**Universidade de Aveiro**

**MPEI 2020/21**

**2º guião para avaliação**

**Joaquim Andrade nº93432**

**Francisco Silva nº93400**



1.a)Abaixo encontra-se o código. Fizemos também uso da função, fornecida pelo docente, crawl, a qual alteramos e . A palavra criada foi “a”.

% Random walk on the Markov chain

% Inputs:

% H - state transition matrix

% 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 state to state until state "last" is reached:

while (1)

a=nextState(H, state(end));

if (a== last) %stops the atribution of the last state to the word

break;

end

state(end+1) = a;

end

end

m=[0 1/3 0 1/4 0; %r

1/2 0 1/2 1/4 0; %o

0 1/3 0 1/4 0; %m

1/2 0 1/2 0 0; %a

0 1/3 0 1/4 0]; %.

%cria uma matriz de transição

basedados= ['r','o','m','a',' '];%cria caracteres na mesma posição que na matriz

palavra= basedados(crawl(m,randi(4),5));%cria um caminho pela matriz e passa-o para uma palavra

1.b)Fazemos também uso da função crawl aqui. Os resultados das cinco palavras que mais apareceram foram:

'o' 0.0832810000000000

'a' 0.0622420000000000

'ro' 0.0418120000000000

'mo' 0.0414750000000000

'ra' 0.0313570000000000

O código foi o seguinte:

m=[0 1/3 0 1/4 0; %r

1/2 0 1/2 1/4 0; %o

0 1/3 0 1/4 0; %m

1/2 0 1/2 0 0; %a

0 1/3 0 1/4 0]; %.

%cria uma matriz de transição

a=cell(10e5, 1);%aloca espaço para 10e5 palavras em cell

for i=1: 10e5 %ciclo cria e aloca 10e5 palavras no cell criado anterior

a{i}=basedados(crawl(m,randi(4),5)) ;

end

pD= length(unique(a)); %número de palavras não repetidas

Mpu= unique(a);

[uc, ~, idc] = unique(a);

counts= accumarray(idc, ones(size(idc)));

M= cell(pD, 2);

for i=1:pD

M(i,1)=Mpu(i); %aloca as palavras não repetidas

M(i,2)= num2cell(counts(i)/10e5); %aloca a probabilidade de se repetirem no array original

end

f=cell(5,2);%5 palavras mais usadas e probabilidade

M=sortrows(M,2); % ordena as probabilidades

for i=1: 5

f(i,1)= M(pD-i+1,1); %guarda aas primeiras 5 palavras

f(i,2)= M(pD-i+1,2); %guarda as primeiras 5 probabilidades

end

1.c)Cálculos das probabilidades teóricas das 5 palavras mais geradas:

'o' 0,25\*1/3 = 0,083333333

'a' 0,25\*0,25 = 0,0625

'ro' 0,25\*0,5\*1/3 = 0,041666666

'mo' 0,25\*0,5\*1/3 = 0,041666666

'ra' 0,25\*0,5\*0,25 = 0,03125

Comparando estes valores com os obtidos na alínea anterior observamos que são muito idênticos e por isso podemos concluir que a simulação foi bem sucedida.

1.d)Ao código da pergunta 1.b) adicionámos o seguinte excerto de código que abre o ficheiro de palavras, interseta-o com as palavras geradas e coma as probabilidades de cada palavra ser gerada. Essa probabilidade deu 0.3531.

fid=fopen('wordlist-preao-20201103.txt');

data=textscan(fid,'%s');

fclose(fid);

g= data{1}(1:end);%abrir e ler o ficheiro para uma célula

a=0;

h=intersect(g, Mpu);%interseta g com as palavras geradas

for i=1: length(Mpu)

if ismember(M(i,1), h)==1%se existir uma palavra em M

a=a+cell2mat(M(i,2)); %somamos a probabilidade dessa palavra

end

end

1.e)Esta função crawl não adiciona o ultimo estado à palavra e para quando n é alcançado.

function state = crawl2(H, first, last, n)%add n

state = [first];

d=1;

while (1)

a=nextState(H, state(end));

if (a== last || n==d)%stops the atribution of the last state to the word

break;

end

state(end+1) = a;

d=d+1;%

if (state(end) == last || n==d)

break;

end

end

end

2

3

4-

5-