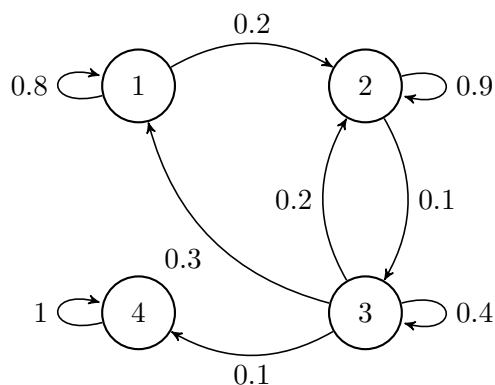


Palavras-chave: Cadeias de Markov, matriz de transição, matriz fundamental, estados absorventes, tempo até à absorção, simulação.

Nota: Adopte a definição da matriz de transição em que o elemento t_{ij} da matriz corresponde à probabilidade de transição do estado j para o estado i .

Considere o seguinte conjunto de páginas web ligadas entre si:



1. Escreva a matriz de transição H (de Hyperlinks), com H_{ji} sendo a probabilidade de ir da página i para a página j num único passo. Crie em Matlab/Octave essa matriz.
2. Qual a probabilidade de começando na página 1 ao fim de 1000 passos estar na página 2? Estava à espera deste valor?
3. Determine a probabilidade de chegar à página j a partir da página i , em 1,2,10 e 100 passos.
4. Determine a matriz Q .
5. Determine a matriz fundamental F .
6. Qual a média (valor esperado) do número de passos necessários para atingir a página 4 começando na página 1? e se começarmos na página 2? e se iniciarmos na página 3?
7. Qual o tempo até à absorção das páginas 1 a 3?
8. Modifique a matriz H para aumentar esse tempo (até à absorção) e recalcule Q , F e o tempo até à absorção.
9. Confirme os valores dos pontos anteriores através de simulação (faça a média de várias simulações). Use o código Octave no verso como base para criar a suas simulações.

```

# an example state transition matrix (page 3 is absorbing)
H = [0.9 0.1 0 ;
0.5 0.4 0.1 ;
0 0 1 ];
# the fundamental matrix
Q = H(1:2,1:2);
F = inv(eye(2)-Q)

# given a transition matrix and the current state,
# this function returns the next state
function state = nextState(H, currentState)
# find the probabilities of reaching all pages starting at the current one
probVector = H(:,currentState); # Attention: it is a column vector
# n is the number of pages, that is, H is n x n
n = length(probVector);
# pick the next page randomly according to those probabilities
state = discrete_rnd(1:n, probVector);
endfunction

# random walk on the graph according to state transition matrix H
# first = initial state, last = terminal or absorbing state
function state = crawl(H, first, last)
# the sequence of states will be saved in the vector "state"
# initially, the vector contains only the initial state
state = [first];
# keep moving from page to page until page "last" is reached
while (1)
state(end+1) = nextState(H, state(end));
if (state(end) == last) break; endif
endwhile
endfunction

# pick the next page randomly according to those probabilities
# states = vector with states (numbers), probVector = probability vector
function state = discrete_rnd(states, probVector)
#... To be developed

# how to use crawl()
state = crawl(H, 1, 3);

```