

CAIM LAB5: PageRank

Jiajian Cheng
Yimin Pan

November 2019

1 Introducción

En esta práctica aplicaremos el algoritmo de **Pagerank** para una red de aeropuertos y rutas, cuyo datos son extraídos desde OpenFlights y leídos por el programa en formato txt.

El algoritmo de **Pagerank** calcula la importancia de un nodo (documento) basándose en las aristas que entra en él y la importancia del nodo origen.

$$P_{k+1}[i] = \lambda \cdot \sum_j \frac{P_k[j] \cdot w(j, i)}{\text{out}(j)} + \frac{(1 - \lambda)}{n}, \quad \forall (j, i) \in \text{Routes}$$

- $w(i, j)$ es el peso de la arista del nodo i a j , en nuestro caso es el número de repeticiones de la misma arista.
- $\text{out}(i)$ representa el número de aristas que tiene como origen el nodo i .
- λ es el factor damping.
- $1 - \lambda$ es el factor de teleportación, es decir un salto aleatorio.

2 Implementación

Al inicio, el vector P está inicializado todos a $\frac{1}{n}$, de este modo asegurando siempre $\sum_i P[i] = 1$. En cada iteración actualizamos el pagerank de cada nodo, y como tenemos implementado, el número de iteraciones dependerá de una tolerancia. Definida como el máximo de diferencia entre los elementos de P_k y P_{k+1} (en el código tenemos un vector Q que hace de auxiliar), paramos cuando:

$$\forall i \in P_k \quad \forall j \in P_{k+1} \quad |P[i] - P[j]| \leq \text{tol}$$

Cada vez que se hace menor la diferencia entre iteraciones, sabemos que estamos convergiendo a hacia una la solución. Ya que en realidad estamos calculando una aproximación del pagerank teórico. En la práctica hemos puesto

$tol = 10^{-10}$, por lo que el programa puede llegar a tardar 5s en ejecución.

En cuanto a estructura de datos, para cada aeropuerto a_i tenemos un diccionario de rutas (edges) entrantes a a_i y un índice que indica la posición que ocupa dicho aeropuerto en la lista de aeropuertos. De este modo reducimos la complejidad en respecto a crear una matriz $n \cdot n$, $O(n \cdot m)$ vs $O(n^2)$, siendo m el máximo de aristas que entra en un nodo, por lo tanto $m \ll n$. También nos hemos ahorrado la lista y diccionario de rutas que había en la plantilla, dado que ahora la información de las rutas están incluidos en el diccionario que hay en la clase aeropuerto (la lista también lo hemos eliminado).

3 El problema de los nodos desconexos

En general no hemos encontrado gran dificultad a la hora de implementar *Pagerank*, era simplemente seguir el pseudocódigo del enunciado.

Pero al hacer una iteración nos dimos cuenta que el sumatorio sobre P no se acercaba a 1. Esto se debe a que había nodos que no tenían aristas entrantes, o incluso ni salientes. Pero dado el factor damping $(1 - \lambda)$, sabemos que podemos eventualmente llegar a estos nodos. De este modo, sus pagerank no es 0 y tiene aportación en los demás nodos.

La solución sería añadir aristas virtuales desde estos nodos a todos los $n - 1$ demás, sin embargo este cambio hará que los $O(n)$ aristas se convierta en $O(n^2)$ y consecuentemente un incremento importante en tiempo y espacio. En realidad podemos calcular la aportación de los nodos desconexos a los otros nodos sin tener que modificar el grafo.

- Por la forma de calcular el pagerank, sabemos que existe una parte que no varía de iteración en iteración. Todos los nodos i que no tienen aristas de salida, si los conectamos con aristas virtuales a todos los demás nodos, $out(i) = n - 1$, $n = |G|$. Por otra parte, el factor λ tampoco varía. Si el número de nodos de este tipo es m , tenemos $m \cdot \frac{\lambda}{n-1}$. En el código lo llamamos **disconnectedPRfixed**.
- La parte que puede variar es $P_k[j] \cdot w(j, i)$, pero como no existe repeticiones para estas aristas, sabemos que $w(j, i) = 1$, por lo tanto solamente tenemos que determinar cuánto vale $P_k[j]$. En el código lo llamamos **disconnectPRvariable**. Para la primera iteración está claro que vale $\frac{1}{n}$ ya que es el valor que hemos inicializado el vector P . Después lo calculamos como:

$$disconnectPRvariable_{k+1} = disconnectPRvariable_k \cdot disconnectedPRfixed + \frac{1 - \lambda}{n}.$$

Es decir, la aportación total de los nodos desconexos (no tienen otras aristas entrantes que las virtuales) más el factor de teleportación.

De este modo, para calcular el pagerank de los aeropuertos conexos, solo tenemos que sumarle $disconnectPRvariable \cdot disconnectedPRfixed$.