Métodos Numéricos
Segundo Cuatrimestre 2013
Trabajo Práctico 3

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Introducción

A partir de la evolución de Internet durante la década de 1990, el desarrollo de motores de búsqueda se ha convertido en uno de los aspectos centrales para su efectiva utilización. Hoy en día, sitios como Yahoo, Google y Bing ofrecen distintas alternativas para realizar búsquedas complejas dentro de un red que contiene miles de millones de páginas web.

En sus comienzos, una de las características que distnguió a Google respecto de los motores de búsqueda de la época fué la calidad de los resultados obtenidos, mostrando al usuario páginas relevantes a la búsqueda realizada. El esquema general de los orígenes de este motor de búsqueda es brevemente explicando en Brin y Page [2], donde se mencionan aspectos técnicos que van desde la etapa de obtención de información de las páginas disponibles en la red, su almacenamiento e indexado y su posterior procesamiento, buscando ordenar cada página de acuerdo a su importancia relativa dentro de la red. El algoritmo utilizado para esta última etapa es denominado PageRank y es uno (no el único) de los criterios utilizados para ponderar la importancia de los resultados de una búsqueda. En este trabajo nos concentraremos en el estudio y desarrollo del algoritmo PageRank.

El problema

El algoritmo PageRank se basa en la construcción del siguiente modelo. Supongamos que tenemos una red con n páginas web $Web = \{1, \ldots, n\}$ donde el objetivo es asignar a cada una de ellas un puntaje que determine la importancia relativa de la misma respecto de las demás. Para modelar las relaciones entre ellas, definimos la *matriz de conectividad* $W \in \{0,1\}^{n \times n}$ de forma tal que $w_{ij} = 1$ si la página j tiene un link a la página i, y $w_{ij} = 0$ en caso contrario. Además, ignoramos los *autolinks*, es decir, links de una página a sí misma, definiendo $w_{ii} = 0$. Tomando esta matriz, definimos el grado de la página j, n_j, como la cantidad de links salientes hacia otras páginas de la red, donde $n_j = \sum_{i=1}^{n} w_{ij}$. Además, notamos con x_j al puntaje asignado a la página $j \in Web$, que es lo que buscamos calcular.

La importancia de una página puede ser modelada de diferentes formas. Un link de la página $u \in Web$ a la página $v \in Web$ puede ser visto como que v es una página importante. Sin embargo, no queremos que una página obtenga mayor importancia simplemente porque es apuntada desde muchas páginas. Una forma de limitar esto es ponderar los links utilizando la importancia de la página de origen. En otras palabras, pocos links de páginas importantes pueden valer más que muchos links de páginas poco importantes. En particular, consideramos que la importancia de la página v obtenida mediante el link de la página u es proporcional a la importancia de la página u e inversamente proporcional al grado de u. Si la página u contiene n_u links, uno de los cuales apunta a la página v, entonces el aporte de ese link a la página v será x_u/n_u. Luego, sea $L_k \subseteq Web$ el conjunto de páginas que tienen un link a la

página k. Para cada página pedimos que

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j}, \qquad k = 1, \ldots, n. \tag{1}$$

Definimos $P \in \mathbb{R}^{n \times n}$ tal que $p_{ij} = 1/n_j$ si $w_{ij} = 1$, y $p_{ij} = 0$ en caso contrario. Luego, el modelo planteado en (1) es equivalente a encontrar un $x \in \mathbb{R}^n$ tal que $Px = x$, es decir, encontrar (suponiendo que existe) un autovector asociado al autovalor 1 de una matriz cuadrada, tal que $x_i \geq 0$ y $\sum_{i=1}^{n} x_i = 1$. En Bryan y Leise [3] y Kamvar et al. [4, Sección 1] se analizan ciertas condiciones que debe cumplir la red de páginas para garantizar la existencia de este autovector.

Una interpretación equivalente para el problema es considerar al *navegante aleatorio*. Éste empieza en una página cualquiera del conjunto, y luego en cada página j que visita sigue navegando a través de sus links, eligiendo el mismo con probabilidad $1/n_j$. Una situación particular se da cuando la página no tiene links salientes. En ese caso, consideramos que el navegante aleatorio pasa a cualquiera de las página de la red con probabilidad $1/n$. Para representar esta situación, definimos $v \in \mathbb{R}^{n \times n}$, con $v_i = 1/n$ y $d \in \{0, 1\}^n$ donde $d_i = 1$ si $n_i = 0$, y $d_i = 0$ en caso contrario. La nueva matriz de transición es

$$\begin{aligned} D &= v d^t \\ P_1 &= P + D. \end{aligned}$$

Además, consideraremos el caso de que el navegante aleatorio, dado que se encuentra en la página j, decida visitar una página cualquiera del conjunto, independientemente de si esta se encuentra o no referenciada por j (fenómeno conocido como *teletransportación*). Para ello, consideramos que esta decisión se toma con una probabilidad $c \geq 0$, y podemos incluirlo al modelo de la siguiente forma:

$$\begin{aligned} E &= v \bar{1}^t \\ P_2 &= c P_1 + (1 - c) E, \end{aligned}$$

donde $\bar{1} \in \mathbb{R}^n$ es un vector tal que todas sus componenetes valen 1. La matriz resultante P_2 corresponde a un enriquecimiento del modelo formulado en (1). Probabilísticamente, la componente x_j del vector solución (normalizado) del sistema $P_2 x = x$ representa la proporción del tiempo que, en el largo plazo, el navegante aleatorio pasa en la página $j \in Web$.

En particular, P_2 corresponde a una matriz *estocástica por columnas* que cumple las hipótesis planteadas en Bryan y Leise [3] y Kamvar et al. [4], tal que P_2 tiene un autovector asociado al autovalor 1, los demás autovalores de la matriz cumplen $1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$ y, además, la dimensión del autoespacio asociado al autovalor λ_1 es 1. Luego, la solución al sistema $P_2 x = x$ puede ser calculada de forma estándar utilizando el método de la potencia.

Enunciado

El objetivo del trabajo es implementar el algoritmo PageRank, considerando dos métodos distintos para el cálculo del autovector principal de la matriz P_2. Para ello, se considera el entorno de aplicación real del algoritmo, donde el número total de páginas, n, es considerablemente grande (i.e., todas las páginas web accesibles públicamente). El programa tomará como

entrada un archivo con la representación de grafo de conectividad, construirá la matriz P_2 definida en la sección anterior y ejecutará el algoritmo PageRank utilizando el método de la potencia y una variante del mismo para distintas instancias de prueba. Se pide:

1. En base a su definición, P_2 no es una matriz esparsa. Sin embargo, en Kamvar et al. [4, Algoritmo 1] se propone una forma alternativa para computar $x^{(k+1)} = P_2 x^{(k)}$. Mostrar que el cómputo propuesto es equivalente. Utilizarlo para mejorar el espacio requerido en memoria para el almacenamiento de la matriz P_2 y el tiempo de ejecución requerido para hacer la multiplicación entre matrices y vectores.

2. Basándose en el análisis del punto anterior, implementar el método de la potencia para calcular el autovector principal de la matriz P_2.

3. Implementar la variante del Método de la Potencia propuesta en Kamvar et al. [4, Sección 5], denominada Extrapolación Cuadrática. El método de Cuadrados Mínimos involucrado debe ser resuelto utilizando la Factorización QR de la matriz mediante alguno de los métodos vistos en la materia.

4. Realizar experimentos considerando distintas instancias de prueba. Para ello, se podrá utilizar el código adjuntada para la generación de instancias en base a datos reales, o cualquier otra herramienta que el grupo considere necesaria. Evaluar también los algoritmos alguno de los conjuntos de instancias provistos en [1]. Para cada algoritmo, analizar el tiempo de ejecución, la evolución del error entre iteraciones consecutivas y considerar distintos criterios de parada. Además, analizar la calidad del ordenamiento obtenido en términos de la relevancia de las páginas.

Formato de archivos

El programa deberá recibir como parámetro un archivo con la representación esparsa de la matriz (grafo) de conectividad. El archivo contendrá una línea con la cantidad de páginas (n), la cantidad de links (m) y luego una lista con un link por línea, indicando la página de origen y destino separadas por un espacio. A modo de ejemplo, a continuación se muestra el archivo de entrada correspondiente al ejemplo propuesto en Bryan y Leise [3, Figura 1]:

```
4
8
1   2
1   3
1   4
2   3
2   4
3   1
4   1
4   3
```

Una vez ejecutado el algoritmo, el programa deberá generar un archivo de salida que contenga una línea por cada página, acompañada del puntaje obtenido por el algoritmo PageRank, ordenados en forma decreciente en función de este último valor.

Para generar instancias, es posible utilizar el código Python provisto por la cátedra. Es importante mencionar que, para que el mismo funcione, es necesario tener acceso a Internet. En caso de encontrar un bug en el mismo, por favor contactar a los docentes de la materia a través de la lista. Desde ya, el código puede ser modificado por los respectivos grupos agregando todas aquellas funcionalidades que consideren necesarias.

Fechas de entrega

- *Formato Electrónico:* Domingo 10 de Noviembre de 2013, hasta las 23:59 hs, enviando el trabajo (informe + código) a la dirección `metnum.lab@gmail.com`. El subject del email debe comenzar con el texto `[TP3]` seguido de la lista de apellidos de los integrantes del grupo.
- *Formato físico:* Lunes 11 de Noviembre de 2013, de 17 a 18 hs.

Importante: El horario es estricto. Los correos recibidos después de la hora indicada serán considerados re-entrega.

Referencias

[1] Stanford large network dataset collection. `http://snap.stanford.edu/data/#web`.

[2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, April 1998.

[3] Kurt Bryan and Tanya Leise. The linear algebra behind google. *SIAM Review*, 48(3):569–581, 2006.

[4] Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Extrapolation methods for accelerating pagerank computations. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 261–270, New York, NY, USA, 2003. ACM.