STATISTICAL COMPUTATIONAL METHODS

Seminar Nr. 4, Markov Chains, Applications and Simulations

4. (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 drought, 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 2.13, 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.

First, give the initial data and allocate memory for the values of the Markov chain X:

```
% Simulate Markov chain in Problem 4, Seminar 4.
clear all
Nm = 30; % length of sample path (of Markov chain)
P0 = [0.2 0.8]; % initial distr. of sunny/rainy
P = [0.7 0.3; 0.4 0.6]; % trans. prob. matrix
```

```
X = zeros(1, Nm); % allocate memory for X
```

Then, define a 2-column matrix $P1$ that will contain the forecast at each step (on each row). The first forecast (i.e., the first row) will just be the initial distribution $P0$:

```
P1(1, :) = P0; % P1 will contain forecast at each step;
               % first row contains the forecast in day t = 1
```

Next we simulate a generalized 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 $P_0(2) = 1 - P_0(1)$.

```
U = rand;
X(t) = 1 * (U < P0(1)) + 2 * (U >= P0(1));
```

The forecast for the next day will be $P1(t) * P$.

```
P1(t + 1, :) = P1(t, : ) * P; % forecast for next day
```

Then, at each step we do the same with updated vector $P0$, from the $X(t)^{\text{th}}$ row of matrix P:

```
P0 = P(X(t),:); % updated vector P0;
                % prepare the distribution of X(t + 1);
                % its pdf is the (X(t))th row of matrix P
```

Now put it all together, for each time step $t = \overline{1, Nm}$:

```
% Simulate Markov chain in Problem 4, Seminar 4.
clear all

% Initial data
Nm = 30; % length of sample path  (of Markov chain)
P0 = [0.2 0.8]; % initial distr. of sunny/rainy
P = [0.7 0.3; 0.4 0.6]; % trans. prob. matrix

X = zeros(1, Nm); % allocate memory for X
```

```
P1(1, :) = P0; % P1 will contain forecast at each step;
               % first row contains the forecast in day t = 1

for t = 1 : Nm
    X(t) = 1 * (rand < P0(1)) + 2 * (rand >= 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),:); % updated vector P0;
                    % prepare the distribution of X(t + 1);
                    % its pdf is the (X(t))th row of matrix P
end
X
P1
```

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

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

and the matrix $P1$ with the pdfs for each day:

$$P1 =$$

$$\begin{array}{cc} 0.2000 & 0.8000 \\ 0.4600 & 0.5400 \\ 0.5380 & 0.4620 \\ 0.5614 & 0.4386 \\ \vdots & \end{array}$$

<u>**Note**</u>: You will **not** get the exact same thing!! It's a simulation ...

b) Now, we need to find any long runs, "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)  % the first day is sunny
        sunny = longstr(1 : 2 : end); % long streaks of sunny
        rainy = longstr(2 : 2 : end); % long streaks of rainy
 else   % the first day is rainy
        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 4, Seminar 4.
clear all
Nm = 30; % length of sample path (of the Markov chain);
N = input('nr. of simulations = ');
maxs = zeros(1, N);
maxr = zeros(1, N);
for j = 1 : N
    X = zeros(1, 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) % the first day is sunny
        sunny = longstr(1 : 2 : end); % long streaks of sunny days
        rainy = longstr(2 : 2 : end); % long streaks of rainy days
    else % the first day is rainy
        sunny = longstr(2 : 2 : end); % long streaks of sunny days
        rainy = longstr(1 : 2 : end); % long streaks of rainy days
    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 drought is %1.4f\n',mean(maxs >= 7))
fprintf(' prob. of flooding is %1.4f\n\n',mean(maxr >= 7))
```

Here are several runs:

```
>> probl4_sem4_SCM
 nr. of simulations = 5e3

 probability of drought is 0.4652
 probability of flooding is 0.2236
>>
>> probl4_sem4_SCM
 nr. of simulations = 1e4

 probability of drought is 0.4881
 probability of flooding is 0.2433
>>
>> probl4_sem4_SCM
```

```
 nr. of simulations = 5e4

 probability of drought is 0.4687
 probability of flooding is 0.2321
>>
>> probl4_sem4_SCM
 nr. of simulations = 1e5

 probability of drought is 0.4703
 probability of flooding is 0.2327
>>
>> probl4_sem4_SCM
 nr. of simulations = 5e5

 probability of drought is 0.4734
 probability of flooding is 0.2339
```

(the last one took some time ...)

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