Práctica 5

Representación del Conocimiento

Práctica de Probabilidad

Autores: Álvaro López García,
Jairo González Gómez,
Nicolás Rodrigo Pérez.

Fecha: 6 de Enero de 2022

"Pagerank"

Pagerank es un algoritmo para puntuar la importancia de una página web utilizando un modelo probabilístico del comportamiento de los usuarios en la web. Primero se modelan las páginas web conocidas mediante un grafo dirigido donde una página web esta unida a otra si hay un enlace dentro de esa página.

El usuario navega por las páginas web aleatoriamente, saltando de una página web a otra en cada unidad de tiempo eligiendo aleatoriamente uno de los enlaces. Además, cada enlace tiene una probabilidad de ser elegido.

En el siguiente ejemplo, vemos una representación gráfica de tres páginas web con sus enlaces.

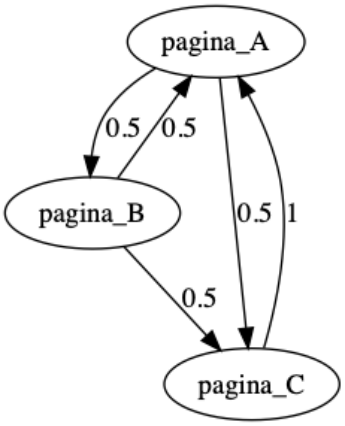

Figura 1: Ejemplo de representación de enlaces entre páginas web.

La distribución de los usuarios esta dada por una distribución de probabilidad que depende del tiempo.

Supongamos que en tiempo cero, la distribución de probabilidad de estar en cada una de las páginas es (0.1, 0.2, 0.7). Según el modelo matemático, denotando la probabilidad de estar en la pagina A en el momento t como $P(A(t))$, tenemos que

$$
\begin{aligned}
P(A(t)) &= P(A(t)|B(t-1))P(B(t-1)) + P(A(t)|C(t-1))P(C(t-1)),\\
P(B(t)) &= P(B(t)|A(t-1))P(A(t-1)) + P(B(t)|C(t-1))P(C(t-1)),\\
P(C(t)) &= P(C(t)|A(t-1))P(A(t-1)) + P(C(t)|B(t-1))P(B(t-1)).
\end{aligned}
$$

El siguiente código demuestra como calcular las diferentes iteraciones para el ejemplo anterior.

```
import numpy as np

M = np.array([[0,0.5,0.5],
              [0.5,0,0.5],
              [1,0,0]
])

v = np.array([0.1,0.1,0.8])
v= np.matmul(v,M)
print(v)
```

```
[0.85 0.05 0.1]
```

Modificar el código anterior para calcular que pasará después de que pasen suficientes unidades de tiempo. ¿A qué tienden las probabilidades?

```
import numpy as np

```

```
M = np.array([[0,0.5,0.5],
              [0.5,0,0.5],
              [1,0,0]
])

v = np.array([0.1,0.2,0.7])
print np.matmul(v,M)
```

Para ilustrar esto hemos desarrollado un script[1] en `Python` que nos mostrará la evolución de los valores de probabilidad de las distintas páginas y nos devolverá esta imagen en un archivo (‘‘`out.png`’’). Este script se basa en el código proporcionado y nos servirá para ilustrar este fenómeno a lo largo de toda la práctica.

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

MAX_ITERS = 15

# Matriz de adyacencia con pesos entre las distintas páginas web
M = np.array([[0, 0.5, 0.5],
    [0.5, 0, 0.5],
    [1, 0, 0]
])

# Distribucion inicial de probabilidades
v = np.array([0.1,0.2,0.7])

def nextIter(v, M):
    return v @ M

# Array con la evolución de los valores de las probabilidades
probs = v.copy().T

# Cálculo de la evolucion de las probabilidades
for i in range(MAX_ITERS):
    v = nextIter(v, M)
    probs = np.vstack((probs, v.T))

# Ploteo de la evolución de las probabilidades
plt.title("Evolución de las probabilidades")

for i in range(probs.shape[1]):
    print(probs[:,i])
    plt.plot(probs[:,i], label="Pagina " + chr(i + 65))

plt.xlabel('Valores de t')
plt.ylabel('Valor de P(X(t))')
plt.legend()
plt.savefig("out.png")
plt.show()
```

Y a continuación mostramos el resultado de una ejecución:

[1] Archivo `Pregunta1.py` adjunto a la entrega

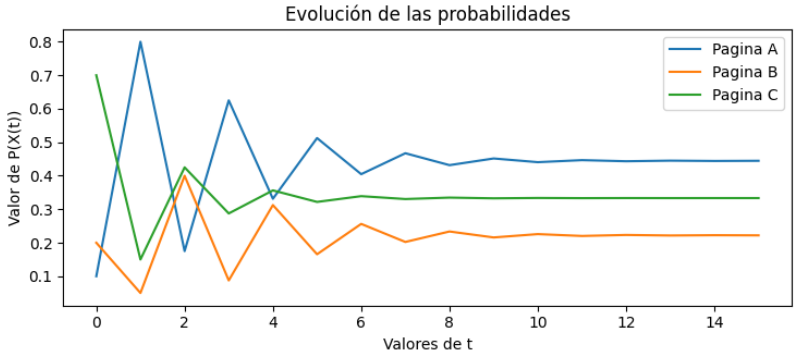

Figura 2: Resultados de la ejecución para los valores proporcionados.

Como podemos ver en la Figura 2, los valores de probabilidad tienden a estabilizarse tras un número determinado de iteraciones (que nosotros hemos fijado arbitrariamente a 15), y dichos valores los interpretamos como la "importancia" de una página.

Estas probabilidades miden la importancia de las páginas web y permiten establecer su importancia.

1) ***¿Qué pasa si se cambia la distribución inicial de los usuarios? Haced una prueba con una distribución diferente y comprobad si se tienden a distribuir igual después de varias iteraciones.***

   Los valores tenderán a ser los mismos independientemente de la distribución inicial de los usuarios, por lo que los pesos de las aristas del grafo serán los que dominen los valores de convergencia de las probabilidades.

   Ejemplo de evolución de las probabilidades con valores `[0.5,0.3,0.2]`:

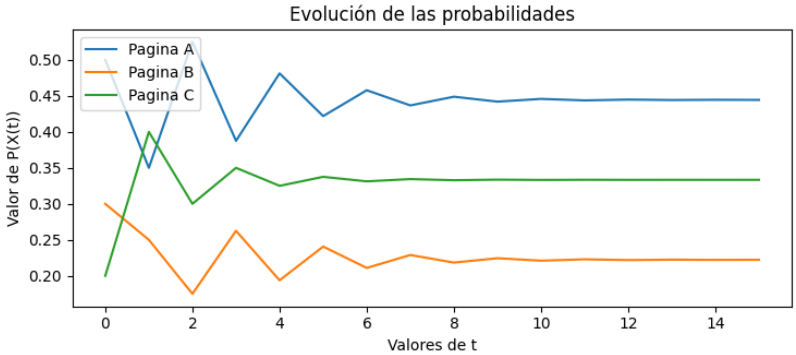

Figura 3: Resultados de la ejecución para los valores proporcionados.

   En PageRank los enlaces de las páginas web tienen una distribución uniforme respecto del evento de ser elegidos por el usuario, es decir se suele dar a cada enlace de la página la probabilidad de uno dividido entre el número total de enlaces.

   Si una página no tiene enlaces, se supone que los usuarios se quedan indefinidamente en la página.

2) ***Calculad la importancia de cada una de estás páginas, independientemente de la distribución de los usuarios.***

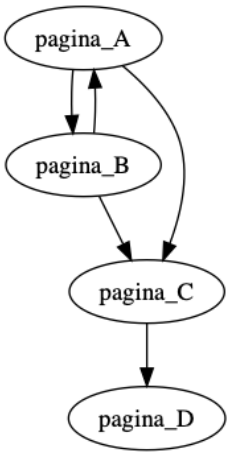

Figura 4: Otro ejemplo de representación de enlaces entre páginas web.

Según lo descrito en el enunciado, todo enlace ha de ser igual de probable de ser escogido por el usuario en cualquier momento dado. De tal forma que los enlaces salientes de una página web han de tener todos la misma probabilidad. Y por tanto, nuestra matriz de adyacencia ha de cumplir la propiedad de que todas sus filas sumen 1 (la probabilidad de que el usuario salga de esa página, que siempre va a darse según la especificación del problema).

Hemos de mencionar también que siendo consecuentes con lo dicho en el enunciado de que cuando una página no tiene enlaces salientes, el usuario ha de quedarse en ella, hemos de añadir un nuevo enlace en el grafo propuesto de D a D. Pues de lo contrario, todo aquel usuario que navegase hasta llegar a la página D, "desaparecería" en la siguiente iteración del algoritmo.

Por lo tanto el grafo quedaría de la siguiente forma:

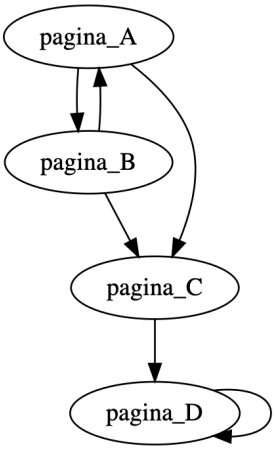

Figura 5: Grafo modificado según lo descrito.

Y de acuerdo con lo mencionado, podremos establecer la importancia de una página (que denotaremos como $I(X)$, donde X es la página de la que vamos a calcular la importancia) de la siguiente forma:

$$I(X) = \sum \frac{I(Z)}{\text{nº total de enlaces salientes de } Z}$$

donde Z representa cada una de las páginas que tienen un enlace que apunta a X.

Aplicando esa expresión a las distintas páginas propuestas en la figura 5, podemos establecer las siguientes relaciones de dependencia entre las importancias de las páginas:

- Página A: $I(A) = \frac{I(B)}{2}$
- Página B: $I(B) = \frac{I(A)}{2}$

- Página C: $I(C) = \frac{I(A)}{2} + \frac{I(B)}{2}$
- Página D: $I(D) = \frac{I(C)}{1} + \frac{I(D)}{1}$

De este modo establecemos relaciones de dependencia entre las distintas importancias de las páginas sin que estas dependan del tiempo ni de ninguna distribución inicial de usuarios. Pero con el fin de hallar los valores concretos de cada una, hemos de añadir una condición mas, que el total de las sumas de las distintas importancias de las páginas sea 1. Pues enfocándolo de nuevo desde el punto de vista probabilista, esto representaría la probabilidad de que un usuario esté en una página, que ha de ser 1 pues esto siempre se va a dar.

Y de acuerdo a esto podemos formar el siguiente sistema de ecuaciones:

$$\begin{cases} I(A) = \frac{I(B)}{2} \\ I(B) = \frac{I(A)}{2} \\ I(C) = \frac{I(A)}{2} + \frac{I(B)}{2} \\ I(D) = I(C) + I(D) \\ I(A) + I(B) + I(C) + I(D) = 1 \end{cases}$$

Y resolviendo el sistema de ecuaciones nos queda que: $I(A) = I(B) = I(C) = 0$, $I(D) = 1$.

Con el fin de comprobar empíricamente que nuestros resultados son correctos, hemos introducido los valores de adyacencia de la figura 5 y una distribución de probabilidad cualquiera en el script que hemos mencionado anteriormente, que nos plotea la evolución de las probabilidades en función del tiempo. Y hemos obtenido la siguiente gráfica:

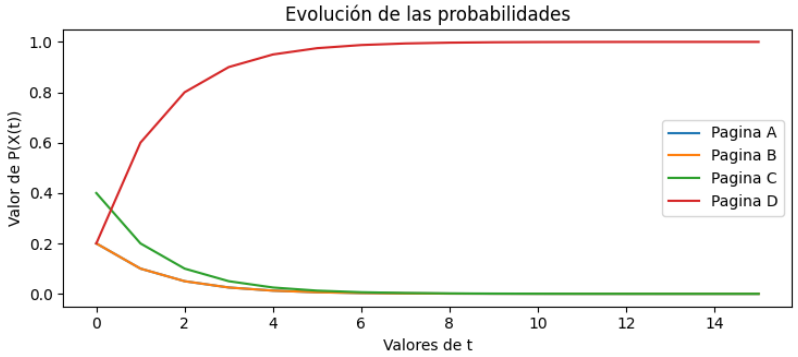

Figura 6: Evolución en el tiempo de las probabilidades de las páginas de la figura 5

Por lo que hemos comprobado que nuestros resultados, obtenidos sin depender de una distribución inicial de usuarios ni del tiempo son correctos.

En los ejemplos anteriores, independientemente de la distribución inicial de los usuarios, la distribución final de los usuarios cuando el tiempo tiende a infinito es la misma pero esto no siempre es así.

3) ***Poned ejemplos varias páginas web, que dependiendo de la distribución inicial de los usuarios den resultados distintos para la importancia de la página y otras páginas web donde no exista ese límite.***

Nota: Estos ejemplos tienen que estar escritos en lenguaje dot y se debe entregar conjuntamente la imagen en formato png.

Nos hemos dado cuenta de que este fenómeno se da en aquellos grafos ciclo, con una distribución inicial de usuarios no uniforme[2].

[2] Archivo `ejemplo3.png` adjunto a la entrega.

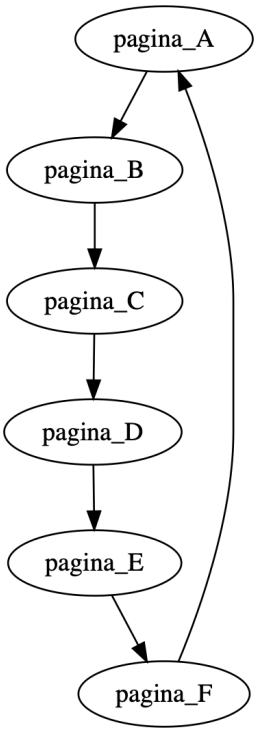

Figura 7: Grafo ciclo para el caso de 6 nodos.

En un caso "estándar" las aristas del grafo harían que estadísticamente se determinase el flujo de usuarios por las páginas hacia una mayoritariamente. Aquí esto no sucede, pues todos los nodos tienen el mismo grado. Y por tanto, conceptualmente lo que sucederá será que si la distribución inicial de usuarios no es uniforme, habrá una masa de usuarios mas grande que el resto, que estará viajando por el ciclo hasta el infinito.

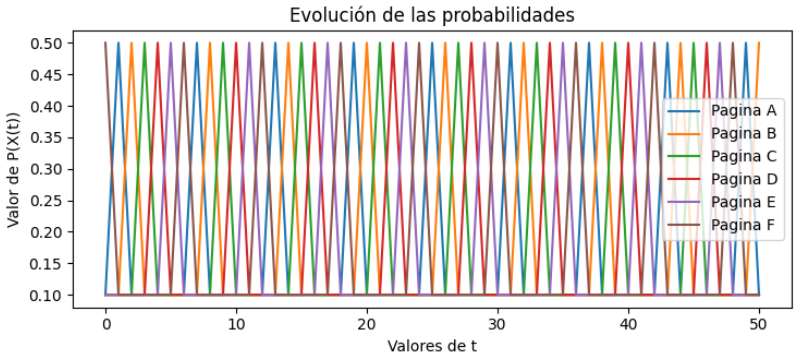

Figura 8: Ejecución de la figura 7 durante 100 iteraciones.

Podemos ver como esta gráfica ilustra lo que mencionábamos anteriormente, pues en cada iteración se produce un pico en la página "en la que se encuentra" esa masa mayoritaria de usuarios. Y dado que las aristas del grafo no determinan un camino mayoritario, estos valores nunca llegan a converger.

Una forma de evitar estos problemas es definir una constante d que representa una probabilidad ir a una página cualquiera. Es decir, con una probabilidad de d el usuario elige una página al azar. En este caso, se puede demostrar que siempre se converge para cualquier de probabilidad inicial.

4) **Se pide dar la importancia de las páginas dadas en el ejercicio 3 para un valor de $d = 0,01$.**

Para la implementación de la constante d y todo lo que ello implica, hemos modificado la cabecera del código desarrollado del siguiente modo:

```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

MAX_ITERS = 100

# Probabilidad de salto aleator
d = 0.01

# Matriz de adyacencia con pesos entrer las distintas páginas web
M = np.array([[0, 1, 0, 0, 0, 0],
              [0, 0, 1, 0, 0, 0],
              [0, 0, 0, 1, 0, 0],
              [0, 0, 0, 0, 1, 0],
              [0, 0, 0, 0, 0, 1],
              [1, 0, 0, 0, 0, 0],
])

M = M + d
sums = np.sum(M, axis=1)
M_new = np.array([])

for i in range(len(sums)):
    M_new = np.hstack((M_new, (M[i,:] / sums[i])))

M = np.split(M_new, M.shape[0])

# Distribucion de probabilidad segun enlaces por pagina
v = np.array([0.1, 0.1, 0.1, 0.1, 0.1, 0.5])
...
```

Lo que hacemos en esta sección del código es inicializar todas las variables pertinentes, sumar d a todos los elementos de `M`, y normalizar los variables de cada fila de forma que se mantenga la propiedad de que todas las filas de `M` sumen 1.

Tras aplicar estas modificaciones obtenemos el siguiente output:

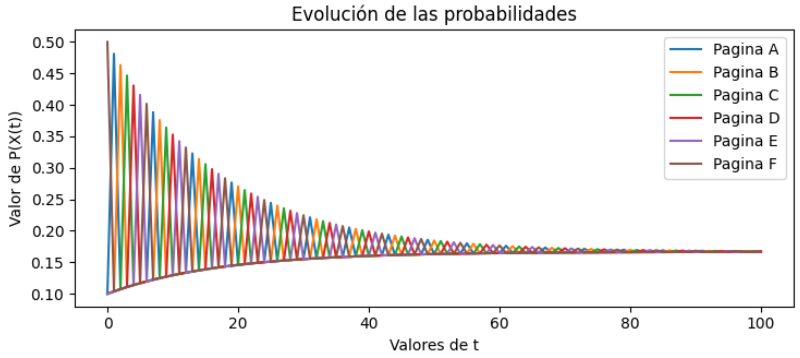

Figura 9: Ejecución de la figura 7, con la implementación de la constante d con valor $0,01$.

Podemos determinar así el valor de la importancia de todas las páginas en $0,166$. Este resultado es bastante lógico, pues si aleatoriamente los usuarios pueden saltar a una página cualquiera, estos tenderán a repartirse uniformemente (en promedio) entre todas las páginas. Por lo que la importancia de cada página se situará entorno a $1/6 \approx 0,166$.