Probabilidades y entropía de una fuente, parte 3: fuente con memoria

En esta segunda parte se modificó el procesamiento del texto ya que las probabilidades de los estados y que obtenía eran demasiado pequeños e incoherentes. Para la sección de lectura y limpieza del libro, reemplazé todos los símbolos que no pertenieran a mi alfabeto por un caracter vacío en lugar de reemplazar uno por uno.

```
clear all
close all

%------------  LECTURA Y LIMPIEZA DEL LIBRO ----------------------
%s = input('Ingrese nombre del archivo con extension .txt: ', 's');
s='women.txt';
fileID = fopen(s,'r');
A = fscanf(fileID,'%c');
A = lower(A);  %Pasa a minusculas
A = regexprep(A, '[^a-z .?!-]', '');
N=length(A);

[alphabet,~,idx] = unique(A);% encontrar los caracteres unicos
freq = histcounts(idx,numel(alphabet)); % Obtener la frecuencia de cada elemento
simb=length(alphabet); % Cardinalidad del alfabeto

```

Matriz de transiciones

Si el alfabeto $\Omega = \{, !, -, ., ?, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$ y $\#\Omega = 31$, quiere decir que la matriz de transiciones es de 31×31 como se muestra a continuación

$$matrix\_tran = \begin{bmatrix} P(\Omega(0)|\Omega(0)) & \cdots & P(\Omega(0)|\Omega(31)) \\ \vdots & & \vdots \\ P(\Omega(31)|\Omega(0)) & \cdots & P(\Omega(31)|\Omega(31)) \end{bmatrix} \tag{1}$$

Primero en un ciclo for, se contabilizan los pares de símbolos (que representan los estados) y una vez llena la columna, se suman todos los valores para obtener el número de veces que cierto símbolo fijo apareció después de otro . Así es posible obtener en la misma variable `matrix_tran` las probabilidades de los estados a partir de su frecuencia.

```
%Llena primero columnas y luego filas
for i = 1:simb
  for j = 1:simb
  pair = [alphabet(j) alphabet(i)];
  idx = strfind(A, pair);
  matrix_tran(j,i) = numel(idx);
  end
states = sum(matrix_tran(:,i));
matrix_tran(:,i) =matrix_tran(:,i)./states;
end
```

Esta matriz es una representación del sistema de ecuaciones de la forma:

$$\begin{array}{ccccccc} P_{\Omega(0)} & = & P_{\Omega(0)}P(\Omega(0)|\Omega(0)) & + & \cdots & P_z P(\_|z) \\ \vdots & & & & & \vdots \\ P_z & = & P_z P(z|\_) & + & \cdots & P_z P(z|z) \end{array}$$

Por lo que hay que considerar la condición de normalización y sustituirla en cualquier fila de la matriz para obtener la solcuión del vector de probabilidades. Para este programa se sustituyó en la primera fila.

Vector de Probabilidades

La solución queda de la siguiente forma:

$$\begin{bmatrix} P(\Omega(0)|\Omega(0)) & \cdots & P(\Omega(31)|\Omega(0)) \\ P(\Omega(0)|\Omega(1)) & \cdots & P(\Omega(31)|\Omega(1)) \\ \vdots & & \vdots \\ P(\Omega(0)|\Omega(31)) & \cdots & P(\Omega(31)|\Omega(31)) \end{bmatrix} \begin{bmatrix} P_{\Omega(0)} \\ P_{\Omega(1)} \\ \vdots \\ P_{\Omega(31)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2}$$

```
1
2  matrix_tran(1,:) = 1;
3  for i=2:simb
4    matrix_tran(i,i) = matrix_tran(i,i)- 1;
5  end
6
7  prob_states= zeros(simb,1);
8  r = [1 ; zeros(simb-1,1)];
9  %-- Verificamos que la matriz tiene inversa con su determinante
10 if (det(matrix_tran) ~= 0)
11   disp('La matriz de trancisiones tiene inversa')
12   prob_states = matrix_tran\r;
13 end
```

Finalmente se obtiene el vector de probabilidades con el que se puede calcular la Entropía de los estados y la entropía total.

```
1    %--------- ENTROPiA ----------
2    for i=1:simb
3      H(i) = -prob_states(i)*log2(prob_states(i));
4    end
5
6    T=table(alphabet',H');
7    T.Properties.VariableNames = ["Alfabeto","Entropia"]
8    disp(['Total de informacion en ' num2str(N) ' simbolos con memoria 1 = ' num2str(
       sum(H)) ' bits/simb'])
```

Resultados

"Para generar el texto, se utilizó la función `randsrc`, la cual permite obtener números dentro de un rango $[1, \Omega]$ a partir del vector de probabilidades de los estados.

```
  %--- GENERACIoN DE TEXTO--------
% Generar texto de 100 caracteres
caracters =300;
x = 1:simb;
text_a = round(randsrc(1, caracters, [x; prob_states']));

text='';
for i=1:caracters
  text =  [text alphabet(text_a(i))];
end

disp('Texto generado a partir del analisis:')
disp(text)
```

A continuación, se presenta el texto generado:

> st tn ciuv wuhr etrvn stnrpen o c sodu ferr.hfnopcnn hvelapbhotvi bsbsateaamheslitcqpej irt kh e rdi lbnopf at siy f v rst ebret.b tli smehe a .ae tuhhete reae uul aes nonfidpao.enr trhffdhhsr-nesieutimladhasaa afdte.twf belbloa ddeh br tohlagviiahnneey ttr ben o nhrotteoa phhtmmsrl.at h wohh

Y la entropía de la fuente con memoria 1

$$\mathbf{H(S) = 4{,}149} \text{ bits/símb}$$

Conclusión

Seguí la metodología vista en clase, pero el vector de probabilidades de estados cuando la fuente tiene memoria 1 y cuando la fuente no tiene memoria es la misma, me hace falta generalizar para $\mathbf{n}$ memorias y corroborar si mi código está correcto o incorrecto.
Suponiendo que es correcto, quiere decir que que la fuente no tiene memoria porque las probabilidades condicionales son iguales a las probabilidades de los símbolos, lo cual es completamente erróneo y es signo de que se debe "debuggear.[el] código puntualmente.

```
T =

  31×2 table

    Alfabeto    Entropía
    ________    _________

                  0.44088
       !        0.0033113
       -         0.019056
       .         0.067301
       ?        0.0029489

       :            :

       v         0.050471
       w          0.11239
       x         0.011184
       y         0.082168
       z        0.0027818

    Display all 31 rows.

Total de información en 317128 símbolos = 4.149 bits/símb
Total de información en 317128 símbolos EQUIPROBABLES= 4.9542 bits/símb
La matriz de trancisiones tiene inversa

T =

  31×2 table

    Alfabeto    Entropía
    ________    _________

                  0.44089
       !        0.0032783
       -         0.019056
       .         0.067302
       ?        0.0029489

       :            :

       v         0.050471
       w          0.11239
       x         0.011184
       y         0.082168
       z        0.0027818

    Display all 31 rows.
```

```
Total de información en 317128 símbolos con memoria 1 = 4.149 bits/símb
Texto generado a partir del análisis:
st tn ciuv wuhr etrvn stnrpen o c  sodu ferr.hfnopcnn  hvelapbhotvi↙
bsbsateaamheslitcqpej  irt kh e rdi lbnopf at siy f v rst ebret.b  tli smehe a .ae↙
tuhhete reae  uul aes  nonfidpao.enr trhffdhhsrnesieutimladhasaa afdte.twf belbloa ddeh↙
br tohlagviiahnneey ttr ben o nhrotteoa phhtmmsrl.at h wohh

T =

  31×2 table

    Alfabeto    Entropía
    ________    _________

                  0.44089
       !         0.0032783
       -          0.019056
       .          0.067302
       ?         0.0029489
       a            0.25723
       b          0.072296
       c             0.124
       d           0.18351
       e            0.3422
       f           0.11268
       g            0.0918
       h           0.20584
       i           0.22382
       j          0.010311
       k          0.039278
       l           0.15108
       m           0.11243
       n           0.23547
       o           0.24635
       p          0.093346
       q          0.010444
       r           0.22739
       s           0.22124
       t           0.27816
       u           0.11761
       v          0.050471
       w           0.11239
       x          0.011184
       y          0.082168
       z         0.0027818

>>
```