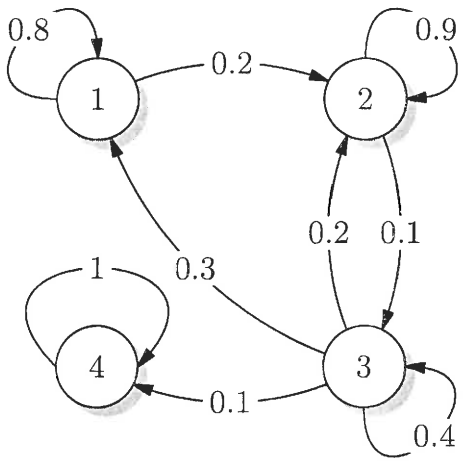
MPEI — crawl simulation

Paulo J S G Ferreira

3rd December 2014

Consider the following set of linked pages:



- Write the matrix H (transition probability matrix: H_{ij} is the probability of going from page i to page j, in one step).
- Find the probability of reaching page j starting from page i, in 1, 2, 10 and 100 steps.
- Find Q and the fundamental matrix $F = (I - Q)^{-1}$.
- What is the average number of steps required to reach state 4 starting from page 1? And if one starts in page 2? What about page 3?
- Confirm the values by simulation (average several runs). The `octave` code on the next page may help.
- Modify the matrix H to increase the absorption time.

futuro = nextState(H, actual) → determina o ponto seguinte do trajeto, seguindo o ponto actual

```
# an example state transition matrix (page 3 is absorving)
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,:);
    # 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, 1);
endfunction

# random walk on the graph according to state transition matrix H
# first = initial state, last = terminal or absorving 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

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

Cada crawler vai ter que possuir uma função deste tipo (discrete_rnd) que lhe obtiva uma posição seguinte tendo probabilidade

1º → nº de nós que vai retornar

→ é passado um conjunto de símbolos e a prob. de aparecerem

→ retorna valores inteiros aleatórios 1º [1 2 3] [0.5 0.4 0.1]

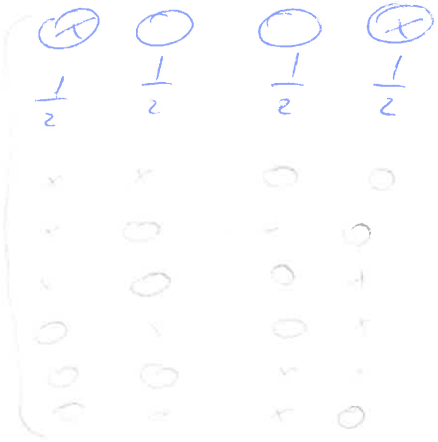
→ podemos fazer um ciclo e contar as vezes que passo por cada estado, antes de ir para o estado absorvente

Pedro Matos
nº73941

1º Trie
x = round(1,100) < 0,5
sum(x)
ans = 55

2º Trie
x = round(1,100) < 0,5
Sum(x)
ans = 44

3º Trie
x = round(1,100000) < 0,5
ans = 49975



Trabalhos a realizar

2.1
```
total = 0;
for k = 1 : 100
    m1 = sum(round(1,1000) < 0,5);
    maxm1 = max(m1, maxm1),
    total = total + m1;
end
total /100
```
ans = 500,15
maxm1 = 534
minm1 = 464

Para milerentos com contagem de 64:
```
...
for 1 : 1000
m1 = Sum (round(1,1000) < 1/64);
...
```
ans = 15 6,43
maxm1 = 26
minm1 = 3