_1) = 0.069$; $P(E_2) = 0.207$, $P(E_3) = 0.145$

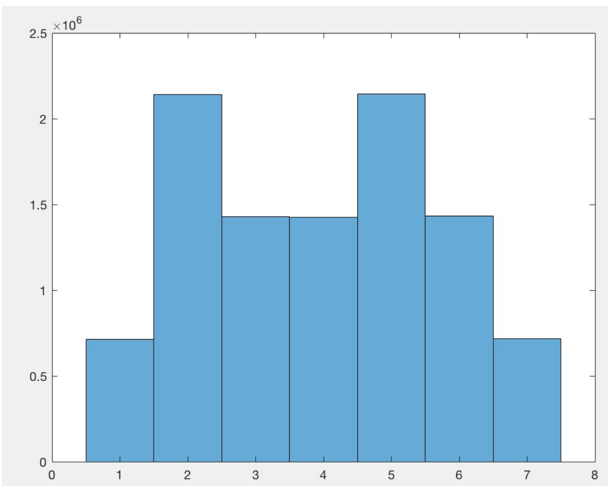
$P(E_4) = 0.135$; $P(E_5) = 0.22$; $P(E_6) = 0.149$

$P(E_7) = 0.079$

✓ the states E_5 and E_2 accounts for 42.7% of the total, they present the highest probability.

✓ States E_3, E_6 and E_4 have approximately 0,14 each.

✓ the less probable states are E_1 and E_7. this behaviour can be explained by the amount and strenght of the transition probabilities to a given state.



(T = 10000000)

3.7)

(a)

pi0 = [1, 0, 0, 0, 0, 0, 0]

Pi0 * A^30:

P_n = {0.0717   0.2149   0.1430   0.1430   0.2141   0.1422   0.0710}

This results is very similar to the results shown in the empirical histogram above.

(b)

$$A\rho^* = \lambda \rho^*$$
$$A\rho^* - \lambda \rho^* = 0$$
$$A\rho^* - \lambda I \rho^* = 0$$
$$\rho^*(A - \lambda I) = 0$$
so... $|A - \lambda I| = 0$ if $\rho^* \neq 0$

solving the above gives us the eigenvalues of A, which allows us to discover the eigenvectors ρ^*

The left eigenvector::

```
>> [v,d,w] = eig(A);
>> w

w =

    0.1768    0.2504    0.4065    0.4065   -0.1768    0.2504         0
   -0.5303   -0.5666   -0.3811    0.3811   -0.5303    0.5666         0
    0.3536    0.1771   -0.2875   -0.2875   -0.3536    0.1771   -0.7071
    0.3536    0.1771   -0.2875   -0.2875   -0.3536    0.1771    0.7071
   -0.5303    0.1660    0.6506   -0.6506   -0.5303   -0.1660         0
    0.3536   -0.6045    0.1684    0.1684   -0.3536   -0.6045         0
   -0.1768    0.4007   -0.2695    0.2695   -0.1768   -0.4007         0

>> colsum = sum(w,1)

colsum =

    0.0000    0.0000    0.0000    0.0000   -2.4749    0.0000    0.0000
```

The eigenvalues:

```
d =

   -0.5000         0         0         0         0         0         0
         0   -0.3158         0         0         0         0         0
         0         0    0.0157         0         0         0         0
         0         0         0    0.4843         0         0         0
         0         0         0         0    1.0000         0         0
         0         0         0         0         0    0.8158         0
         0         0         0         0         0         0    0.2500
```

The eigenvector associated with the eigenvalue=1 :

```
 -0.1768
 -0.5303
 -0.3536
 -0.3536
 -0.5303
 -0.3536
 -0.1768
```

When normalizing by the sum of the vector we obtain p*= [0.0714, 0.2142, 0.1428, 0.2142, 0.1428, 0.0714]. The coefficients follow the pattern of the empirical histogram plotted in 3.6, and the theoretical state probabilities calculated in 3.7.

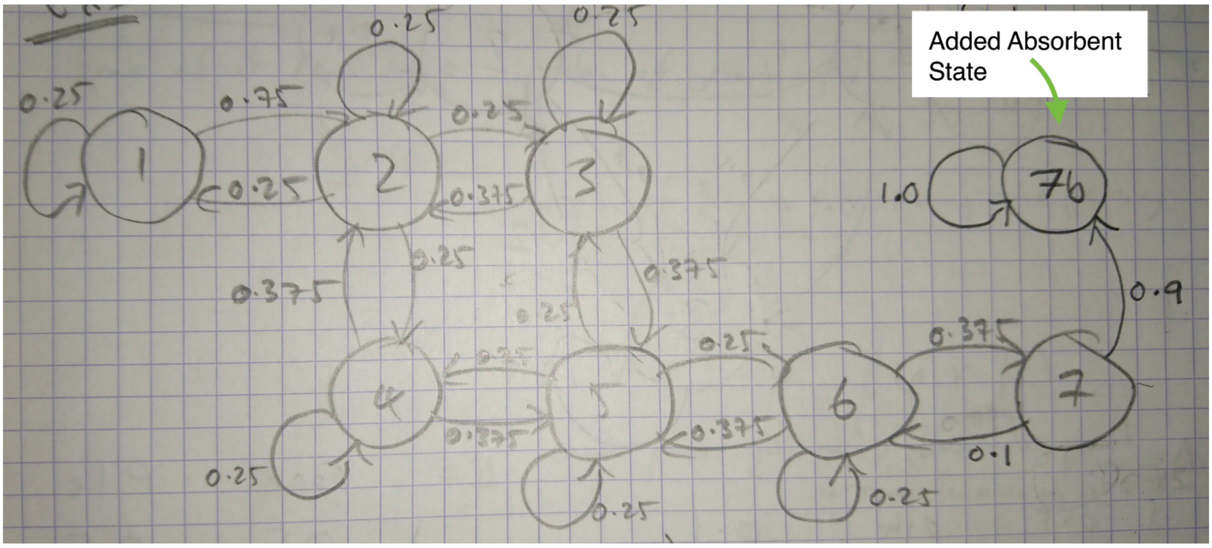
(c)

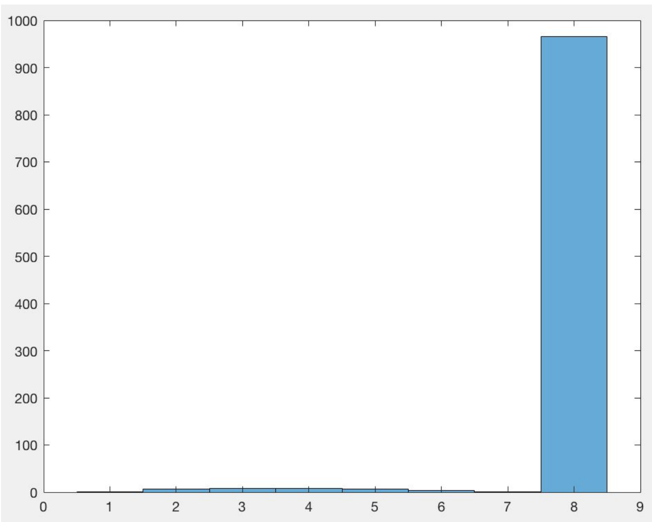
p* does not depend on $\pi 0$, as N is large (30). p* could depend on $\pi 0$ if N is small, e.g. N=3.

3.8)

The estimated sequence of states is 74.8% the same as the real sequence of states. This result is expected as Viterbi's algorithm provides one approximation of the sequence with a highly reduced computational complexity.

3.9)






```
T = 1000 ;
% emission matrix
B = [0.7, 0.3, 0, 0, 0, 0, 0, 0;
     0.1, 0.7, 0.1, 0.1, 0, 0, 0, 0;
     0, 0.15, 0.7, 0, 0.15, 0, 0, 0;
     0, 0.15, 0, 0.7, 0.15, 0, 0, 0;
     0, 0, 0.1, 0.1, 0.7, 0.1, 0, 0;
     0, 0, 0, 0, 0.15, 0.7, 0.15, 0;
     0, 0, 0, 0, 0, 0.15, 0.7, 0.15;
     0, 0, 0, 0, 0, 0, 0.3, 0.7];
% transition matrix
A = [0.25, 0.75, 0, 0, 0, 0, 0, 0;
     0.25, 0.25, 0.25, 0.25, 0, 0, 0, 0;
     0, 0.375, 0.25, 0, 0.375, 0, 0, 0;
     0, 0.375, 0, 0.25, 0.375, 0, 0, 0;
     0, 0, 0.25, 0.25, 0.25, 0.25, 0, 0;
     0, 0, 0, 0, 0.375, 0.25, 0.375, 0;
     0, 0, 0, 0, 0, 0.1, 0, 0.9;
     0, 0, 0, 0, 0, 0, 0, 1.0];

pi0 = [1, 0, 0, 0, 0, 0, 0, 0];
```

Percentages: 0.1% 0.7% 0.8% 0.8% 0.7% 0.3% 0.1% 96.5%

The addition of an absorbent state captures the vast majority of the probability distribution. This is because only one transition to this state is needed before all subsequent states are guaranteed to be this one.

Exercise 4.

4.1)

Transition matrix is the probability of moving from one coin to another.
Observable values are the results of the coin flip: the sequence of heads or tails.

4.2)

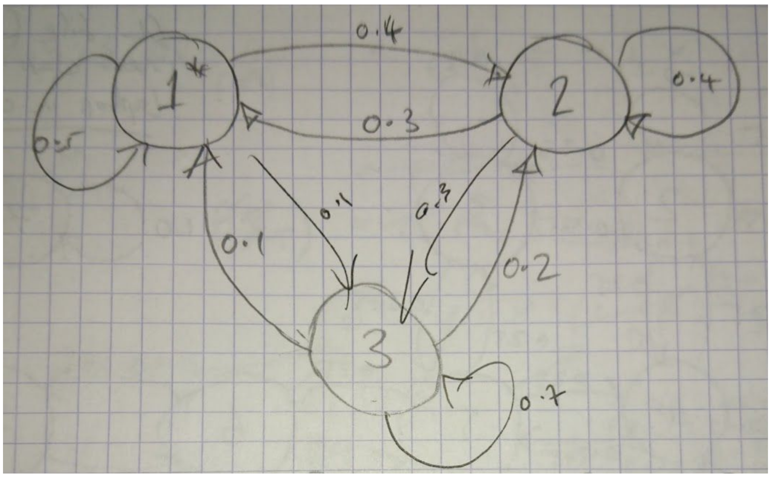
A is the transition matrix, B is the emission probabilities:

A =	1	2	3
1	0.5	0.4	0.1
2	0.3	0.4	0.3
3	0.1	0.2	0.7

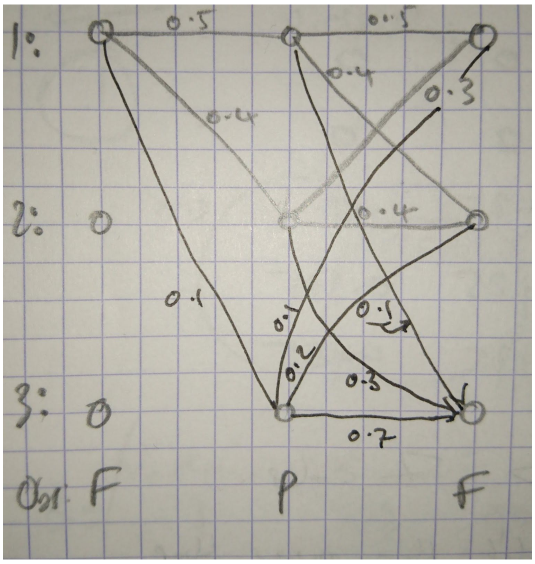
B =	Face	Pile
1	0.5	0.5
2	0.75	0.25
3	0.25	0.75

4.3)

The state transition graph:



The trellis:



4.4)
Sequence = FPF

Pièce 1 : P(f) = 0.5
Pièce 2 : P(f)= 0.75
Pièce 3 : P(f) = 0.25

All possible state paths, with conditional probabilities for sequence FPF:
1,1,1: (1.0*0.5 * 0.5*0.5 * 0.5*0.5**) +**
1,1,2: (1.0*0.5 * 0.5*0.5 * 0.4*0.75**) +**
1,1,3: (1.0*0.5 * 0.5*0.5 * 0.1*0.25**) +**
1,2,1: (1.0*0.5 * 0.4*0.25 * 0.3*0.5**) +**
1,2,2: (1.0*0.5 * 0.4*0.25 * 0.4*0.75**) +**
1,2,3: (1.0*0.5 * 0.4*0.25 * 0.3*0.25**) +**
1,3,1: (1.0*0.5 * 0.1*0.75 * 0.1*0.5**) +**
1,3,2: (1.0*0.5 * 0.1*0.75 * 0.2*0.75**) +**
1,3,3: (1.0*0.5 * 0.1*0.75 * 0.7*0.25**)**
= 0.1121875
= 11.21875% (45 multiplications + 8 additions = 53 calculations required)

4.5)

Using forward method:

$\alpha_n(j) = P(O_1, O_2, O_3, \dots O_n, x_n = E_j/\lambda)$

$\alpha_1(j) = \pi_j b_j(O_1)$ for $j = 1, \dots, N$ # of states

$\alpha_{n+1}(j) = b_j(O_{n+1}) \sum_{i=1}^{N} \alpha_n(i) a_{ij}$ for $n = 1, \dots, T-1$ et $j = 1, \dots, N$

t1
$\alpha_1(1) = \pi_1 \cdot b_{(1)}(f) \rightarrow \alpha_1(1) = (1)\cdot(0.5) = 0.5$
$\alpha_1(2) = \pi_2 \cdot b_{(2)}(f)$ 0
$\alpha_1(3) = \cancel{\pi_3} \cdot b_{(3)}(f)$ 0

$n=2$, $j=1$: $\alpha_2(1) = b_1(P)\cdot\left[\alpha_1(1)a_{11} + \alpha_1(2)a_{21} + \alpha_1(3)a_{31}\right] = 0.125$

$j=2$: $\alpha_2(2) = b_2(P)\cdot\left[\alpha_1(1)a_{12} + \alpha_1(2)a_{22} + \alpha_1(3)a_{32}\right] = 0.25\cdot0.2 = 0.05$

$j=3$: $\alpha_2(3) = b_3(P)\left[\alpha_1(1)a_{13} + \alpha_1(2)a_{23} + \alpha_1(3)a_{33}\right]$

$\alpha_2(3) = 0.75\cdot(0.05) = 0.0375$

$n=3$, $j=1$: $\alpha_3(1) = b_1(f)\left[\alpha_2(1)a_{11} + \alpha_2(2)a_{21} + \alpha_2(3)a_{31}\right] = 0.0406$

$j=2$: $\alpha_3(2) = b_2(f)\left[\alpha_2(1)a_{12} + \alpha_2(2)a_{22} + \alpha_2(3)a_{32}\right] = 0,058$

$j=3$: $\alpha_3(3) = b_3(f)\left[\alpha_2(1)a_{13} + \alpha_2(2)a_{23} + \alpha_2(3)a_{33}\right] = 0,0134$

$P(O_T|\lambda) = \sum_{j=1}^{N} \alpha_T(j) = \alpha_3(1) + \alpha_3(2) + \alpha_3(3) = \boxed{0.112}$

fpf

This method required only 41 calculations. This is only marginally less than performing an exhaustive calculation in 4.4, but this only a very short sequence. For a much longer sequence, the reduction in number of calculations would be much more significant, as the number of calculations in the exhaustive method would increase exponentially.
Result of forward algorithm = **0.112**

4.6)
Hmmdecode result for forward+backward algorithm: **0.1293**
This is similar to the previous results, but not exactly the same, due to differences in the algorithm implementations. Specifically, more information is used to calculate each state probabilities (alpha) with the forward+backward algorithm, as all preceding and subsequent states are used in every calculation, as opposed to the forward algorithm which only uses the preceding states.

The forward–backward algorithm has time complexity $O(N^2T)$, which is the same as the complexity of the forward algorithm.

4.7)
P(f,f,f) = 0.1635
P(p,p,p) = 0.1033

The difference can be explained by the the following facts:

- The emission probability for Face are higher in state 2, and lower in state 3
- The transition probabilities from both state 1 and state 2 are higher to state 2 than to state 3. This means it is more likely to reach state 2 from two of the three possible states, and therefore state 2 occurs more often.
- As state 2 occurs more often, and emits Face more often than Pile, Face is emitted more often than Pile, and so the probability of FFF is higher than for PPP.

4.8)

✓ Compute Viterbi of the sequence "F P F"

$\delta_n(j)$ and prove $\delta_n(j) = $

$$\begin{matrix} 0.5 & 0.125 & 0.03125 \\ 0 & 0.05 & 0.0375 \\ 0 & 0.0375 & 0.0065 \end{matrix}$$

Viterbi : estimate the sequence of states given a sequence of observations

$$X_{T\,optimal} = \underset{X_T = x_1, x_2, \ldots, x_T}{\arg\max} P(X_T, O_T / \lambda)$$

$\boxed{n=1}$ $\delta_1(j) = \pi_j b_j(O_1)$ and $\psi_1(j) = 0 \quad 1 \leq j \leq N$

$\delta_1(1) = \pi_1 b_1(F) = (1)(0.5) = 0.5 \quad ; \quad \psi_1(1) = 0$

$\delta_1(2) = \pi_2 b_2(F) = 0 \quad ; \quad \psi_1(2) = 0$

$\delta_1(3) = \pi_3 b_3(F) = 0 \quad ; \quad \psi_1(3) = 0$

$\boxed{n=2}$ $\delta_2(1) = \max\limits_{1 \leq i \leq 3} [\delta_1(i) a_{ij}] \times b_j(O_n)$

$(0.5)(0.5) \cdot (0.5) \quad 0 \quad 0$

$\delta_2(1) = \max\limits_{1 \leq i \leq 3} [\delta_1(1) a_{11}, \cancel{\delta_1(2) a_{21}}, \cancel{\delta_1(3) a_{31}}] \cdot b_1(P)$

$\delta_2(1) = 0.125 \quad ; \quad \psi_2(1) = 1$

$(0.5)(0.4)(0.25) \quad 0 \quad 0$

$\delta_2(2) = \max\limits_{1 \leq i \leq 3} [\delta_1(1) a_{12}, \delta_1(2) a_{22}, \cancel{\delta_1(3) a_{32}}] \cdot b_2(P)$

$\psi_2(2) = 1$

$\delta_2(2) = 0.05$

$(0.5)(0.1)(0.75) \quad 0 \quad 0$

$\delta_2(3) = \max\limits_{1 \leq i \leq 3} [\delta_1(1) a_{13}, \cancel{\delta_1(2) a_{23}}, \cancel{\delta_1(3) a_{33}}] \, b_3(P)$

$\delta_2(3) = 0.0375 \quad ; \quad \psi_2(3) = 1$

$$\max \; \overset{0.0625}{[(0.125)(0.5)} \quad \overset{0.015}{(0.05)(0.3)} \quad \overset{0.00375}{(0.0375)(0.1)]} \; \overset{\cdot 0.5}{(0.5)}$$

$\boxed{n=3}$ $j=1$

$$\delta_3(1) = \max_{1 \le i \le 3} \left[\delta_2(1) a_{1,1}, \; \delta_2(2) a_{21}, \; \delta_2(3) a_{31} \right] b_1(F)$$

$$\delta_3(1) = 0{,}03125 \; ; \; \psi_3(1) = 1$$

$$\max \; \overset{0.05}{[(0.125)(0.4)} \quad \overset{0.02}{(0.05)(0.4)} \quad \overset{0.0075}{(0.0375)(0.2)]} \; 0.75$$

$$\delta_3(2) = \max_{1 \le i \le 3} \left[\delta_2(1) a_{12}, \; \delta_2(2) a_{22}, \; \delta_2(3) a_{32} \right] b_2(F)$$

$$\delta_3(2) = 0{,}0375 \rightarrow \psi_3(2) = 1$$

$$[0.0125 \quad 0.015 \quad 0.02625] \; (0.25)$$

$$\max \left[(0.125)(0.1), \; (0.05)(0.3) \; (0.0375)(0.7) \right] (0.25)$$

$$\delta_3(3) = \max_{1 \le i \le 3} \left[\delta_2(1) a_{13}, \; \delta_2(2) a_{23}, \; \delta_2(3) a_{33} \right] b_3(F)$$

$$\delta_3(3) = 0{,}0065625 \; ; \; \psi_3(3) = 3$$

Final state

$$x^*_3 = \arg\max_{1 \le i \le N} \left[\delta_3(1), \delta_3(2), \delta_3(3) \right] = 2$$

$$P^*(x) = \max_{1 \le i \le N} \left[\delta_3(1), \delta_3(2), \delta_3(3) \right] = 0.0375$$

Previous states:

T_2, T_1

$$x^*_2 = \psi_3(x^*_3) = \psi_3(2) = 1$$

$$x^*_1 = \psi_2(x^*_2) = \psi_2(1) = 1$$

Optimal sequence is:

$$\boxed{X^*_T = 1, 1, 2.}$$

4.9)

Viterbi calculates the most probable state path for a sequence of observations. Given the sequence FPF, the Matlab calculated the most probable state path as: **[1, 1, 2]**

4.10)

Original parameters:

```
% col1 = heads, col2 = tails
B = [0.5, 0.5;
     0.75, 0.25;
     0.25, 0.75];
% transition matrix
A = [0.5, 0.4, 0.1;
     0.3, 0.4, 0.3;
     0.1, 0.2, 0.7];
```

T=1000

Guess parameters for Baum-Welch - **no a priori knowledge:**

```
% col1 = heads, col2 = tails
B_guess = [0.5, 0.5;
           0.5, 0.5;
           0.5, 0.5];
% transition matrix
A_guess = [0.33, 0.33, 0.34;
           0.33, 0.33, 0.34;
           0.33, 0.33, 0.34];
```

A_est:

0.3300	0.3300	0.3400
0.3300	0.3300	0.3400
0.3300	0.3300	0.3400

B_est:

0.4765	0.5235
0.4765	0.5235
0.4765	0.5235

- With no a priori knowledge, the estimated parameters based solely on the sequence are very poor, and do not resemble the original parameters much at all.

Guess parameters for Baum-Welch - **some priori knowledge (better guesses):**

```
% emission matrix
% col1 = heads, col2 = tails
B_guess = [0.4, 0.6;
           0.7, 0.3;
           0.15, 0.85];
% transition matrix
A_guess = [0.45, 0.35, 0.2;
           0.35, 0.45, 0.2;
           0.2, 0.15, 0.65];
```

A_est:

0.4919	0.3298	0.1783
0.3647	0.4836	0.1517
0.2412	0.2951	0.4637

B_est:

0.4359	0.5641
0.9914	0.0086
0.1488	0.8512

- With better guesses, the estimated values of A and B are much closer to the originals.

T=10000
Using the better guesses, the algorithm was run using a larger value of T. The resulting values more closely approximated the original values of A and B.

A_est:

0.4423	0.3595	0.1982
0.3429	0.4763	0.1807
0.2519	0.2403	0.5078

B_est:

0.4627	0.5373
0.7569	0.2431
0.1174	0.8826

T=100000
Again using the better guesses, the algorithm was run using an even larger value of T. The resulting values even more closely approximated the original values of A and B.

A_est:

0.4443	0.4259	0.1298
0.4155	0.4349	0.1496
0.1919	0.1775	0.6306

B_est:

0.4551	0.5449
0.7030	0.2970
0.2134	0.7866

Exercise 5.

Exo 5 Estimate yearly average temperature in 6 years, that happened 5000 years ago!
We wanna know which of these years were cold or warm (F, C). This is possible to be estimated by measuring the thickness of tree rings.

P_1 = narrow P_2 = medium , P_3 = wide

States $\begin{cases} E_1 = \text{chaude} \\ E_2 = \text{froid} \end{cases}$

chaude

$\begin{cases} b_1(\text{low}) = 0.1 \\ b_1(\text{medium}) = 0.4 \\ b_1(\text{large}) = 0.5 \end{cases}$

$b_2(\text{low}) = 0.7$
$b_2(\text{medium}) = 0.2$
$b_2(\text{large}) = 0.1$

$$A = \begin{array}{c} \\ E_1 \\ E_2 \end{array} \begin{array}{c} \begin{array}{cc} E_1 & E_2 \end{array} \\ \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array} \qquad B = \begin{array}{c} \\ b_1 \\ b_2 \end{array} \begin{array}{c} \begin{array}{ccc} \text{low} & \text{medium} & \text{large} \end{array} \\ \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \end{array}$$

π_0 (0.5 , 0.5)

Knowing the sequence of observations:
[low, medium, low, large, medium, low]
Deduce the optimal sequence of states.

Using Viterbi's algorithm : $j = 1, 2$, $n = 6$

$n=1$

$\delta_1(1) = \pi_1 b_1(\text{low}) = (0.5)(0.1) = 0.05$ $\psi_1(1) = 0$

$\delta_1(2) = \pi_2 b_2(\text{low}) = (0.5)(0.7) = 0.35$ $\psi_1(2) = 0$

$n=2$

$$\delta_2(1) = \max_{1 \le i \le 2} \Big[\overset{(0.05)(0.7)}{\delta_1(1) a_{11}} , \overset{(0.35)(0.4)}{\delta_1(2) a_{21}} \Big] \overset{(0.4)}{b_1(\text{medium})}$$

$\boxed{\delta_2(1) = 0{,}056 \; ; \; \psi_2(1) = 2}$

$$\delta_2(2) = \max_{1 \le i \le 2} \Big[\overset{(0.05)(0.3)}{\delta_1(1) a_{12}} , \overset{(0.35)(0.6)}{\delta_1(2) a_{22}} \Big] \overset{(0.2)}{b_2(\text{medium})}$$

$\boxed{\delta_2(2) = 0{,}042 \; ; \; \psi_2(2) = 2}$

0.03920 ✓ $\quad 0.0168$

$(0.056)(0.7) \quad (0.042)(0.4) \quad (0.1)$

$n=3$

$$\delta_3(1) = \max_{1\le i\le 2} [\delta_2(1)a_{11}, \delta_2(2)a_{21}]\, b_1(low)$$

$$\boxed{\delta_3(1) = 0.00392 \; ; \; \psi_3(1) = 1}$$

$(0.056)(0.3) \quad (0.042)(0.6) \quad (0.7)$

$$\delta_3(2) = \max_{1\le i\le 2} [\delta_2(1)a_{12}, \delta_2(2)a_{22}]\, b_2(low)$$

$$\boxed{\delta_3(2) = 0.01764 \; ; \; \psi_3(2) = 2}$$

$(0.00392)(0.7) \quad (0.01764)(0.4) \quad (0.5)$

$n=4$

$$\delta_4(1) = \max_{1\le i\le 2} [\delta_3(1)a_{11}, \delta_3(2)a_{21}]\, b_1(large)$$

$$\boxed{\delta_4(1) = 0.003528 \; ; \; \psi_4(1) = 2}$$

$0.001176 \quad 0.010584$

$(0.00392)(0.3) \quad (0.01764)(0.6) \quad (0.1)$

$$\delta_4(2) = \max_{1\le i\le 2} [\delta_3(1)a_{12}, \delta_3(2)a_{22}]\, b_2(large)$$

$$\boxed{\delta_4(2) = 0.000176 \; ; \; \psi_4(2) = 1}$$

$(0.003528)(0.7) \quad (0.000176)(0.4) \quad (0.4)$

$n=5$

$$\delta_5(1) = \max_{1\le i\le 2} [\delta_4(1)a_{11}, \delta_4(2)a_{21}]\, b_1(medium)$$

$$\boxed{\delta_5(1) = 0{,}00098784 \; , \; \psi_5(1) = 1}$$

$(0.003528)(0.3) \quad (0.000176)(0.6) \quad (0.2)$

$$\delta_5(2) = \max_{1\le i\le 2} [\delta_4(1)a_{12}, \delta_4(2)a_{22}]\, b_2(medium)$$

$$\boxed{\delta_5(2) = 0{,}00021168 \; , \; \psi_5(2) = 2}$$

$0{,}00098784(0.7) \quad (0.00021168)(0.4) \quad (0.1)$

$n=6$

$$\delta_6(1) = \max_{1\le i\le 2} [\delta_5(1)a_{11}, \delta_5(2)a_{21}]\, b_1(low)$$

$$\boxed{\delta_6(1) = 0{,}000069149 \; ; \; \psi_6(1) = 1}$$

$0{,}000296352$

$0{,}00098784(0.3) \quad (0{,}00021168)(0.6) \quad (0.7)$

$$\delta_6(2) = \max_{1\le i\le 2} [\delta_5(1)a_{12}, \delta_5(2)a_{22}]\, b_2(low)$$

$$\boxed{\delta_6(2) = 0{,}0002074464 \; ; \; \psi_6(2) = 1}$$

the optimal sequence is given by

$x^*_6 = \arg\max[\delta_6(1), \delta_6(2)] = 2$

$P x^* = 0{,}00006 4149$

$x^*_5 = \psi_6(x^*_6) = \psi_6(2) = 1$

$x^*_4 = \psi_5(x^*_5) = \psi_5(1) = 1$

$x^*_3 = \psi_4(x^*_4) = \psi_4(1) = 2$

$x^*_2 = \psi_3(x^*_3) = \psi_3(2) = 2$

$x^*_1 = \psi_2(x^*_2) = \psi_2(1) = 2$

$x^* [F, F, F, C, C, F]$

It would be possible to have an estimation of the sequence of states by using the a priori probabilities we have.

Exercise 6.

Exo 6

(a) number of parameters to estimate.

$$A = \begin{array}{c|c} & E_1 \; \cdots \; E_N \\ \hline \begin{matrix} E_1 \\ \vdots \\ E_N \end{matrix} & \left(\begin{matrix} & & \\ & & \\ & & \end{matrix} \right)_{N \times N} \end{array}$$

for each $E_1, \ldots, E_j$ where $E_j = \{\mu_1, \ldots, \mu_K\}$

$1 \leq j \leq N$ and $\sigma_j = \dim(K \times K)$

Finally:

$$\boxed{\#\,param = K^2 \cdot N + K(N) + N^2}$$

4 Fold CV

(c)

N \ K	2	4	6
2	15.38%	20.00%	4.62%
4	9.23%	9.23%	6.15%
6	12.31%	0.00%	6.15%
8			

$N=3$; $K=3$ = 6.15%

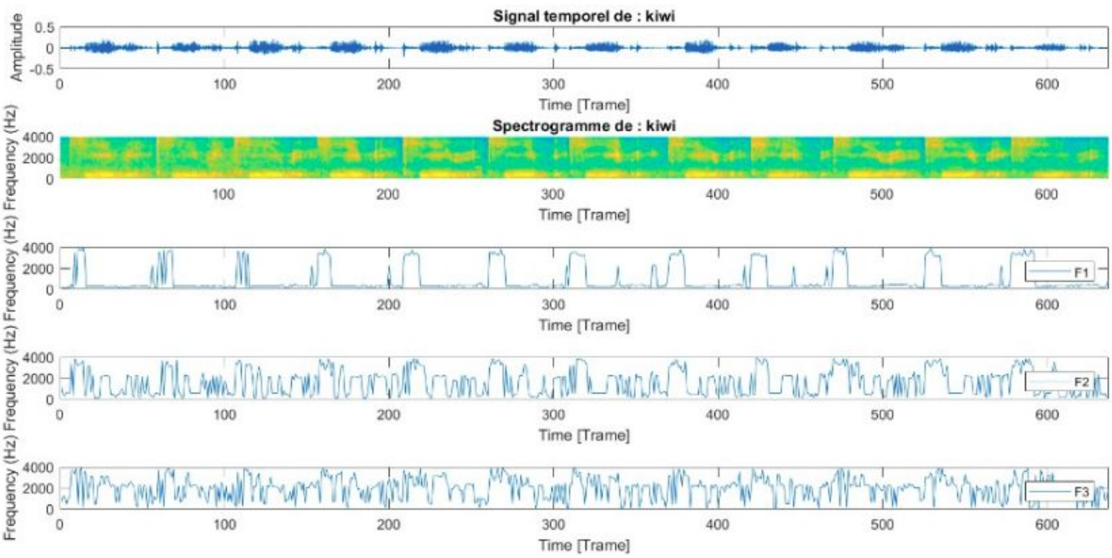
(b) for $N=K=3$

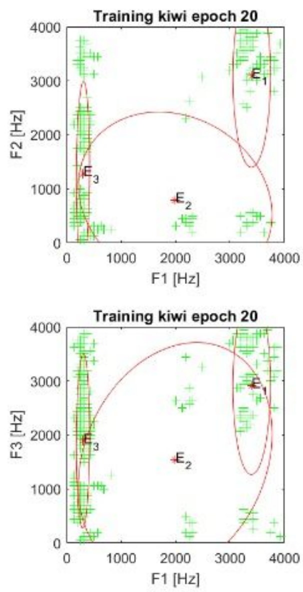
when comparing the states of the GMM markov chain against the peaks of the spectrogram of the word "Kiwi".
the beginning of the word "K" corresponds to E_1 of the GMM and is composed by high frequency components.

(b) for $N=K=3$

when comparing the states of the GMM markov chain agains the peaks of the spectrogram of the word "Kiwi". the beginning of the word "K" corresponds to E_1 of the GMM and is composed by high frequency components.

E_2 correspond to the end of the word. E_3 is the transition from "i" to "w".





(c) When using a K low, it is more difficult to modelize large words with complex sounds, as there is a relationship between the states of the model and parts of the word.