El Algoritmo PageRank

Junio 2020

Al pensar en cómo debería funcionar un motor de búsqueda como Google, a primera vista parece razonable imaginar que todo lo que hace el algoritmo es mantener indexadas a todas las páginas Web, y cuando el usuario inserta un comando de búsqueda, el algoritmo navega a través del índice y cuenta las ocurrencias de las palabras a buscar. Luego, los primeros resultados son las páginas con el mayor número de ocurrencias de las palabras de interés.

Este era el método considerado correcto en los años 90, cuando los motores de búsqueda usaban *sistemas de clasificación basados en texto*. Sin embargo, este enfoque conlleva algunos problemas. Por ejemplo, la búsqueda de un término común como "Internet" fue problemática. La primera página mostrada por un motor de búsqueda de este estilo estaba escrita en chino, con repetidas ocurrencias de la palabra, sin contener información adicional. Como otro ejemplo, supongamos que se quiere buscar información sobre la Universidad de Cornell. Luego, se busca la palabra "Cornell", y se espera que "www.cornell.edu" sea el primer resultado. Sin embargo, al contar el número de ocurrencias en las páginas de la palabra buscada, este puede no ser el caso: ¿qué pasa si alguien decide diseñar una página Web con la palabra "Cornell" un millón de veces? No tendría sentido que esa página sea el primer resultado en la búsqueda.

La utilidad de un motor de búsqueda depende de la *relevancia* del conjunto de resultados que retorna. Podría haber un millón de páginas Web que incluyan una palabra o frase concreta, pero inevitablemente algunas de ellas serán más relevantes, populares o fidedignas. Un usuario no posee la habilidad de revisar todas las páginas que contienen las palabras de interés. Por esta razón, lo esperable es que las páginas relevantes se encuentren entre las primeras 20-30 páginas retornadas.

Los motores de búsqueda modernos utilizan otros métodos de clasificación de resultados para proveer las páginas más relevantes en una búsqueda dada. Uno de los más conocidos e influyentes algoritmos para calcular la relevancia de páginas Web es el Algoritmo Page Rank usado por Google. Fue inventado por Larry Page y Sergey Brin mientras realizaban sus estudios en Stanford, y se convirtió en una patente de Google en 1998. La idea sobre la que se centra este algoritmo es que la importancia de cualquier página puede ser determinada al mirar las páginas que poseen vínculos (links) a ella. Si creamos una página Web i e incluimos un vínculo a la página j, esto significa que consideramos a j como importante y relevante

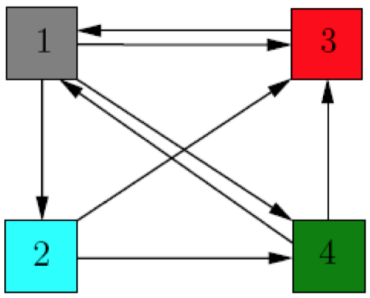
para el tópico particular de nuestra página. Por otro lado, si solo una página tiene vínculos a la página j, digamos, la página k, pero k es una página relevante, podemos decir que k afirma que j es relevante. Independientemente de si hablamos de popularidad o confianza, podemos asignar iterativamente un rango a cada página basado en los rangos de las páginas que apuntan a ella.

Con este objetivo, comenzamos visualizando la Web como un grafo dirigido, donde los nodos representan páginas Web y las aristas representan los vínculos entre ellas.

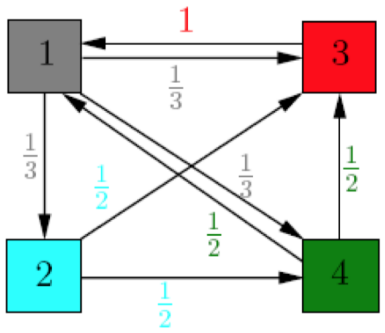
Supongamos, por ejemplo, que nuestra Internet consiste de cuatro páginas: página 1, 2, 3 y 4, referidas entre sí de la siguiente manera:

- La página 1 tiene vínculos a las páginas 2, 3, y 4.
- La página 2 tiene vínculos a las páginas 3, y 4.
- La página 3 tiene vínculos a la página 1.
- La página 4 tiene vínculos a las páginas 1 y 3.

Podemos plasmar la información en un grafo dirigido con cuatro nodos, uno por cada página. Cuando la página i refiere a la página j, añadimos una arista dirigida del nodo i al j. Naturalmente, con el objetivo de calcular el rango de cada página, se ignoran links de navegación como los botones "Atrás" y "Adelante", pues sólo nos importa conocer las conexiones entre diferentes páginas. Luego de analizar las páginas y sus conexiones, tenemos el siguiente grafo:



En nuestro modelo, cada página debería transferir equitativamente su importancia a las páginas a las que refiere. El nodo 1 tiene 3 aristas salientes, por lo que pasará $\frac{1}{3}$ de su importancia a cada uno de esos nodos. El nodo 3 tiene solo una arista saliente, por lo que pasará toda su importancia al nodo 1. En general, si un nodo tiene k aristas salientes, pasará $\frac{1}{k}$ de su importancia a cada uno de los nodos a los que refiere. Visualizamos mejor este proceso mediante la asignación de un peso a cada arista.



Llamemos A a la matriz de transición del grafo. Es decir, $A = \begin{bmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{bmatrix}$.

Enfoque desde Sistemas Dinámicos

Supongamos que inicialmente la importancia está uniformemente distribuida entre los 4 nodos, es decir, cada uno tiene $\frac{1}{4}$. Llamemos v al vector de rango inicial, con todas sus entradas iguales a $\frac{1}{4}$. Cada link entrante aumenta la importancia de una página, por lo que, como primer paso, actualizamos el rango de cada página sumando al valor actual la importancia de los vínculos entrantes. Esto es equivalente a multiplicar la matriz A con v. Entonces, en el primer paso, el nuevo vector de importancia es $v_! = Av$. Repitiendo el proceso, en el paso dos, el vector de importancia es $v_2 = A(Av) = A^2v$. Luego, mediante cálculo computacional, tenemos:

$$v = \begin{pmatrix} 0{,}25 \\ 0{,}25 \\ 0{,}25 \\ 0{,}25 \end{pmatrix}, \quad Av = \begin{pmatrix} 0{,}37 \\ 0{,}08 \\ 0{,}33 \\ 0{,}20 \end{pmatrix}, \quad A^2v = \begin{pmatrix} 0{,}43 \\ 0{,}12 \\ 0{,}27 \\ 0{,}16 \end{pmatrix},$$

$$A^3v = \begin{pmatrix} 0{,}35 \\ 0{,}14 \\ 0{,}29 \\ 0{,}20 \end{pmatrix}, \quad A^4v = \begin{pmatrix} 0{,}39 \\ 0{,}11 \\ 0{,}29 \\ 0{,}19 \end{pmatrix}, \quad A^5v = \begin{pmatrix} 0{,}39 \\ 0{,}13 \\ 0{,}28 \\ 0{,}19 \end{pmatrix},$$

$$A^6v = \begin{pmatrix} 0{,}38 \\ 0{,}13 \\ 0{,}29 \\ 0{,}19 \end{pmatrix}, \quad A^7v = \begin{pmatrix} 0{,}38 \\ 0{,}12 \\ 0{,}29 \\ 0{,}19 \end{pmatrix}, \quad A^8v = \begin{pmatrix} 0{,}38 \\ 0{,}12 \\ 0{,}29 \\ 0{,}19 \end{pmatrix}.$$

Nótese que las iteraciones $v, Av, \ldots, A^k v$ tienden al valor de equilibrio $v^* = \begin{pmatrix} 0{,}38 \\ 0{,}12 \\ 0{,}29 \\ 0{,}19 \end{pmatrix}$. Llamamos a este el vector PageRank de nuestro grafo.

Enfoque desde el Álgebra Lineal

Llamemos x_1, x_2, x_3 y x_4 a la importancia de las cuatro páginas. Analizando la situación de cada nodo, tenemos el siguiente sistema:

$$\begin{cases} x_1 = x_3 + \frac{1}{2}x_4 \\ x_2 = \frac{1}{3}x_1 \\ x_3 = \frac{1}{3}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_4 \\ x_4 = \frac{1}{3}x_1 + \frac{1}{2}x_2 \end{cases}$$

Esto es equivalente a preguntarse las soluciones de la ecuación

$$A \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

Es decir, debemos buscar los autovectores asociados al autovalor 1 de la matriz. Realizando los cálculos, tenemos que el autoespacio asociado al autovalor 1 es $\left\langle \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix} \right\rangle$. Como PageRank solo debería reflejar la importancia relativa de los nodos, y como los autovectores son múltiplos por un escalar entre sí, podemos elegir cualquiera de ellos como nuestro vector PageRank. Elegimos el vector v^* como el único autovector con la suma de sus entradas igual a 1 (en ocasiones se referirá a él como el autovector probabilístico correspondiente al autovalor 1). Entonces, el vector

$$v^* = \frac{1}{31} \begin{bmatrix} 12 \\ 4 \\ 9 \\ 6 \end{bmatrix} \sim \begin{bmatrix} 0{,}38 \\ 0{,}12 \\ 0{,}29 \\ 0{,}19 \end{bmatrix}$$

es nuestro vector PageRank.

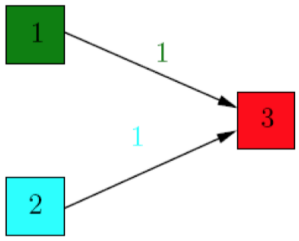
Enfoque Probabilístico

Como la importancia de una página Web es medida a través de su popularidad (la cantidad de vínculos hacia ella que tienen otras páginas), podemos ver la importancia de la página i como la probabilidad de que una persona aleatoria que abre un navegador de Internet y comienza a visitar vínculos, visite la página i. Podemos interpretar los pesos de cada arista de manera probabilística: la persona actualmente visitando la página 2 tiene probabilidad $\frac{1}{2}$ de ir a la página 3 y probabilidad $\frac{1}{2}$ de ir a la página 4. Podemos modelar este proceso como un camino aleatorio sobre grafos. Cada página tiene probabilidad $\frac{1}{4}$ de ser elegida como página inicial. Entonces, la probabilidad inicial es dada por el vector $x = \left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)^T$. La probabilidad de que la página i sea visitada luego de k pasos es igual a $A^k x$. En este caso, la sucesión $Ax, A^2x, \ldots, A^kx$ converge a un único vector probabilístico v^*. En este contexto, v^* es denominado *distribución estacionaria* y será nuestro vector PageRank. La i-ésima entrada de v^* es la probabilidad de que en un momento dado la persona visite la página i.

El vector PageRank v^* calculado indica que la página 1 es la más relevante. Esto puede parecer inesperado, ya que dos vínculos apuntan a la página 1 mientras que tres vínculos apuntan a la página 3. Sin embargo, el nodo 3 tiene solo una arista saliente que apunta al nodo 1, transfiriendo *toda* su importancia a ese nodo. Es importante notar, además, que el rango de cada página no es solo la suma ponderada de las aristas entrantes de cada nodo. Intuitivamente, en el primer paso, un nodo recibe un voto de importancia de sus vecinos directos, en el segundo paso de los vecinos de sus vecinos, y así sucesivamente.

Nodos sin Aristas Salientes (Nodos Colgantes)

Consideremos el siguiente ejemplo:



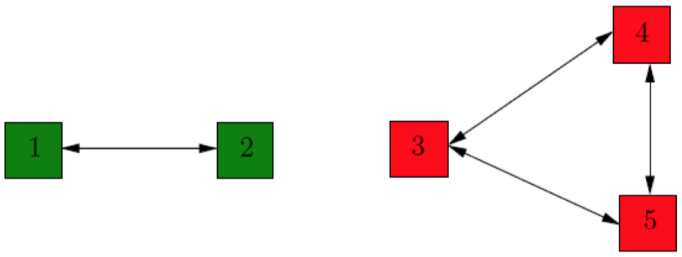
En este caso, la matriz de transición es $A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$, y realizando los cálculos llegamos al vector PageRank $v^* = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$. Es decir, el rango de todas las páginas es 0. Claramente, esto no tiene sentido, ya

que intuitivamente la página 3 debería tener mayor importancia.

Una solución fácil para este problema es reemplazar la columna correspondiente al nodo colgante 3 con una columna cuyas entradas son $\frac{1}{3}$. De esta manera, la importancia del nodo 3 estaría igualmente distribuida entre los otros nodos del grafo, en vez de perderse.

Componentes Desconectados

Veamos el siguiente ejemplo:



Si una persona que navega por las páginas comienza en el primer componente conectado, no tiene forma de visitar la página 5, pues los nodos 1 y 2 no tienen vínculos a ninguna página conectada con la página 5. En este caso, la matriz de transición es

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

Nótese que $v = (1, 1, 0, 0, 0)^T$ y $u = (0, 0, 1, 1, 1)^T$ ambos son autovectores asociados al autovalor 1, y son linealmente independientes. Entonces, clasificar páginas de acuerdo al primer componente conectado relativamente al segundo componente conectado es ambiguo.

Sea n la cantidad de nodos del grafo. Para resolver estos problemas, se fija una constante positiva p entre 0 y 1, llamada el **factor *damping*** (un valor típico es $p = 0{,}15$). Luego, se define la matriz PageRank M (también llamada la matriz Google) del grafo de la siguiente manera:

$$M = (1-p) \cdot A + p \cdot B, \text{ con } B = \frac{1}{n} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Problema 1. Probar que M es una matriz estocástica por columnas con entradas positivas.

La matriz M representa el modelo de la persona que navega a través de las páginas de la siguiente manera: la mayoría del tiempo, la persona seguirá los vínculos desde páginas; desde la página i, sigue los links salientes y va hacia los vecinos de la página i. Con un porcentaje menor de probabilidad, la persona dejará la página actual y elegirá arbitrariamente una página diferente de la Web, e irá allí. Esta probabilidad es reflejada por p, y como puede ir a cualquier página, cada página tiene probabilidad $\frac{1}{n}$ de ser elegida.

Problema 2. Realizar nuevamente los cálculos para el Page Rank reemplazando la matriz de transición A por la matriz M para los grafos explicativos de nodos colgantes y de componentes desconectados. Continúan ocurriendo en los grafos los problemas mencionados al utilizar M?

Intuitivamente, la matriz M conecta los grafos y se deshace de los nodos colgantes. Un nodo sin aristas salientes ahora tiene probabilidad $\frac{p}{n}$ de ir a cualquier otro nodo.

Definición 1. Una matriz M de tamaño $n \times n$ es **estocástica por columnas** si para todo $i = 1, \ldots, n$, $\sum_{j=1}^{n} M_j^i = 1$.

Del método anterior se pueden deducir los siguientes teoremas;

Teorema 1. *(**Teorema de Perron-Frobenius**) Si M es una matriz estocástica por columnas con entradas positivas, entonces:*

1. 1 *es un autovalor de multiplicidad 1 de M.*
2. 1 *es el mayor autovalor: todos los otros autovalores son de valor absoluto menor a* 1.
3. *Los autovectores correspondientes al autovalor* 1 *tienen solo entradas positivas o solo entradas negativas. En particular, para el autovalor* 1 *existe un único autovector donde la suma de sus entradas es igual a* 1.

Teorema 2. *(**Convergencia del Método de Potencia**) Sea M una matriz estocástica por columnas de tamaño $n \times n$. Sea v^* el vector probabilístico correspondiente al autovalor* 1. *Sea z el vector columna con entradas iguales a $\frac{1}{n}$. Entonces, la sucesión $z, Mz, \ldots, M^k z$ converge al vector v^*.*

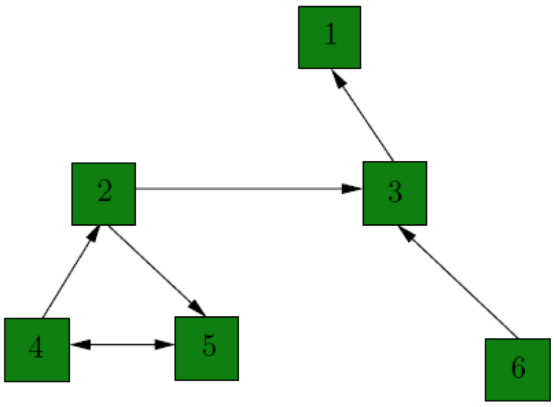
A luz de lo visto anteriormente, se puede concluir que el vector PageRank de un grafo de páginas Web con matriz de transición A y factor *damping* p, en el único autovector probabilístico de la matriz M, correspondiente al autovalor 1.

Desde el punto de vista matemático, calcular el autovector probabilístico es, en teoría, simple. Sin embargo, es ineficiente realizar estos cálculos para tamaños grandes. Una forma alternativa de calcular este vector es mediante el Método de Potencia. Computacionalmente hablando, es considerablemente más sencillo multiplicar los vectores $x, Mx, \ldots, M^n x$ hasta la convergencia que calcular los autovectores de

M para un tamaño en el orden de, por ejemplo, 10^{10}. De hecho, es posible calcular solo los primeros términos de la sucesión para obtener una buena aproximación.

Para una matriz aleatoria, se conoce que el Método de Potencia converge lentamente. Sin embargo, lo que hace que funcione en este caso es el hecho de que el grafo de la Web es poco denso: un nodo i tiene un número pequeño de links salientes (a lo sumo unos cientos, en contraposición con las 3^{10} páginas de la Web). Entonces, la matriz de transición A tiene muchas entradas nulas.

Problema 3. Calcular el vector Page Rank del siguiente grafo, con la constante *damping* p tomando los valores $p = 0$, $p = 0{,}15$, $p = 0{,}5$ y $p = 1$.



Problema 4. Calcular el vector Page Rank del siguiente árbol dirigido, considerando $p = 0{,}15$. Interpretar los resultados en términos de la relación entre el número de vínculos entrantes y el rango de cada nodo.