SERRANO ARRESE, JULIA

Ejercicio. Entrega. Calcular e interpretar el pagerank de un grafo 6x6

Realizar y contestar las siguientes cuestiones.
Se debe entregar el grafo, el código utilizado, los resultados obtenidos y contestar a las preguntas.
Entregar el código y volcar los resultados.
Responder a las preguntas.
Contestar a las preguntas después del enunciado.

Sea la siguiente matriz de conectividad

$$C = \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{matrix}$$

Dibujar el grafo dirigido 6x6 que corresponde con dicha matriz.

En la entrega utilizar la notación: P1 -> P2

Calcular el vector Nj=sum(C). Calcular la matriz A. Calcular la matriz S.

Aplicaciones Numéricas Ejercicio 2023

SERRANO ARRESE, JULIA

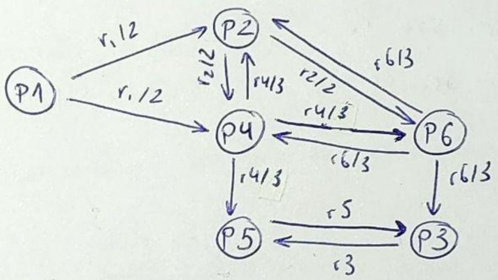
JULIA SERRANO ARRESE

EJERCICIO. CALCULAR E INTERPRETAR EL PAGERANK DE UN GRAFO 6x6

1. Dibujar el grafo dirigido 6x6 correspondiente a la matriz C.

$$C = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

P1 → P2 P2 → P4 P3 → P5 P4 → P2 P5 → P3 P6 → P2
P1 → P4 P2 → P6 P4 → P5 P6 → P3
P4 → P6 P6 → P4



2. Calcular el vector N_j

$N_j = \text{sum}(C)$ $N_j = [\,2 \;\; 2 \;\; 1 \;\; 3 \;\; 1 \;\; 3\,]$ Suma de las columnas de C

3. Calcular la matriz A

$A = C \,./\, N_j$

$$A = \begin{pmatrix} 0/2 & 0/2 & 0/1 & 0/3 & 0/1 & 0/3 \\ 1/2 & 0/2 & 0/1 & 1/3 & 0/1 & 1/3 \\ 0/2 & 0/2 & 0/1 & 0/3 & 1/1 & 1/3 \\ 1/2 & 1/2 & 0/1 & 0/3 & 0/1 & 1/3 \\ 0/2 & 0/2 & 1/1 & 1/3 & 0/1 & 0/3 \\ 0/2 & 1/2 & 0/1 & 1/3 & 0/1 & 0/3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0.\hat{3} & 0 & 0.\hat{3} \\ 0 & 0 & 0 & 0 & 1 & 0.\hat{3} \\ 0.5 & 0.5 & 0 & 0 & 0 & 0.\hat{3} \\ 0 & 0 & 1 & 0.\hat{3} & 0 & 0 \\ 0 & 0.5 & 0 & 0.\hat{3} & 0 & 0 \end{pmatrix}$$

4. Calculamos vector d_j

$$d_j(K) = \begin{cases} 1 & \text{si } N_j(K) = 0 \\ 0 & \text{si } N_j(K) \neq 0 \end{cases}$$

Como $N_j(K)\ \forall K \Rightarrow d_j = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$

SERRANO ARRESE, JULIA

Sea $r = (r_i)_{i=1}^{6}$ el vector de prioridades. Escribir las 6 ecuaciones lineales del sistema:

$$S\,r = r \qquad (1)$$

Escribir lar ecuaciones con la notación r1+r2+…=r3 A partir de las ecuaciones anteriores: Si es posible, calcular el valor de alguna componente del vector r. Si es posible, ordenar alguna componente del vector r.

5. Calculamos matriz transición modificada S.

$$S = A + \frac{1}{N} \cdot e \cdot d_j$$

Como $d_j = 0 \Rightarrow S = A$

6. Escribir las 6 ecuaciones lineales del sistema $Sr = r$

$$r_i = \sum_{j \in L_i} \frac{r_j}{N_j}, \; i = 1, \dots n \qquad \text{donde } L_i = \{ P_j \text{ que verifican } P_j \to P_i \}$$

Por lo tanto:

$$\begin{cases} r_1 = 0 \\ r_2 = r_1/2 + r_4/3 + r_6/3 \\ r_3 = r_5 + r_6/3 \\ r_4 = r_1/2 + r_2/2 + r_6/3 \\ r_5 = r_3 + r_4/3 \\ r_6 = r_2/2 + r_4/3 \end{cases} \qquad \begin{cases} 1r_1 = 0 \\ 1/2\, r_1 - r_2 + \quad + 1/3\, r_4 + \quad + 1/3\, r_6 = 0 \\ 1r_3 + \quad + 1r_5 + 1/3\, r_6 = 0 \\ 1/2\, r_1 + 1/2\, r_2 + \quad + 1r_4 + \quad + 1/3\, r_6 = 0 \\ 1r_3 + 1/3\, r_4 + 1r_5 = 0 \\ 1/2\, r_2 + \quad + 1/3\, r_4 + 1r_6 = 0 \end{cases}$$

Ponemos las ecuaciones en forma de matriz para aplicar el método de Gauss-Jordan:

$$\left(\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1 & 0 & 1/3 & 0 & 1/6 & 0 \\ 0 & 0 & -1 & 0 & 1 & 1/3 & 0 \\ 1/2 & 1/2 & 0 & -1 & 0 & 1/3 & 0 \\ 0 & 0 & 1 & 1/3 & -1 & 0 & 0 \\ 0 & 1/2 & 0 & 1/3 & 0 & -1 & 0 \end{array}\right) \underset{\sim}{\text{aplicando las transformaciones}} \left(\begin{array}{cccccc|c} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Por lo tanto, hemos hallado el valor de las siguientes componentes:

$\underline{r_1 = 0}$

$\underline{r_2 = 0}$

$r_3 = r_5$

$\underline{r_4 = 0}$

$r_5 = r_5$

$\underline{r_6 = 0}$

$P_1 = P_2 = P_4 = P_6$

$P_3 = P_5$

SERRANO ARRESE, JULIA

¿Verifica la matriz S el teorema de Perron-Frobenius? Comprobarlo. Indicar claramente el motivo.
Sí lo verifica, ya que se trata de una matriz irreducible, estocástica por columnas y no negativa.

Comprobación:

- Como se puede ver en el grafo dibujado, se puede acceder a cualquier nodo del grafo desde cualquier otro nodo del grafo, por lo que se verifica que se trata de una matriz irreducible
- La matriz no tiene ningún elemento negativo y la suma de los elementos de sus columnas es 1, por lo que se verifica las condiciones de: no negativa y estocástica

Comprobación suma columnas:
1 → 0.5 + 0.5 = 1
2 → 0.5 + 0.5 = 1
3 → 1
4 → ⅓ + ⅓ + ⅓ = 1
5 → 1
6 → ⅓ + ⅓ + ⅓ = 1

Por lo tanto, la matriz S verifica el teorema de Perron-Frobenius.

Hacer con Matlab.

Se ha realizado los apartados anteriores también en matlab con este código:

```
%Ejercicio. Calcular e interpretar el pagerank de un grafo 6x6
% Matriz de conectividad
C = [0 0 0 0 0 0;
    1 0 0 1 0 1;
    0 0 0 0 1 1;
    1 1 0 0 0 1;
    0 0 1 1 0 0;
    0 1 0 1 0 0];
N = length(C);
% Vector Nj
Nj = sum(C);
% Matriz transición A
A = C./Nj;
% comprobamos que la suma de las columnas sea 1
sum(A);     %=1
% calcular dj
Dj=zeros(1,N);
Dj(find(Nj==0))=1;
%calcular la matriz de transición modificada S
S=C;
% e vector columna de 1's
e = ones(N, 1);
% Construir la matriz S
S = A + ((1/N) * e * Dj);
```

A partir del estado r=(1 0 0 0 0 0), calcular el estado 100 de la tabla de estados para la matriz S, esto es, realizar 100 iteraciones r=S*r. ¿Cuál es el valor de r? Calcular sum(r). ¿r verifica la ecuación (1)?.

```
%Calcular el estado 100 de la tabla de estados
r = [1 0 0 0 0 0]';
for i = 1:100
   r = S*r;
end
% Calcular la suma de r
suma_r = sum(r);

S_r = (S*r)'
```

SERRANO ARRESE, JULIA

El valor de r es:
0 0.0000 0.3750 0.0000 0.6250 0.0000
y sum(r) = 1
S*r es: 0 0.0000 0.6250 0.0000 0.3750 0.0000

Se verifica que todos los elementos de r son iguales a S*r menos los 3 y 5 que están intercambiados.

Calcular **la matriz de Google G** para un alfa=0.85.

¿Verifica la matriz G el teorema de Perron-Frobenius? Comprobarlo.

Sea r el pagerank de la matriz G:

$$G\,r = r$$

```
%Calcular la matriz de Google G para un alfa=0.85.
alfa = 0.85;
G=getG(S, alfa);
sum(G);     %=1
```

G =

0.0250	0.0250	0.0250	0.0250	0.0250	0.0250
0.4500	0.0250	0.0250	0.3083	0.0250	0.3083
0.0250	0.0250	0.0250	0.0250	0.8750	0.3083
0.4500	0.4500	0.0250	0.0250	0.0250	0.3083
0.0250	0.0250	0.8750	0.3083	0.0250	0.0250
0.0250	0.4500	0.0250	0.3083	0.0250	0.0250

La matriz G sí verifica el teorema de Perron-Frobenius, ya que se trata de una matriz irreducible, estocástica por columnas y no negativa.

Comprobación:

- Se trata de una matriz irreducible
- La matriz no tiene ningún elemento negativo y la suma de los elementos de sus columnas es 1, por lo que se verifica las condiciones de: no negativa y estocástica
-

(la comprobación de que es estocástica se ha hecho en matlab con sum(G) que ha dado 1 por lo que queda verificado)

Vamos a calcular el vector r. Partir del vector r=(1 0 0 0 0 0) y realizar 100 iteraciones r=G*r ¿cuál es el valor final de r? Calcular sum(r). Calcular

$$norm(G * r - r)$$

```
r = [1 0 0 0 0 0]';
for i = 1:100
   r = G*r;
end
```

Valor final de r: 0.0250 0.0898 0.3465 0.0997 0.3477 0.0914

```
norm(G*r-r);    %2.8222e-08
sum(r);         %1
```

Aplicaciones Numéricas Ejercicio 2023

SERRANO ARRESE, JULIA

Calcular el pagerank r que verifique la siguiente condición de tolerancia, con tol=1e-15

$$norm(G * r - r) < tol$$

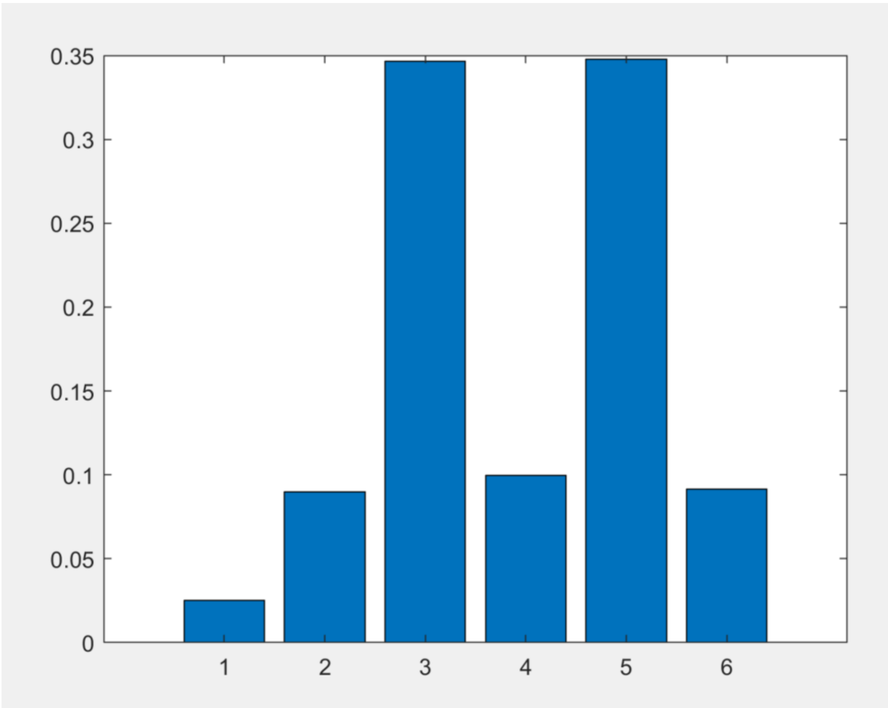
```
n = length(G);
tol=1e-15;
x0=rand(n,1);
[lambda,r,v_err,iter,IC] = metodoPotenciaTol(G,x0,tol);
```

El pagerank r es: 0.0250 0.0898 0.3465 0.0997 0.3477 0.0914

Ordenar las 6 páginas del grafo según su pagerank.

```
orden_pagerank = get_ranking(r');  %5 3 4 6 2 1

bar(r)
```



¿Verifica el vector r las ecuaciones (1)? Explicar el motivo.

r no verifica las ecuaciones (1), ya que no se cumple que: P1 = P2 = P4 = P6 y P3 = P5

Aplicaciones Numéricas **Ejercicio 2023**

SERRANO ARRESE, JULIA

Calcular las siguientes probabilidades en **un paso** (el surfista aleatorio va de un nodo a otro con un solo link)

¿Cuál es la probabilidad $p(P_5 -> P_4)$? ¿Cuál es la probabilidad $p(P_4 -> P_2)$?

$p(P_5 -> P_4)$ = a45 = 0

$p(P_4 -> P_2)$ = a24 = 1/3

Calcular las siguientes probabilidades en **dos pasos** (el surfista aleatorio va de un nodo a otro con un link y luego a otro nodo con otro link)

¿Cuál es la probabilidad $p(P_5 -> P_4 -> P_2)$? Este es, ¿Cuál es la probabilidad que un surfirsta que está en el nodo P5 pase al nodo P4 y luego al nodo P2?.

$p(P_5 -> P_4 -> P_2)$ = 0 * 1/3 = 0

¿Cuál es la probabilidad $p(P_5 -> ... -> P_2)$?. Este es, ¿Cuál es la probabilidad que un surfirsta que está en el nodo P5 pase a cualquier otro nodo (P1, P2, P3,…., ó P6) y luego pase al nodo P6?

Con el enunciado no me queda claro si el nodo final es P2 o P6, aunque ambas opciones tienen 0 de probabilidad, ya que P5 y P3 están conectados entre ellos pero no van hacia ningún otro nodo.

SERRANO ARRESE, JULIA

Funciones auxiliares utilizadas:

```
function G=getG(S, alfa)
   N=size(S,1);
   e=ones(N,1);
   G=alfa*S+(1-alfa)*(1/N)*(e*e');
return
```

```
function [lambda,x,v_err,iter,IC] = metodoPotenciaTol(A,x0,tol)
% calcula el autovalor dominante y su autovector asociado
% de la matriz A utilizando el metodo de la potencia.
% El metodo continua iterando hasta que la diferencia entre 2
% estimaciones consecutivas del autovalor sea <= que la tol o
% se alcance nmax
%   Entradas:
%   - A: Matriz cuadrada de tamaño nxn.
%   - x0: Vector inicial de tamaño n.
%   - tol: Tolerancia para la convergencia.
%   - nmax: Número máximo de iteraciones.
%
%   Salidas:
%   - lambda: Autovalor dominante.
%   - x: Autovector asociado al autovalor dominante.
n = size(A,1);
if nargin == 1
   x0=rand(n,1); % Vector de arranque, por ejemplo vector aleatorio
   nmax=10;% N° iteraciones a realizar
   tol = 1e-06; % Tolerancia
end
x0=x0/norm(x0); %normalizamos vector de arranque
x1 = A*x0;
lambda= x0'*x1; % Otra opción lambda= norm(x1);
err = tol + 100;
iter=0;
% while err > tol && iter <= nmax
while norm(A*x1-x1) > tol
   x=x1; x=x/norm(x);
   x1=A*x; lambda_new= x'*x1; % Otra opción lambda= norm(x1);
   err = abs(lambda_new - lambda);
   lambda=lambda_new;
   iter = iter + 1;
   v_err(iter) = err;
   % Imprimir información en cada iteración
   fprintf('Iter: %2d  Lambda: %f  Error: %e\n', iter, lambda, err);
end
x = x/sum(x);
IC = (lambda-n)/(n-1);
fprintf('------------------------------\n\n')
```

```
function indices_ordenados = get_ranking(vector)
   [~, indices_ordenados] = sort(vector, 'descend');
end
```