STATISTICAL COMPUTATIONAL METHODS

Seminar Nr. 4, Markov Chains, Applications and Simulations

1. (Computer mode) A computer system can operate in two different modes. Every hour, it remains in the same mode or switches to a different mode according to the transition probability matrix

$$P = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}.$$

a) If the system is in Mode I at 5:30 pm, what is the probability that it will be in Mode I at 8:30 pm on the same day?
b) In the long run, in which mode is the system more likely to operate?

Solution:

a) This is a stochastic process, with two states, 1, "the system operates in Mode I" and 2, "the system is in Mode II". So, it is *discrete-state*. The time is measured "every hour", so it is also *discrete-time*. In order to predict the future, we only need to know the present, i.e. how the computer changes modes from one hour to the next, hence, it is also *Markov* and, thus, a *Markov chain*. Also, since the probabilities of switching from one mode to another are the same at any time (hour), it is a *homogeneous* Markov chain.
The initial time is 5:30 pm. Now, 8:30 pm is 3 hours after 5:30 pm, so we want to compute

$$p_{11}^{(3)} = P(X_3 = 1 \mid X_0 = 1),$$

for which we need the 3-step transition probability matrix, $P^{(3)} = P^3$. In Matlab,

```
>> P = [0.4 0.6; 0.6, 0.4]
P =
    0.4000    0.6000
    0.6000    0.4000
>> P^3
ans =
    0.4960    0.5040
    0.5040    0.4960
```

$$P^3 = \begin{bmatrix} p_{11}^{(3)} & p_{12}^{(3)} \\ p_{21}^{(3)} & p_{22}^{(3)} \end{bmatrix} = \begin{bmatrix} 0.496 & 0.504 \\ 0.504 & 0.496 \end{bmatrix}.$$

So that probability is 0.496.

b) For the "long run", we need the steady-state distribution. Notice that P (and P^3) has all nonzero entries, so the Markov chain is *regular*, which means a steady-state distribution does exist. We find it by solving the system $\pi P = \pi, \sum_{x=1}^{2} \pi_x = 1$,

$$\begin{aligned} [\pi_1 \;\; \pi_2] \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix} &= [\pi_1 \;\; \pi_2] \\ \pi_1 + \pi_2 &= 1, \end{aligned}$$

i.e,

$$\begin{cases} 0.4\pi_1 + 0.6\pi_2 = \pi_1 \\ 0.6\pi_1 + 0.4\pi_2 = \pi_2 \\ \pi_1 + \pi_2 = 1 \end{cases},$$

with solution $\pi_1 = \pi_2 = 0.5$. So, in the long run, the pdf of the forecast will be

$$\lim_{h\to\infty} P_h = \pi = [\pi_1 \;\; \pi_2] = [0.5 \;\; 0.5].$$

That means that, in the long run, the system is just as likely to operate in Mode I, as it is to operate in Mode II. (Notice that even after only 3 steps, the transition probabilities were already very close to 0.5.)

2. (Genetics) An offspring of a black dog is black with probability 0.6 and brown with probability 0.4. An offspring of a brown dog is black with probability 0.2 and brown with probability 0.8. Rex is a brown dog. What is the probability that his grandchild is black?

Solution:

This is a stochastic process with 2 states. Let "black" be state 1 and "brown" be state 2. The time is measured from one generation to the next ("offspring"), so discreetly. It has the Markov property (only information about the previous generation is needed), so it is a Markov chain. Again, the transition probabilities are stationary (the same at any time), thus, so is the Markov chain (stationary or homogeneous).

The transition probability matrix is

$$P = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}.$$

Rex is a brown dog (state 2), so the initial situation is

$$P_0 = [0 \;\; 1].$$

His grandchild is two generations away, so we need the first component of P_2,

$$P_2 \;\; = \;\; P_0 \cdot P^2 \;\; = \;\; [0 \;\; 1] \begin{bmatrix} p_{11}^{(2)} & p_{12}^{(2)} \\ p_{21}^{(2)} & p_{22}^{(2)} \end{bmatrix} \;\; = \;\; [p_{21}^{(2)} \;\; p_{22}^{(2)}],$$

i.e. $p_{21}^{(2)}$. We could compute the entire matrix P^2, or just that one entry. The entry $p_{21}^{(2)}$ in P^2 is obtained from the second row and first column of P:

$$p_{21}^{(2)} \;\; = \;\; [p_{21} \, p_{22}] \begin{bmatrix} p_{11} \\ p_{21} \end{bmatrix} \;\; = \;\; p_{21} \cdot p_{11} + p_{22} \cdot p_{21} \;\; = \;\; (0.2)(0.6) + (0.8)(0.2) \;\; = \;\; 0.28.$$

3. (Traffic lights) Every day, student A takes the same road from his home to the university. There are 4 street lights along his way, and he noticed the following pattern: if he sees a green light at an intersection, then 60% of the time the next light is also green (otherwise, red), and if he sees a red light, then 70% of the time the next light is also red (otherwise, green).
a) If the first light is green, what is the probability that the third light is red?
b) Student B has *many* street lights between his home and the university, but he notices the same pattern. If the first street light on his road is green, what is the probability that the last light is red?

Solution:

This is a Markov chain with 2 states, "green light" state 1 and "red light" state 2.
The transition probability matrix is

$$P = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}.$$

a) The initial situation (first light) is green, so

$$P_0 = [1 \;\; 0].$$

Now, we want the second component of P_2,

$$P_2 \;=\; P_0 \cdot P^2 \;=\; [1\ \ 0] \begin{bmatrix} p_{11}^{(2)} & p_{12}^{(2)} \\ p_{21}^{(2)} & p_{22}^{(2)} \end{bmatrix} \;=\; [p_{11}^{(2)}\ \ p_{12}^{(2)}]],$$

i.e. $p_{12}^{(2)}$. Again, we can compute that directly (without finding the entire matrix P^2), multiplying the first row and second column of P:

$$p_{12}^{(2)} \;=\; [p_{11}\, p_{12}] \begin{bmatrix} p_{12} \\ p_{22} \end{bmatrix} \;=\; p_{11} \cdot p_{12} + p_{12} \cdot p_{22} \;=\; (0.6)(0.4) + (0.4)(0.7) \;=\; 0.52.$$

b) For "many streets" away, we use the steady-state distribution. Since P has all nonzero entries, this Markov chain is regular, so a steady-state distribution exists. We set up the system $\pi P = \pi, \sum\limits_{x=1}^{2} \pi_x = 1$, i.e,

$$\begin{cases} 0.6\pi_1 & + & 0.3\pi_2 & = & \pi_1 \\ 0.4\pi_1 & + & 0.7\pi_2 & = & \pi_2 \\ \pi_1 & + & \pi_2 & = & 1 \end{cases},$$

with solution $\pi_1 = 3/7, \pi_2 = 4/7 \approx 0.5714$. So, after "many streets", the probability of a red light (i.e. that the process is in state 2) is

$$\pi_2 \;=\; 4/7 \;\approx\; 0.5714.$$

4. (Shared device) A computer is shared by 2 users who send tasks to it remotely and work independently. At any minute, any connected user may disconnect with probability 0.5, and any disconnected user may connect with a new task with probability 0.2. Let $X(t)$ be the number of concurrent users at time t.

a) Find the transition probability matrix.

b) Suppose there are 2 users connected at 10:00 a.m. What is the probability that there will be 1 user connected at 10.02?

c) How many connections can be expected by noon?

Solution:

The number of concurrent users at time t, $X(t)$, can take the values $\{0, 1, 2\}$, the time changes by the minute (a discreet set), the probabilities of connecting/disconnecting depend only on the previous value of concurrent users and they are the same at any time (minute), so this is a homogeneous Markov chain with three states: $0, 1$ and 2.

a) Let us find each row of the transition probability matrix P.

For the first row, we want the transition probabilities form state 0 to each of the states $0, 1, 2$,

$$\text{Prow1} = [p_{00}\ \ p_{01}\ \ p_{02}].$$

If $X_0 = 0$, i.e. there are no users at time $t = 0$, then X_1, the number of new connections within the next minute is the number of successes in $n = 2$ trials, with probability of success ("to connect") $p = 0.2$, i.e. has $Bino(2, 0.2)$ distribution. We find it in Matlab with

```
>> Prow1 = binopdf(0:2,2,0.2)
Prow1 =
    0.6400    0.3200    0.0400
```

For the second row, suppose $X_0 = 1$, i.e one user is connected, the other is not. We compute each transition probability.

$$\begin{aligned}
p_{10} &= P\big((\text{the connected user disconnects}) \cap (\text{the disconnected user does not connect})\big) \\
&\overset{ind}{=} 0.5 \cdot 0.8 = 0.4, \\
p_{11} &= P\Big[\big((\text{the connected user does not disconnect}) \cap (\text{the disconnected user does not connect})\big) \\
&\quad \cup \big((\text{the connected user does disconnect}) \cap (\text{the disconnected user does connect})\big)\Big] \\
&= 0.5 \cdot 0.8 + 0.5 \cdot 0.2 = 0.5, \\
p_{12} &= P\big((\text{the connected user does not disconnect}) \cap (\text{the disconnected use does connect})\big) \\
&= 0.5 \cdot 0.2 = 0.1.
\end{aligned}$$

So, row 2 of P is

```
Prow2  = [0.4    0.5    0.1]
Prow2 =
    0.4000    0.5000    0.1000
```

Finally, if $X_0 = 2$, i.e. both users are connected, then no new users can connect and the number of disconnections is $Bino(2, 0.5)$ distributed, so

```
>> Prow3 = binopdf(2:-1:0,2,0.5)
Prow3 = 0.2500    0.5000    0.2500
```

Then the transition probability matrix is

```
>> P = [Prow1; Prow2; Prow3]
P =

    0.6400    0.3200    0.0400
    0.4000    0.5000    0.1000
    0.2500    0.5000    0.2500
```

b) Initial situation, at 10:00 a.m., there are 2 users connected, so

$$P_0 = [0 \;\; 0 \;\; 1].$$

At 10:02, after two steps, we want

$$P_2 = P_0 \cdot P^2,$$

in Matlab,

```
>> P_2 = [0 0 1] * P^2
P_2 =
    0.4225    0.4550    0.1225
```

So, the probability that at 10:02 there is one user connected is the second component of the vector above,

$$p_{21}^{(2)} = 0.455.$$

c) Noon is *many* minutes after 10:00, so we use the steady-state distribution, which we know exists, since P has all non-zero entries. We find it in Matlab. We write the system in the form

$$Ax = b,$$

with A the coefficients matrix, so a 4×3 (singular) matrix and b the right-hand side constant vector (of dimension 4×1). To set up A, notice that for the equations $\pi P = \pi$, the coefficients on the left are those of the *transpose* of P, P^T (in Matlab the transpose is prime, '). Then we have to subtract the π_1, π_2, π_3 from the right-hand side, i.e. the identity matrix I_3 and finally add a row of 1's, the coefficients from the last equation $\sum_{x=0}^{2} \pi_x = 1$. So the matrix A is

```
>> A = [P' - eye(3); 1 1 1]
A =
   -0.3600    0.4000    0.2500
    0.3200   -0.5000    0.5000
    0.0400    0.1000   -0.7500
    1.0000    1.0000    1.0000
```

and the vector b

```
>> b = [0; 0; 0; 1]
b =
     0
     0
     0
     1
```

Then we solve the system by

```
>> x = A\b
x =
    0.5102
    0.4082
    0.0816
```

So, the steady-state distribution is $\pi = [\pi_0 \;\; \pi_1 \;\; \pi_2] = [0.5102 \;\; 0.4082 \;\; 0.0816]$.
Then the expected nr. of connections by noon is the expected value of the random variable with pdf

$$\begin{pmatrix} 0 & 1 & 2 \\ 0.5102 & 0.4082 & 0.0816 \end{pmatrix},$$

i.e. $0 \cdot 0.5102 + 1 \cdot 0.4082 + 2 \cdot 0.0816 = 0.5714$, between 0 and 1, (slightly) more probable 1.

5. (Weather forecast) Recall the example at the lecture, about Rainbow City with sunny/rainy days (state 1 was “sunny” and state 2 was “rainy”), with transition probability matrix

$$P = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}.$$

a) If the initial forecast is 80% chance of rain, write a Matlab code to generate the forecast for the next 30 days.
b) In Rainbow City, if there are 7 days or more of sunshine, there is the danger of a water shortage, and if it rains for a week or more, there is the threat of flooding. Local authorities need to be prepared for each situation. Use the code from part a) to conduct a Monte Carlo study for estimating the probability of a water shortage and the probability of flooding.

Solution:

a) We simulate a Markov chain with 2 states, 1 and 2, initial distribution $P_0 = [0.2 \quad 0.8]$ and transition probability matrix P. We use Algorithm 2.11, Lecture 5.

Algorithm

1. Given:
   N_M = sample path size (length of Markov chain),
   $P_0 = [P_0(1) \ \ \ldots \ \ P_0(n)]$,
   $P = [p_{ij}]_{i,j=\overline{1,n}}$.
2. Generate X_0 from its pdf P_0.
3. Transition: if $X_t = i$, generate X_{t+1}, with probability $p_{ij}, j = \overline{1,n}$.
4. Return to step 3 until a Markov chain of length N_M is generated.

At the first step we simulate a Bernoulli variable with pdf

$$\begin{pmatrix} 1 & 2 \\ P_0(1) & P_0(2) \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0.2 & 0.8 \end{pmatrix},$$

i.e. $U < P_0(1)$, with probability $P_0(1)$ and $U \geq P_0(1)$, with probability $P_0(2) = 1 - P_0(1)$.
Then, at each step we do the same with updated vector P_0, from the the $X(t)^{\text{th}}$ row of matrix P.

```
% Simulate Markov chain in Problem 5, Seminar 4.
clear all
Nm = input('length of sample path  (of Markov chain) = ');
X = zeros(length(Nm)); % allocate memory for X
P0 = [0.2 0.8]; % initial distr. of sunny/rainy
P = [0.7 0.3; 0.4 0.6]; % trans. prob. matrix
P1(1, :) = P0; % P1 will contain forecast at each step;
               % first row contains the forecast in day t = 1

for t = 1 : Nm
```

```
    U = rand;
    X(t) = 1 * (U < P0(1)) + 2 * (U >= P0(1));
      % simulate X(1),... X(Nm) sequentially,
      % as Bernoulli variables taking value 1 with
      % prob. P0(1) and value 2 with prob. 1 - P0(1)
    P1(t + 1, :) = P1(t, : ) * P; % forecast for next day
    P0 = P(X(t),:); % prepare the distribution of X(t + 1);
                    % its pdf is the (X(t))th row of matrix P
end
X
```

This returns a sequence of states (of length 30) that looks like this:

$$1\ 1\ 1\ 2\ 2\ 2\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 1\ 2\ 2\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2\ 2\ 2$$

b) Now, we need to find any "long streaks" of sunny/rainy days. First, we find at which indexes of X (i.e. which days in the next 30) the chain changes states (i.e. the weather changes).

```
 i_change = [find(X(1 : end - 1) ~= X(2 : end)), Nm]
          % indexes where states change
```

For the sequence above, that returns

```
i_change =
     3     6    10    13    14    16    26    30
```

Next, we define a vector that contains all these "long streaks".

```
longstr(1) = i_change(1) % first long streak ends at
                         % the first change
for i = 2 : length(i_change)
    longstr(i) = i_change(i) - i_change(i - 1)
     % the remaining long streaks are differences
     % between any two changes
end
```

In my example, that returns

```
longstr =
     3     3     4     3     1     2    10     4
```

We now save long streaks of sunny and long streaks of rainy days in separate vectors, to see if any exceeds 7 days.

```
 if (X(1)==1)
        sunny = longstr(1 : 2 : end); % long streaks of sunny
        rainy = longstr(2 : 2 : end); % long streaks of rainy
 else
        sunny = longstr(2 : 2 : end); % long streaks of sunny
        rainy = longstr(1 : 2 : end); % long streaks of rainy
 end

 maxs = max(sunny) % longest streak of sunny days
 maxr = max(rainy) % longest streak of rainy days
```

In my simulation, I got

```
maxs =
    10
maxr =
     4
```

Now, if everything is ok, we put it all in a loop and simulate many chains, in order to estimate the probability of water shortage and that of flooding. **Make sure you first add ";" at the end of each variable or comment the ones you don't need to see!!**

```
% Simulate Markov chain in Problem 5, Seminar 4.
clear all
Nm = input('length of sample path  = ');
N = input('nr. of simulations = ');
for j = 1 : N
    X = zeros(length(Nm)); % allocate memory for X
    P0 = [0.2 0.8]; % initial distr. of sunny/rainy
    P = [0.7 0.3; 0.4 0.6]; % trans. prob. matrix
    P1(1, :) = P0; % P1 will contain forecast at each step;
```

```
              % first row contains the forecast in day t = 1
    for t = 1 : Nm
        U = rand;
        X(t) = 1*(U < P0(1)) + 2 *( U >= P0(1));
      % simulate X(1),... X(Nm) sequentially,
      % as Bernoulli variables taking value 1 with
      % prob. P0(1) and value 2 with prob. 1 - P0(1)
        P1(t + 1, :) = P1(t, : ) * P; % forecast for next day
        P0 = P(X(t),:); % prepare the distribution of X(t + 1);
                        % its pdf is the (X(t))th row of matrix P
    end
    % X
    i_change=[find(X(1:end-1) ~= X(2:end)), Nm];
    % indices where states change
    longstr(1) = i_change(1);
    % first long streak ends at the first change

    for i = 2 : length(i_change)
        longstr(i) = i_change(i) - i_change(i - 1);
     % the remaining long streaks are differences
     % between any two changes
    end

    if (X(1)==1)
        sunny = longstr(1 : 2 : end); % long streaks of sunny
        rainy = longstr(2 : 2 : end); % long streaks of rainy
    else
        sunny = longstr(2 : 2 : end); % long streaks of sunny
        rainy = longstr(1 : 2 : end); % long streaks of rainy
    end

    maxs(j) = max(sunny); % longest streak of sunny days
    maxr(j) = max(rainy); % longest streak of rainy days
end
```

```
% maxs;
% maxr;
```

Finally, we get our estimates.

```
fprintf('\n prob. of water shortage is %1.4f\n',mean(maxs >= 7))
fprintf(' prob. of flooding is %1.4f\n',mean(maxr >= 7))
```

Here are several runs:

```
>> probl5_sem4_SCM
 length of sample path  = 30
 nr. of simulations = 5e3

 probability of water shortage is 0.4652
 probability of flooding is 0.2236
>>
>> probl5_sem4_SCM
 length of sample path  = 30
 nr. of simulations = 1e4

 probability of water shortage is 0.4881
 probability of flooding is 0.2433
>>
>> probl5_sem4_SCM
 length of sample path  = 30
 nr. of simulations = 5e4

 probability of water shortage is 0.4687
 probability of flooding is 0.2321
>>
>> probl5_sem4_SCM
 length of sample path  = 30
 nr. of simulations = 1e5

 probability of water shortage is 0.4703
 probability of flooding is 0.2327
```

```
>>
>> probl5_sem4_SCM
 length of sample path  = 30
 nr. of simulations = 5e5

 probability of water shortage is 0.4734
 probability of flooding is 0.2339
```

(the last one took some time ...)

So, we estimate the probability of water shortage ≈ 0.47 and the probability of flooding ≈ 0.23.