

El Algoritmo PageRank

Álgebra aplicada

Domingo López Rodríguez

Índice de Contenidos

Introducción	3
El Algoritmo PageRank	3
Bloque I: Relación con el Álgebra Lineal	7
Bloque II: ¿Cómo hacerlo en R?	9

Introducción

A finales de la década de 1990, cuando Google entró en línea, un factor clave que lo distinguía de otros motores de búsqueda era su capacidad para ofrecer constantemente los resultados más relevantes en la parte superior de las listas de búsqueda. A diferencia de otros motores de búsqueda, donde los usuarios a menudo tenían que cribar páginas de enlaces irrelevantes que simplemente coincidían con la consulta de búsqueda, Google presentaba una experiencia más optimizada.

Esta magia reside en parte en el algoritmo PageRank de Google. Este algoritmo asigna un valor cuantitativo a cada página web, valorando esencialmente su importancia. Al aprovechar PageRank, Google puede clasificar las páginas web y priorizar las más importantes (que normalmente se correlacionan con los resultados más relevantes y útiles) en las listas de búsqueda.

Nuestro objetivo es establecer un método para asignar puntuaciones de importancia a cada página web dentro de la base de datos. Esto permitirá al sistema priorizar estas puntuaciones cuando un usuario realice una búsqueda y se identifique un subconjunto de páginas relevantes. El foco de este artículo está en el paso clave de definir y cuantificar la "importancia" de una página web dentro de la estructura interconectada de la web. Es importante tener en cuenta que si bien la calificación de importancia de una página web es un factor significativo, no es el único determinante de cómo se presentan los enlaces en los resultados de búsqueda.

Por supuesto, este método no es exclusivo de la web, sino que es aplicable a cualquier grafo dirigido ponderado.

El Algoritmo PageRank

A lo largo de esta presentación, haremos uso de un ejemplo de juguete para comprender mejor la idea y los conceptos subyacentes.

Consideremos una red de páginas web como la del gráfico inferior, todas relacionadas con un mismo término de una búsqueda.

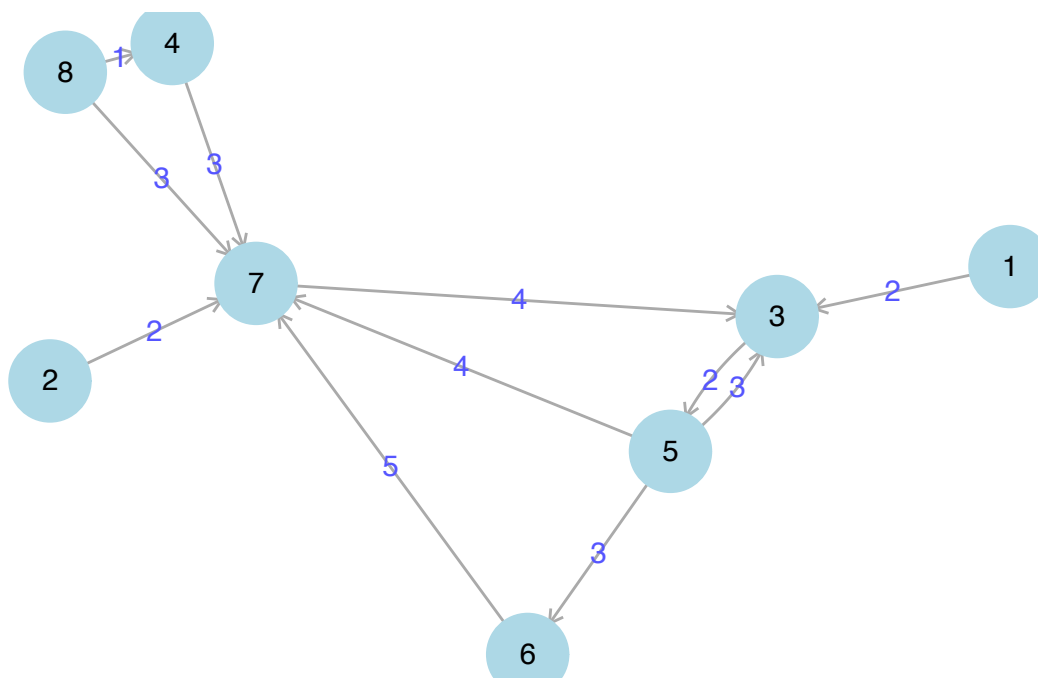


Figura 1. Grafo dirigido ponderado que representa la red de páginas web de ejemplo.

Las flechas de las aristas entre una web y otra indican la existencia de un enlace en la primera a la segunda. Por ejemplo, la web número 7 tiene cinco enlaces entrantes (desde las páginas 2, 4, 5, 6 y 8) y solo uno saliente (hacia la página 3).

El número en la arista del grafo que une las páginas i y j indica el número de visitantes de la página i que a continuación visitaron la página j . Por ejemplo, del total de personas que visitaron la página 8, una visitó después la página 4 y tres visitaron la página 7.

La idea del buscador de Google fue presentar esa lista de 8 posibles webs relacionadas con un tema común priorizada según la importancia de cada web.

¿Cómo sabemos la importancia de cada web? La idea de Google se basaba en que la importancia se propaga: cuantas más veces se pase de una web "A" a otra web "B", más *hereda* la web "B" importancia de "A". Veamos esta idea con nuestro ejemplo sencillo.

Vamos a denotar por $x_i \in \mathbb{R}^+$ la importancia de la web i -ésima. Este marco de trabajo nos dice que la importancia que tiene la web i proviene de las importancias de las webs que la enlazan.

En el ejemplo, la importancia de la web número 7, es decir, x_7 , proviene de las importancias que le

aportan las webs número 2, 4, 5, 6 y 8.

Vamos a presentar en forma de matriz las conexiones entre webs (es realmente la matriz de adyacencia *ponderada* del grafo de antes, traspuesta):

```
Warning: `as_adj()` was deprecated in igraph 2.1.0.  
i Please use `as_adjacency_matrix()` instead.
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 3 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 3 & 4 & 5 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En esta matriz, en (i, j) se almacena el número de veces que se pasa del nodo j al nodo i , es decir, cuántas veces el nodo i recibe un enlace desde el nodo j .

Vamos a normalizar esta matriz, dividiendo cada columna por su suma. Mediante esta normalización, lo que se está expresando en el elemento (i, j) es la probabilidad de transitar desde la web j -ésima a la i -ésima.

i Probabilidades de Transición

En casos reales, se suele conocer estas probabilidades a ciencia cierta, gracias al rastreo que hacen de nuestra actividad en internet. Esta matriz, en ese caso, reproduciría exactamente esas probabilidades de transición.

A esta matriz se la denomina matriz de transiciones y tiene la propiedad de ser *estocástica por columnas*, es decir, sus columnas suman uno. Esta propiedad es esencial para el desarrollo teórico del algoritmo PageRank. Una vez normalizada, la matriz de transición queda como sigue:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 3/10 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3/10 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2/5 & 1 & 0 & 3/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Teniendo en cuenta esta matriz, podemos escribir las relaciones entre las importancias (los x_i) de las distintas webs:

$$\left\{ \begin{array}{rclcl} & & & & 0 & = & x_1 \\ & & & & 0 & = & x_2 \\ x_1 & & & +3/10x_5 & +x_7 & = & x_3 \\ & & & & 1/4x_8 & = & x_4 \\ & x_3 & & & & = & x_5 \\ & & & 3/10x_5 & & = & x_6 \\ & x_2 & +x_4 & +2/5x_5 & +x_6 & +3/4x_8 & = & x_7 \\ & & & & & 0 & = & x_8 \end{array} \right.$$

Por ejemplo, la primera ecuación nos dice que la importancia de la web número 1 es 0 (ya que ninguna web enlaza a dicha página), y, mirando la ecuación correspondiente, vemos que la importancia de la web 3 proviene de la siguiente forma: toda la importancia de la web 1 más 3/10 de la importancia de la web 5, más toda la de la web 7, todo esto relacionado con los enlaces que van de unas a otras.

Si nos fijamos en la forma matricial del problema, estamos buscando un vector de importancias, $x = (x_i)$ que verifique $Mx = x$, es decir, un *punto fijo* de las importancias.

En este caso, una posible solución es:

$$x = \begin{pmatrix} 0 \\ 0 \\ 10/7 \\ 0 \\ 10/7 \\ 3/7 \\ 1 \\ 0 \end{pmatrix}$$

Es común normalizar este vector de importancias para que su suma dé uno. En este caso, nos queda:

$$x = \begin{pmatrix} 0 \\ 0 \\ 1/3 \\ 0 \\ 1/3 \\ 1/10 \\ 7/30 \\ 0 \end{pmatrix} \approx \begin{pmatrix} 0 \\ 0 \\ 0.33333 \\ 0 \\ 0.33333 \\ 0.1 \\ 0.23333 \\ 0 \end{pmatrix}$$

Como resultado, vemos que las páginas más importantes son la 3 y la 5, mientras que las 1, 2, 4 y 8 tienen importancia nula.

Bloque I: Relación con el Álgebra Lineal

Según la formulación anterior del problema, tenemos que buscar un vector $x = (x_1, x_2, \dots, x_n)^t$ tal que $Mx = x$, lo cual es equivalente a encontrar un autovector asociado al autovalor 1.

Por tanto, encontrar los vectores que lo cumplen es resolver el sistema $(M - I)x = 0$ que, en

forma matricial, se puede resolver usando Gauss-Jordan:

$$\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 3/10 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1/4 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3/10 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 2/5 & 1 & -1 & 3/4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -10/7 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -10/7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -3/7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Este último sistema se escribe como:

$$\left\{ \begin{array}{llll} x_1 & & & = 0 \\ & x_2 & & = 0 \\ & & x_3 & -10/7x_7 = 0 \\ & & & x_4 = 0 \\ & & & & x_5 & -10/7x_7 = 0 \\ & & & & & x_6 & -3/7x_7 = 0 \\ & & & & & & x_8 = 0 \\ & & & & & & & 0 = 0 \end{array} \right.$$

Usando este último sistema, podemos obtener las ecuaciones paramétricas de su conjunto solución (el subespacio asociado al autovalor 1) y su base:

$$U = \left\{ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} \in \mathbb{R}^8 \mid \begin{array}{l} x_1 = 0 \\ x_2 = 0 \\ x_3 = 10/7\alpha \\ x_4 = 0 \\ x_5 = 10/7\alpha \\ x_6 = 3/7\alpha \\ x_7 = \alpha \\ x_8 = 0 \end{array}, \alpha \in \mathbb{R} \right\} = \mathcal{L} \left\{ \begin{pmatrix} 0 \\ 0 \\ 10/7 \\ 0 \\ 10/7 \\ 3/7 \\ 1 \\ 0 \end{pmatrix} \right\}$$

Podemos observar por tanto que llegamos a la solución propuesta en el apartado anterior. Es decir, hemos llegado a un subespacio que *determina* la importancia de cada nodo dentro del grafo. Tomamos entonces un vector cuya suma de las componentes sea 1 dentro de ese subespacio y tenemos las puntuaciones de importancia que queríamos.

i ¿Pero seguro que el 1 es autovalor?

Bajo condiciones suficientemente generales en la práctica, podemos asegurar que 1 es autovalor. Es fundamental que la suma por columnas sea 1.

Más aún: $\lambda = 1$ es el mayor autovalor de la matriz M .

i ¿Y si el subespacio asociado tiene dimensión mayor que 1?

No podríamos obtener un único *autovector* asociado a $\lambda = 1$.

Existe una estrategia (la que implementa Google), que permite asegurar que la dimensión del subespacio asociado al autovalor 1 es uno.

Bloque II: ¿Cómo hacerlo en R?

Una vez definida la matriz M de transiciones, es necesario, como hemos comentado antes, normalizar para que las columnas sumen 1. A modo de ayuda, copio aquí un código que puede ayudar a realizar esta normalización:

```
M <- scale(M,  
  center = FALSE,  
  scale = colSums(M))
```

Una vez que tenemos esta matriz de transición, existen múltiples librerías que permiten calcular valores y vectores propios. De hecho, el propio R base tiene la función `eigen()` que calcula tanto los autovalores como los autovectores de una matriz dada (teniendo en cuenta que, por defecto, va a trabajar en $\mathbb{C}$).

Internamente, utilizan técnicas numéricas para los cálculos. Es decir, **no** resuelven los sistema de ecuaciones ni la ecuación característica.

El método más usado se denomina **método de la potencia**, que, partiendo de un vector v_0 , calcula la siguiente sucesión de vectores:

$$v_1 = \frac{Mv_0}{\|Mv_0\|}, v_2 = \frac{Mv_1}{\|Mv_1\|}, \dots, v_{k+1} = \frac{Mv_k}{\|Mv_k\|}$$

hasta que el cambio entre dos de estos vectores sea suficientemente pequeño, es decir, hasta que $\|v_{k+1} - v_k\| < \varepsilon$, para algún valor pequeño de ε . **Nota:** $\|v\|$ denota la norma euclídea (o *módulo*) del vector v . Se puede demostrar que este método de la potencia converge, es decir, se detiene en algún momento, y el vector resultante es un autovector asociado al valor propio de mayor valor (en valor absoluto).

i El método de la potencia en producción

Este método se usa todavía en aplicaciones de alto coste computacional, con grandes matrices (millones de filas y columnas, o incluso más), donde solo se necesite el autovalor *dominante* y su autovector asociado. Concretamente, es el método usado en el algoritmo PageRank de Google.

En R, este método se encuentra implementado en la librería `matlib`, en la función `powerMethod`. Si lo usamos con la matriz M ya normalizada, tenemos:

```
library(matlib)
resultado <- powerMethod(M)
```

Podemos ver el autovalor determinado:

```
resultado$value
```

```
[1] 1
```

y el autovector correspondiente:

```
resultado$vector
```

```
      [,1]
1 0.0000000
2 0.0000000
3 0.6225730
4 0.0000000
5 0.6225725
6 0.1867718
7 0.4358011
8 0.0000000
```

En nuestro caso, como siempre, queremos un vector cuya componentes sumen 1, así que hacemos lo siguiente:

```
resultado$vector / sum(resultado$vector)
```

```
      [,1]
1 0.0000000
2 0.0000000
3 0.33333345
4 0.00000000
5 0.33333317
```

6 0.09999996

7 0.23333342

8 0.00000000

que es prácticamente igual al vector de importancias x que hallamos en la [sección del algoritmo PageRank](#).