Markov processes, lab 1

The aim of the lab is to demonstrate how Markov chains work and how one can use MATLAB as a tool to simulate and analyse them. This includes estimation of transition probabilities. The appendix contains the help texts for the tailor made procedures.

1 Preparations

Read through the instructions and answer the following questions. It is required that you bring your written answers with you to the computer lab. In the beginning of the computer session the lab supervisors will check your answers before you carry out the rest of the assignment.

1. A Markov chain satisfies a particular condition. Which one?

2. What is meant by the transition probability $p_{ij}$?

3. The transition probabilities are gathered in a matrix, the transition probability matrix, that is denoted by $P$. How large are the row sums of this matrix?

4. What is particular about an absorbing state? How can one find out from the transition probability matrix if a state is absorbing?

5. How can one compute the third order transition probabilities if the transition probability matrix $P$ is known?

6. What do the elements in $P^3$ mean?

7. How is a stationary distribution of a Markov chain defined?

8. What is the difference between a transient and a recurrent state?

9. If we know that state 1 is recurrent and intercommunicates with state 2, what can we say about state 2?

10. Give a condition for the existence of a stationary distribution of an irreducible Markov chain.

11. How can one compute the stationary distribution $\pi$ if the transition probability matrix $P$ is known?
12. We have observed a process that can be modelled as a Markov chain.
   (a) How can we estimate the transition probabilities?
   (b) What is the distribution of the number of transitions between two states, \(i\) to \(j\) say, conditional on the chain visiting state \(i\) \(n\) times?
   (c) How can one compute confidence intervals for the estimates in (a)?

13. During the lab you will examine Markov chains with the following three transition probability matrices:

\[
P_1 = \begin{pmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{2} & 0 & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & 0
\end{pmatrix}, \quad P_2 = \begin{pmatrix}
1 & 0 & 0 \\
\frac{1}{6} & \frac{1}{3} & \frac{1}{2} \\
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{pmatrix} \quad \text{and} \quad P_3 = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & \frac{1}{2} & \frac{1}{2} \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix}.
\]

Think about the following four questions for each of the three matrices before the lab.
   (a) Is the chain periodic? If yes, what is the period?
   (b) Is there one or many stationary distributions \(\pi\)?
   (c) As \(n \to \infty\), does \(p(n)\) converge to \(\pi\) irrespective of \(p(0)\)?
   (d) Can one expect to estimate the transition probabilities well?

14. \(P_2\) has one absorbing state. Compute the mean time until absorption when starting in any of the other states.

## 2 Markov chains

Log in at one of the PCs in the computer room MH:230 or MH:231 with your \textit{STIL}-account. Choose the latest version of \textit{Matlab} from the \textit{Start} menu. If you have problems either logging in or starting \textit{Matlab}, ask for help.

After starting \textit{Matlab} write \texttt{mh_init(’fmsf15’)} in the command window in \textit{Matlab}. This will add the directory containing the \texttt{m-files} which you will use in this assignment.

### 2.1 Transition probability matrix \(P_1\)

We start by examining the Markov chain with transition probability matrix

\[
P_1 = \begin{pmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{1}{2} & 0 & \frac{1}{2} \\
\frac{1}{2} & \frac{1}{2} & 0
\end{pmatrix}.
\]
2.1.1 Simulation

Use the procedure `simulering` to simulate 200 steps of the process. Make three simulations, each with its own initial state, and save the results in \( x_1 \), \( x_2 \) and \( x_3 \). To find out what a realisation of the first process looks like you can plot it using the command `stairs(x1)`. In order to magnify the plot you may change the ranges of the axes using the commands `zoom` or `axis`. If you want to plot the first 30 steps you can write

\[
\begin{align*}
\text{>> stairs}(x1) \\
\text{>> axis([1 30 0.5 3.5])}
\end{align*}
\]

In there any particular difference between the three realisations?

**Answer:**

Of course the initial state affects the process at the beginning—for how long, do you think, can one trace this effect?

**Answer:**

One may suspect that the process reaches equilibrium quite soon, so that the distribution of \( X_n \) stabilises; the process follows a stationary distribution. Estimate the stationary distribution by examining how long time the process spends in each state.

**Answer:**

2.1.2 Estimation of transition probabilities

Use `pestimering` to estimate the transition probability matrix from one of the simulated realisations.

**Answer:**

In order to compute confidence intervals one can assume that the distributions of the estimators are approximately normal. We know that a binomial distribution may be approximated by a normal one if \( npq > 10 \). What is the interpretation of \( n \), \( p \) and \( q \) in this case?

**Answer:**
Compute 95% confidence intervals for at least three of the transition probabilities. You may be helped by the extra output parameters from \texttt{pestimering} (\texttt{counts} and \texttt{jumps}).

\textbf{Answer:}

2.1.3 Distribution

The procedure \texttt{seep} shows in a histogram how the distribution of a Markov chain evolves in time.

Start the process in state 1, i.e. with the distribution (1, 0, 0). What happens?

\textbf{Answer:}

Start the process in some other state. Is there a difference?

\textbf{Answer:}

If a Markov chain has a stationary distribution, this satisfies

\[
\pi = \pi P
\]

\[
\sum \pi_i = 1
\]

\[
\pi_i \geq 0 \text{ for all } i.
\]

If the stationary distribution is unique one may compute it as follows:

```matlab
>> n=size(P,1);
>> [zeros(1,n),1]/[P-eye(n),ones(n,1)]
```

Think about how this expression works!

If one does not know if there is a unique stationary distribution one can use more powerful tools. The stationary distributions are left eigenvectors of \( P \). Left eigenvectors and corresponding eigenvalues can be computed using command \texttt{eigv}. The stationary distributions \( \pi \) are then obtained as linear combinations of eigenvectors with eigenvalue 1. The linear combinations must satisfy \( \sum \pi_i = 1 \) and \( \pi_i \geq 0 \) for all \( i \). If MATLAB reports "Warning: Rank deficient" when one tries to solve the system of equations, this method must be applied.
Compute the stationary distribution using any of the above methods.

**Answer:**

Start the process with its stationary distribution. Was the result as you expected?

**Answer:**

Is the process periodic?

**Answer:**

Compare your answer to the following theorem.

**Theorem (Perron-Frobenius):** If $P$ is the transition probability matrix of a finite, irreducible Markov chain with period $d$ ($d = 1$ if the chain is aperiodic),

(a) $\lambda_1 = 1$ is an eigenvalue of $P$.

(b) There are $d$ eigenvalues satisfying $|\lambda_j| = 1$.

(c) The remaining eigenvalues satisfy $|\lambda_j| < 1$.

**Answer:**

### 2.2 Transition probability matrix $P_2$

Now consider the transition probability matrix

$$P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1/6 & 1/3 & 1/2 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}.$$ 

#### 2.2.1 Simulation

Do three simulations again, with different initial states. Use `stairs` to plot them.

Is there any particular difference?

**Answer:**

Of course the initial state affects the process at the beginning—for how long, do you think, can one trace this effect?

**Answer:**
Estimate the stationary distribution by examining how long time the chain spends in each state.

2.2.2 Estimation of transition probabilities

Use `pestimering` to estimate the transition probability matrix using any of the realisations.

Can we make an approximation by a normal distribution? Is $npq > 10$ for all transitions? Why/why not?

How could one in “real life” construct confidence intervals for the transition probabilities of a Markov chain of this kind?

2.2.3 Distribution

Use `seep` and start the process in state 3, that is with the distribution $(0, 0, 1)$. What happens?

Answer:
Compute the stationary distribution (if it exists).

Answer:

Start the chain with its stationary distribution. Is the result what you expected?

Answer:

Is the process periodic?

Answer:

### 2.3 Transition probability matrix $P_3$

Now we examine the Markov chain with transition probability matrix

\[
P_3 = \begin{pmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1/2 & 1/2 \\
1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0
\end{pmatrix}.
\]

#### 2.3.1 Simulation

Do a few simulations, with different initial states. Is there any difference?

Answer:

Of course the initial state affects the process at the beginning—for how long, do you think, can one trace this effect? Compare two realisations with different initial states using \texttt{stairs} and \texttt{axis([0 15 0.5 4.5])}

Answer:

Does the process have a stationary distribution? If so, estimate it by examining how long time the chain spends in each state.

Answer:
2.3.2 Estimation of transition probabilities

Use \texttt{pestimering} to estimate the transition probability matrix using one of the simulated realisations.

Answer:

Can you approximate by a normal distribution? Is $npq > 10$? Use \texttt{pestimering} to count the number of jumps between the different states.

Answer:

2.3.3 Distribution

Use \texttt{seep} and start the chain in state one, that is with the initial distribution $(1, 0, 0, 0)$. What happens?

Answer:

Start the chain in some other state? Was there a difference?

Answer:

Compute the stationary distribution (if there is one) and use it in \texttt{seep}. Was the result as expected?

Answer:
Is the chain periodic? If yes, what is the period?

**Answer:**

Compare your answer to the Perron-Frobenius theorem.

**Answer:**

### 2.4 The Monopoly board game as a Markov chain

We recall the main rules of moving in (the Swedish version of) Monopoly:

- Each turn, the player throws two dice and moves his/her token as many steps as the sum of the dice.
- If the token lands on the space “Go to jail”, it is immediately moved to “jail”.
- If the token lands on a Community Chest space (*Allmänning, §*) or Chance space (*Chans, ?*), the player should draw a card from a pile. Some cards instruct the player to move to some other space.
- A player whose token is in jail throws the dice each turn, and if they agree he/she may move the token out of the jail. The token may always leave jail after three turns, however.

There are some additional rules that we neglect.

The probability of moving from the current space to another one may be described by a transition probability. This probability depends, among other things, on the probabilities of various sums of the dice and the probability of receiving a card that gives an instruction of a further move. The transition probabilities are collected into a transition probability matrix $P$.

The MATLAB-procedure `monop` defines the transition probability matrix, draws the board and one may simulate an arbitrary number of moves.

Make a few simulations using the command `simuler(1,P,n)` and translate them into interpretable lists by `monopgata`.

Use `seep(A,P,60)` to find out how the distribution of the chain evolves, in particular for the about 20 first steps.

**Answer:**
Run deep long enough that the stationary distribution is reached.

Why are some states (spaces) noticeably below/above the average level? Which are these spaces?

**Answer:**

Which states correspond to the jail? Why is not one state sufficient?

**Answer:**

How large portion of the time is spent in jail?

**Answer:**
A MATLAB-procedures

eigv

[V,D]=eigv(X)
Computes left eigenvectors (V) and corresponding eigenvalues (D)

monop

% monop
% Draws the Monopoly board.
% P contains the transition probability matrix, A the initial distribution.
% If a simulation is done, it is stored in z.

monopgata

% G=monopgata(x)
% Translates numerical states into street names.

pestimering

% [Phat,counts,jumps]=pestimering(x);
% Estimates the transition probability matrix and counts the number of
% jumps between the states.
% In data : Vector containing the realisation of the chain.
% Out data: Estimate of transition probability matrix (Phat)
% Vector of visits to the different states (counts)
% Matrix of counts between the different states (jumps)

porand

% n=porand(lambda);
% Generates a random variable from Po(lambda)

psimulering

% fordelning=psimulering(startp,P,n);
% A function that computes how the distribution evolves with time.
% In data : initial distribution (startp)
% transition probability matrix (P)
% number of steps (n)
% Out data: matrix in which each column contains the distribution
% at the different time points.

seep

% seep(startp,P,n) plots the evolution of the distribution of a
% Markov chain with transition probability Pm
% initial distribution startp, for n steps.

simulering

% x=simulering(startvarde,P,n);
% Simulates a Markov chain.
% In data : startvarde is the number of the initial state.
% Transition probability matrix (P)
% Number of steps (n)
% Out data: Vector of the states of the chain at time points 1 through n.
% The initial time point is 1.