1 FUNDAMENTO TEÓRICO

Supongamos que a Alice y a Bob les dan seis páginas de internet que contienen palabras clave de un tema que les interesa a ambos. Cada uno tiene su propia estrategia a la hora de establecer el orden de importancia de las páginas.

1.1 Estrategia de Alice

Alice decide que la red de *enlaces* (*referencias*) entre las páginas puede ser un medio de medir su importancia relativa, de modo que dibuja un diagrama llamado *grafo de web* (*webgraph)* en el que se muestran los enlaces entre las seis páginas (Figura 1). Un enlace directo de la página $i-ésima$ a la página $j-ésima$ significa que la $i-ésima$ página posee un enlace de salida hacia la $j-ésima$, es decir, referencia esa página.

Alice procede de la siguiente forma:

- Ignora enlaces desde o hacia páginas que no sean las seis consideradas.
- Ignora enlaces de una página a sí misma.

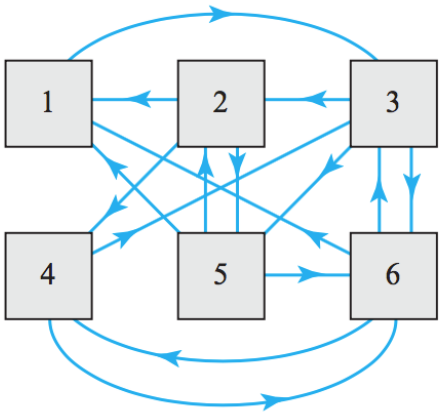


Figura 1

- Ignora enlaces duplicados
- Asume que no existen *páginas colgadas*, es decir, páginas sin enlaces de salida.

Alice diseña una estrategia de navegación en la web según la cual toma una de las páginas (por ejemplo, la 2) pincha en uno de sus enlaces y conecta con otra página. Repite el proceso desde la nueva página navegando de una página a otra. Registra cuántas veces visita cada página después de 10, 100, 1.000, 10.000 y 20.000 clics y lo refleja en la Tabla 1 (Nótese que el número de páginas visitadas es uno más que el número de "clicks").

	Número total de "clicks"					
Pág.	**0**	**10**	**100**	**1,000**	**10,000**	**20,000**
1	0	3	21	165	1,504	3,012
2	1	2	16	148	1,391	2,790
3	0	3	27	271	2,706	5,424
4	0	0	4	100	1,096	2,206
5	0	2	22	155	1,415	2,745
6	0	1	11	162	1,889	3,824

Tabla 1

También crea la Tabla 2 calculando la fracción de visitas (tanto por 1) a cada página con 4 cifras decimales.

	Número total de "clicks"					
Pág.	**0**	**10**	**100**	**1,000**	**10,000**	**20,000**
1	0.0000	0.2727	0.2079	0.1648	0.1504	0.1506
2	1.0000	0.1818	0.1584	0.1479	0.1391	0.1395
3	0.0000	0.2727	0.2673	0.2707	0.2706	0.2712
4	0.0000	0.0000	0.0396	0.0999	0.1096	0.1103
5	0.0000	0.1818	0.2178	0.1548	0.1415	0.1372
6	0.0000	0.0909	0.1089	0.1618	0.1889	0.1912

Tabla 2

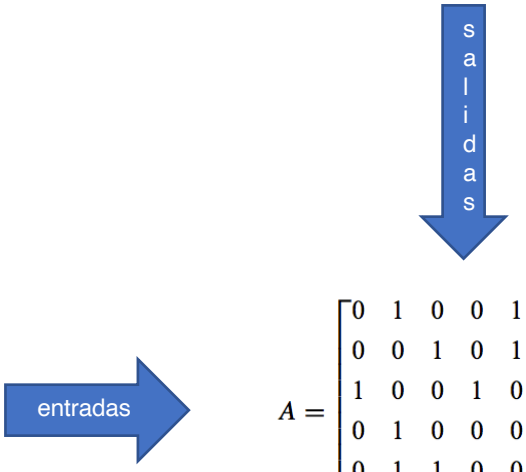
Basándose en 20.000 repeticiones identifica a la página 3 como la más importante ya que es la más visitada y establece un orden de importancia de las páginas:
$3, 6, 1, 2, 5, 4.$

1.2 Interpretación mediante matrices de Markov

Observamos que para cada página de la Tabla 2 las cifras tienden a estabilizarse. Esto no es un fenómeno accidental; veremos que en este ejemplo, independientemente de la página que se elija al principio y los enlaces seleccionados posteriormente el % de visitas a cada página convergerá a un valor límite que depende únicamente de al estructura del grafo de web. Los valores limite de esas fracciones, llamados *page ranks* pueden emplearse para medir la importancia relativa de las páginas.

Aunque el procedimiento que emplea Alice es apropiado para una red pequeña como la suya, no es factible en grandes grafos de web como internet (www). Para éstos podemos obtener una clasificación como la de Alice empleando cadenas de Markov. Como primer paso para explicar el método se define la *matriz de adyacencia* de un grafo de web con n páginas como una matriz nxn cuyo elemento a_{ij} es un 1 si la página $j - ésima$ posee un enlace hacia la página $i - ésima$ y un 0 en caso contrario[1]. Por ejemplo, la matriz de adyacencia del grafo de web de la Figura 1 es:

[1] *Sin embargo, a la hora de dibujar grafos dirigidos, Matlab opera como si la matriz de adyacencia fuera la traspuesta, es decir, las entradas las toma de las columnas y las salidas de las filas. Téngase esto muy en cuenta al dibujar grafos con Matlab*



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Obsérvese que:

- La suma de los elementos de la $i-ésima$ fila representa el numero de enlaces entrantes a la página i procedentes de otras páginas
- La suma de los elementos de la $j-ésima$ columna representa el número de enlaces salientes de la página j hacia otras páginas.

Definición 1. Si navegamos a través de un grafo de web con n páginas, el *vector de estado* $\boldsymbol{x}^{(k)}$ es un vector columna de $n\ x\ 1$ cuyo elemento $i-ésimo$ es la probabilidad de que estemos en la página i después de k "clicks".

Para ilustrar esta idea supongamos que conocemos con certeza que quien navega por la web se encuentra en la página j después de k "clicks", en cuyo caso el elemento $j-ésimo$ del vector $\boldsymbol{x}^{(k)}$ es 1 y todos sus otros elementos son 0. De ello se deduce que el producto $A\boldsymbol{x}^{(k)}$ es el $j-ésimo$ vector columna de la matriz A. Por ejemplo, si sabemos que Alice comienza en la página 2, su vector de estado inicial $\boldsymbol{x}^{(0)}$y el producto $A\boldsymbol{x}^{(0)}$ sería:

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad A\mathbf{x}^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{1}$$

Los elementos de $A\boldsymbol{x}^{(0)}$ indican que desde la página 2 puede ir a cualquiera de las páginas 1, 4 ó 5 ya que son las únicas a las que se enlaza la página 2. Suponiendo que Alice elige aleatoriamente cualquiera de los tres enlaces, cada uno de ellos tendrá una probabilidad de $\frac{1}{3}$ de ser elegido.

La siguiente definición formaliza la idea de que Alice elija aleatoriamente un enlace de salida.

Definición 2. La *matriz de probabilidad de transición* $B = [b_{ij}]$ asociada a una matriz de adyacencia $A = [a_{ij}]$ es la matriz obtenida dividiendo cada elemento de A por la suma de los elementos de la misma columna, es decir:

$$b_{ij} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}}$$

Es evidente que $0 \leq b_{ij} \leq 1$ y que los elementos de las columnas de B suman 1. Como ejemplo la matriz de probabilidad de transición asociada a (1) anterior es:

$$B = \begin{bmatrix} 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/2 & 0 & 1/3 \\ 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 1/3 & 0 \end{bmatrix} \qquad (2)$$

Esta matriz proporciona la probabilidad al ir navegando de una página a otra. Por ejemplo, si sabemos que Alice está inicialmente en la página 2, su vector de estado con 4 cifras decimales después de un "click" será:

$$\mathbf{x}^{(1)} = B\mathbf{x}^{(0)} = \begin{bmatrix} 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 1 & 0 & 0 & 1/2 & 0 & 1/3 \\ 0 & 1/3 & 0 & 0 & 0 & 1/3 \\ 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 1/2 & 1/3 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 0 \\ 0 \\ 1/3 \\ 1/3 \\ 0 \end{bmatrix} \approx \begin{bmatrix} 0.3333 \\ 0 \\ 0 \\ 0.3333 \\ 0.3333 \\ 0 \end{bmatrix}$$

Y sus vectores de estado tras sucesivos "clicks" formarán la secuencia:

$$\boldsymbol{x}^{(k)} = B\boldsymbol{x}^{(k-1)}, \quad k = 1, 2, 3, \ldots \qquad (3)$$

Se sigue que los sucesivos vectores de estado con 4 cifras decimales serán:

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}^{(1)} = \begin{bmatrix} 0.3333 \\ 0 \\ 0 \\ 0.3333 \\ 0.3333 \\ 0 \end{bmatrix}, \quad \mathbf{x}^{(2)} = \begin{bmatrix} 0.1111 \\ 0.1111 \\ 0.5000 \\ 0 \\ 0 \\ 0.2778 \end{bmatrix}, \quad \mathbf{x}^{(3)} = \begin{bmatrix} 0.1296 \\ 0.1667 \\ 0.2037 \\ 0.1296 \\ 0.2037 \\ 0.1667 \end{bmatrix},$$

$$\mathbf{x}^{(5)} = \begin{bmatrix} 0.1533 \\ 0.1245 \\ 0.3014 \\ 0.1121 \\ 0.1286 \\ 0.1800 \end{bmatrix}, \quad \mathbf{x}^{(10)} = \begin{bmatrix} 0.1562 \\ 0.1366 \\ 0.2700 \\ 0.1101 \\ 0.1366 \\ 0.1905 \end{bmatrix}, \quad \mathbf{x}^{(15)} = \begin{bmatrix} 0.1544 \\ 0.1365 \\ 0.2727 \\ 0.1090 \\ 0.1365 \\ 0.1910 \end{bmatrix}$$

1.3 Autovalores de la matriz de transición

Si suponemos que el vector de estado $\boldsymbol{x}^{(k)}$ tiende hacia un límite $\boldsymbol{x}$ cuando k (número de "clicks") aumenta indefinidamente, de (3) se deduce que

$$\boldsymbol{x} = B\boldsymbol{x}$$

Es decir, $\boldsymbol{x}$ es un autovector de B correspondiente al autovalor 1. Si reducimos el vector $\boldsymbol{x}$ de forma que sus elementos sumen 1, los elementos de $\boldsymbol{x}$ se pueden interpretar como la fracción de veces que se espera que cada página sea visitada después de infinitos "clicks". Por ejemplo, con ayuda de un CAS[2] se obtiene que para la matriz B de (2) ese autovector es:

$$\mathbf{x} = \frac{1}{110}\begin{bmatrix}17\\15\\30\\12\\15\\21\end{bmatrix} \approx \begin{bmatrix}0.1545\\0.1364\\0.2727\\0.1091\\0.1364\\0.1909\end{bmatrix}$$

Compara este resultado con el de la Tabla 2.

1.4 Estrategia de Bob

Aunque Bob coincide con Alice en la definición de clasificación de páginas, se da cuenta de que puede resultar engañoso para algunos grafos web. Por ejemplo, en la Figura 2 (a) el grafo web consiste en dos grupos de páginas no enlazados.

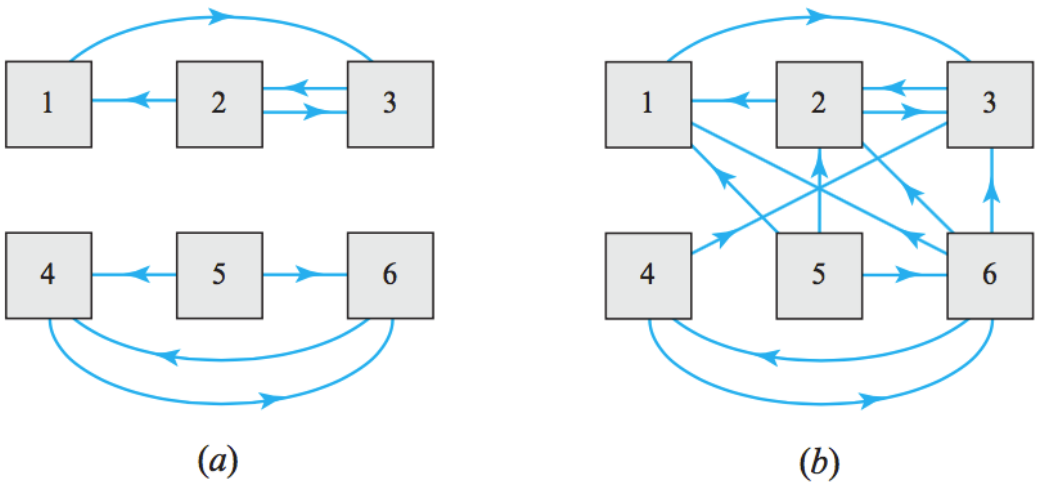


Figura 2

En este caso si un vector de estado inicial tiene probabilidad cero de enlazar a todas las páginas de uno de los grupos, lo mismo pasará en los siguientes vectores de estado, de modo que nunca se accederá a ninguna de las páginas de ese grupo. La Figura 2

[2] *CAS = Computer Algebra System (como Matlab)*

(b) representa un ejemplo más sutil. En este caso el grupo de páginas 1, 2 y 3 no tiene enlaces hacia el grupo de las páginas 4, 5, 6 de modo que cuando se abandona el grupo 4, 5, 6 se queda "atrapado" en el grupo 1, 2, 3 de modo que el conteo de acceso a las páginas 4, 5 y 6 tenderá a cero por lo que la clasificación de páginas en ese grupo será cero.

La solución de Bob a este problema es asumir que no esta obligado a seguir los enlaces que aparecen en la página en que se encuentra, pudiendo elegir, con una probabilidad determinada, cualquier página de la red. Concretamente, Bob supone que existe una probabilidad δ, denominada *factor de atenuación*, de que salte a una página referenciada en la que se encuentra y una probabilidad de $1-\delta$ de elegir la siguiente página aleatoriamente. Si hay n páginas en la red, en el último caso, la probabilidad de elegir una página aleatoriamente es $\frac{1-\delta}{n}$

Para poner en práctica su estrategia, Bob crea una nueva matriz de probabilidad de transición $M = [m_{ij}]$ en la que:

$$m_{ij} = \delta b_{ij} + \frac{1-\delta}{n} \qquad (4)$$

con b_{ij} según la definición 2. Entonces reemplaza (3) con el esquema iterativo:

$$\boldsymbol{x}^{(k)} = M\boldsymbol{x}^{(k-1)}, \quad k = 1, 2, 3, \ldots \qquad (5)$$

Por tanto, la matriz M tiene la forma:

$$M = \delta B + \frac{1-\delta}{n}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \qquad (6)$$

De lo que se sigue que el esquema iterativo de (5) se puede escribir de la forma:

$$\boldsymbol{x}^{(k)} = \delta B\boldsymbol{x}^{(k-1)} + \frac{1-\delta}{n}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad (7)$$

Como ejemplo, consideremos el grafo de web de la Figura 2 (b), en este caso su matriz de adyacencia y su matriz de transición son:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 1/3 & 1/4 \\ 0 & 0 & 1 & 0 & 1/3 & 1/4 \\ 1 & 1/2 & 0 & 1/2 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/3 & 0 \end{bmatrix}$$

Por lo general elegiremos un vector de estado inicial en el que todos sus elementos sean iguales (misma probabilidad de todas las páginas):

$$\boldsymbol{x}^{(0)} = \begin{bmatrix} 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \\ 1/6 \end{bmatrix}$$

La estrategia iterativa de Alice $\boldsymbol{x}^{(k+1)} = B\boldsymbol{x}^{(k)}$ resulta:

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \end{bmatrix}, \quad \mathbf{x}^{(5)} = \begin{bmatrix} 0.1999 \\ 0.4043 \\ 0.3930 \\ 0.0007 \\ 0.0000 \\ 0.0022 \end{bmatrix}, \quad \mathbf{x}^{(10)} = \begin{bmatrix} 0.1992 \\ 0.4015 \\ 0.3992 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}, \quad \mathbf{x}^{(15)} = \begin{bmatrix} 0.1998 \\ 0.4002 \\ 0.4000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix}$$

En comparación, cuando Bob implemente su esquema iterativo modificado empezando con el mismo vector de estado inicial, pero con $\delta = 0{,}85$ obtendrá:

$$\mathbf{x}^{(0)} = \begin{bmatrix} 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \\ 0.1667 \end{bmatrix}, \quad \mathbf{x}^{(5)} = \begin{bmatrix} 0.1891 \\ 0.3480 \\ 0.3550 \\ 0.0352 \\ 0.0250 \\ 0.0477 \end{bmatrix}, \quad \mathbf{x}^{(10)} = \begin{bmatrix} 0.1890 \\ 0.3464 \\ 0.3576 \\ 0.0350 \\ 0.0250 \\ 0.0469 \end{bmatrix}, \quad \mathbf{x}^{(15)} = \begin{bmatrix} 0.1892 \\ 0.3462 \\ 0.3578 \\ 0.0350 \\ 0.0250 \\ 0.0469 \end{bmatrix}$$

A partir de estos resultados observamos que mientras que en la estrategia de Alice las páginas 4, 5 y 6 obtienen cero, en la de Bob se obtienen resultados no nulos más razonables.

Matemáticamente, incluir un factor de atenuación de δ $(0 \leq \delta \leq 1)$ asegura que la matriz M sea regular y que para cualquier vector de estado inicial normalizado las iteraciones $\boldsymbol{x}^{(k)}$ convergen a un autovector de componentes positivas que corresponde al autovalor 1.

1.5 El algoritmo “Page Rank” de Google

Hemos calculado el pageRank al estilo de Google para nuestro internet. Google es una máquina que hace justamente eso, visitan páginas y estudian la conectividad de unas con otras y le asignan pesos de importancia en la red, el denominado *pageRank*.

Aunque la teoría de cadenas de Markov había sido empleada anteriormente para clasificar nodos de redes, la introducción del factor de atenuación representó la mayor innovación del algoritmo PageRank empleado por el motor de búsqueda de Google. Este algoritmo toma su nombre de Larry Page quien, junto con Sergey Brin, fundó la compañía Google a finales de los 90.

Además, Google tiene herramientas secretas. No se conoce como asigna las probabilidades de visita asociadas a cada navegante que nosotros hemos supuesto que

eran (1/número de páginas) para tener en cuenta cuando un navegante no sigue los enlaces de la página que está visitando.

Por supuesto, en Google pueden tocar los números asociados con el pageRank, por ejemplo, si pagas, para posicionarte mejor en las búsquedas.

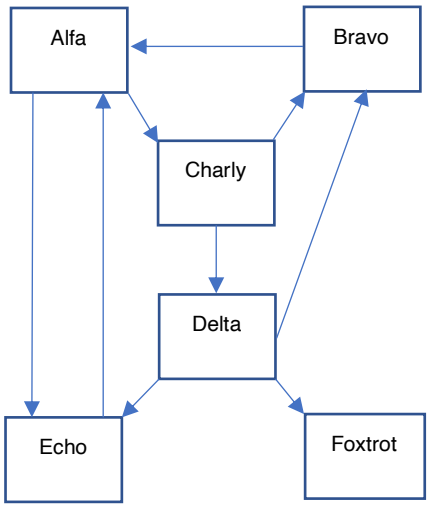
Engañar a Google cada vez es más difícil, no es buena idea poner muchos enlaces a la misma página desde la tuya porque entonces nos penalizan. Así acaban con el *linkspam*. Ahora tienen en cuenta tus búsquedas anteriores y tus visitas para darte un resultado de búsqueda personalizado de modo que si dos personas hacen la misma búsqueda conectados a sus cuentas de Google obtienen diferentes resultados.

2 EJERCICIO PRÁCTICO

El propósito de esta práctica es trabajar con grafos dirigidos y aplicar la teoría vista anteriormente para clasificar un numero de páginas (o urls) de internet de acuerdo con dicho grafo.

Ejercicio

Imaginemos un conjunto (muy reducido) de urls que está representado en el siguiente grafo:



El objetivo es clasificar las distintas urls de acuerdo con su importancia primero aplicando la teoría descrita anteriormente y después mediante el algoritmo que Matlab nos proporciona para ello.

a) (1,5 puntos) Calcula la matriz de adyacencia (unos y ceros) del grafo anterior colocando las urls en orden alfabético (es decir, la primera será “Alfa”, la segunda “Bravo”, etc), a dicha matriz la llamaremos A. Si alguno de los nodos (urls) no tiene salidas (es decir, si al navegar llegamos a esa página, nos quedaríamos bloqueados, evítalo añadiendo un enlace a todas las demás. Para mostrar por pantalla la matriz emplea la secuencia de comandos:

```
disp('la matriz de adyacencia es:')
c = array2table(A);
disp(c)
```

b) (1,5 puntos) Calcula, mediante los comandos de Matlab, la matriz de probabilidad de transición que llamaremos B.

c) (2 puntos) Suponiendo que el factor de atenuación es $\delta = 85\%$, calcula la nueva matriz de probabilidad de transición que llamaremos M. Comprueba que las columnas de M suman 1. Comprueba que 1 es el autovalor principal de M mediante el comando `eig` de Matlab

d) (2 puntos) Calcula el autovector que corresponde al autovalor 1 que llamaremos $\boldsymbol{x}$. Obtén, a partir de $\boldsymbol{x}$ la clasificación de las páginas. Ten en cuenta que el vector $\boldsymbol{x}$ es la solución del sistema homogéneo $(M - I)\boldsymbol{x} = 0$ y que el comando `null` calcula dicha solución. El vector x muestra la clasificación de las páginas, pero debes darla por orden, indicando el nombre de cada url. Para ello crea un vector de caracteres con los nombres de las url: `url = {'Alfa', 'Bravo', 'Charly','Delta', 'Echo', 'Foxtrot'};` de esa forma podrás indexar las posiciones de acuerdo con el vector $\boldsymbol{x}$ mediante bucles `for-if`, no es fácil, dedícale un tiempo a resolverlo.

e) (2 puntos) Dibuja el grafo dirigido poniendo los nombres en cada nodo mediante el comando `digraph(A,{'Alfa','Beta',…})` de Matlab (recuerda la nota 1 del fundamento teórico de esta práctica y ten en cuenta que el comando `digraph` representa la traspuesta de la matriz de adyacencia). Debes representar el verdadero grafo (recuerda que en el apartado a) has modificado el grafo en los nodos que no tenían salidas)

f) (1 punto) Verifica que el comando `centrality` de Matlab da como resultado la misma ordenación calculada en el apartado d). Lee la ayuda de Matlab de este comando para añadir el tipo (`type`) adecuado. ¿El vector obtenido mediante el algoritmo `centrality` de Matlab es el mismo que el obtenido en el apartado d)? ¿Por qué? ¿Hay alguna relación entre ellos? Estudia cómo opera el algoritmo del comando de Matlab y con qué tolerancia por defecto calcula.

Tarea

Crea un script de Matlab que responda a las cuestiones y sube el archivo "Practica5Grupoxx.m" que hayas empleado para resolver las preguntas. Sigue las indicaciones de formato de archivo del curso

Lecturas adicionales de ayuda

La web de MatWorks dedica un apartado a el algoritmo PageRank, puedes consultarlo aquí: https://es.mathworks.com/help/matlab/math/use-page-rank-algorithm-to-rank-websites.html