Métodos Numéricos
Segundo Cuatrimestre 2014
Trabajo Práctico 2

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

Tirate un qué, tirate un *ranking...*

Motivación

Luego de su repentina y efímera irrupción durante el año 2011, un grupo de la movida tropical[1] está buscando recuperar la notoriedad y los niveles de popularidad otrora alcanzados. El retorno incluye, entre otras cosas, un mega recital gratuito, giras por las principales *bailantas* y por el interior del país.[2]

Para que toda esta movida sea exitosa, los miembros del grupo han acordado con su *community manager* que, además de tener una participación destacada en Pasión de Sábado, es necesario que la llegada a través de los medios electrónicos y las redes sociales sea muy efectiva, al igual que en 2011, alcanzando a la mayor cantidad posible de gente y poder, nuevamente, sentarse en el living de *la diva de los teléfonos*. La conclusión a la que llegaron es que necesitan que cada vez que realiza una búsqueda relacionada con la movida tropical, su página se encuentre entre las primeras que muestran los buscadores.

Con ese motivo, se han contactado con el equipo de R+D de Métodos Numéricos, donde en la primera reunión el cliente propuso *comprar clicks en publicidades*. Ésta, si bien es una alternativa viable, representa un gasto importante para la escala de inversión con la que se dispone. Luego de una reunión del equipo técnico, se les hizo una contrapropuesta: estudiar el comportamiento de los buscadores y, a cambio de shows libres de costo y presentaciones privadas, buscar en qué páginas conviene figurar para mejorar el posicionamiento virtual del grupo.

Contexto

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 distinguió a Google respecto de los motores de búsqueda de la época fue 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 explicado en Brin y Page [3], 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

[1] Por cuestiones de privacidad, no haremos público de qué grupo se trata.

[2] A riesgo de exponer su edad, los miembros de la cátedra quieren destacar a aquellos próceres que llevaron a este género musical a las primeras planas, como Alcides, Sebastián, Miguel *Conejito* Alejandro, Ráfaga, La Nueva Luna, Comanche y, como dejar fuera, al *MAESTRO* Antonio Ríos.

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.

Los métodos, Parte I: PageRank

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 [4] y Kamvar et al. [5, 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

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

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:

$$\begin{aligned} E &= v\bar{1}^t \\ P_2 &= cP_1 + (1-c)E, \end{aligned}$$

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 [4] y Kamvar et al. [5], 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.

Una vez calculado el ranking, se retorna al usuario las t páginas con mayor ranking.

Los métodos, Parte II: Hyperlink-Induced Topic Search

Un método alternativo es propuesto en Kleinberg [6], denominado *Hyperlink-Induced Topic Search* (HITS). La intuición del método se basa en el análisis intríniseco de la red, donde una noción de *autoridad* se transfiere de una página a otra mediante los links que las relacionan. El objetivo es, dada una búsqueda concreta, retornar un subconjunto acotado de páginas relevantes. Con este fin, se considera que existen páginas que cumplen un rol de *autoridad* sobre un tema específico y se busca modelar la relación entre estas páginas y aquellas que apuntan a varias de estas autoridades, denominadas *hubs*. En la práctica, los autores observan que suele existir una especie de equilibrio en la relación entre hubs y autoridades, y se busca aprovechar esta relación para el desarrollo del algoritmo. Intuitivamente, un buen *hub* es una página que apunta a muchas autoridades, y una buena *autoridad* es una página que es apuntada por muchos *hubs*.

El procedimiento consiste en los siguientes pasos. Dada una búsqueda concreta, se utiliza en primer lugar un *buscador* simple (por ejemplo, basado en texto) para obtener un conjunto acotado de paginas (digamos, 200), llamado *root set*. Luego, asumiendo que la estructura de la red es conocida, es busca extender este conjunto agregando páginas que son apuntadas y que apuntan a las páginas de *root set*, hasta llegar a una sub-red de un tamaño determinado. En el contexto del trabajo práctico, asumiremos que este paso ha sido realizado y que contamos con el grafo que considera la sub-red.

Formalmente, y retomando la notación introducida en la sección anterior, consideramos que las páginas de nuestra sub-red se encuentran en el conjunto $Web = \{1, \ldots, n\}$. Para modelar las relaciones entre las páginas, adoptamos una definición similar: consideramos la matriz de adyacencia $A \in \{0,1\}^{n \times n}$ donde $a_{ij} = 1$ si existe un link de la página i a la página j.[3] Para cada página $i \in Web$ se considera el *peso de autoridad* x_i y el *peso de hub* y_i. Consecuentemente, se definen los vectores $x, y \in \mathbb{R}^n$ los vectores de pesos de autoridad y

[3] Notar que $A = W^t$.

hubs, respectivamente, y supondremos además que se encuentran normalizados. Las páginas con mayores valores de x_i e y_i son consideradas mejores *autoridades* y *hubs*, respectivamente.

La relación mencionada entre los distintos tipos de páginas se expresan numéricamente de la siguiente forma. Dados los vectores x, y, la operación de transferencia de los *hubs* a la autoridad $j \in Web$ puede expresarse de la siguiente forma:

$$x_j = \sum_{i:i \to j} y_i. \tag{2}$$

Análogamente, el peso de un hub está dado por la siguiente ecuación

$$y_i = \sum_{j:i \to j} x_j. \tag{3}$$

Las ecuaciones (2) y (3) podemos expresarlas matricialmente de la siguiente manera:

$$x = A^t y \tag{4}$$
$$y = Ax, \tag{5}$$

aplicando luego el paso de normalización correspondiente. Los autores proponen comenzar con un y_0 incial, aplicar estas ecuaciones iterativamente y demuestran que, bajo ciertas condiciones, el método converge. Finalmente, en base a los rankings obtenimos, se retorna al usuario las mejores t *autoridades* y los mejores t *hubs*.

Enunciado

El objetivo del trabajo es experimentar en el contexto planteado utilizando los algoritmos de ranking propuestos. Para ello, se considera un entorno que, dentro de nuestras posibilidades, simule el contexto real de aplicación donde se abordan instancias de gran escala (es decir, n, el número total de páginas, es grande). El archivo tomará como entrada un archivo que especifique el algoritmo, los parámetros del mismo y un puntero al grafo de la red y retorne como resultado el ranking obtenido para cada página. Los detalles sobre el input/output del programa son especificados en la siguiente sección.

El trabajo consistirá en estudiar distintos aspectos de los siguientes métodos: PageRank, HITS, e In-deg, éste último consiste en definir el ranking de las páginas utilizando solamente la cantidad de ejes entrantes a cada una de ellas, ordenándolos en forma decreciente. Para tener una descripción más completa de los dos primeros métodos, se propone:

1. Considerar el trabajo de Kleinberg [6] con los detalles sobre HITS, en particular las secciones 1, 2 y 3.

2. Considerar el trabajo de Bryan y Leise [4] donde se explica la intución y algunos detalles técnicos respecto a PageRank. Además, en Kamvar et al. [5] se propone una mejora del mismo. Si bien esta mejora queda fuera de los alcances del trabajo, en la Sección 1 se presenta una buena formulación del algoritmo. En base a su definición, P_2 no es una matriz esparsa. Sin embargo, en Kamvar et al. [5, Algoritmo 1] se propone una forma alternativa para computar $x^{(k+1)} = P_2 x^{(k)}$. Este resultado puede ser utilizado para mejorar el almacenamiento de los datos.

3. (Opcional) Completar la demostración del Teorema 3.1 de Kleinberg [6], incluyendo el detalle de los puntos que el autor asume como triviales.

En la práctica, el grafo que representa la red de páginas suele ser esparso, es decir, una página posee relativamente pocos links de salida comparada con el número total de páginas. A su vez, dado que n tiende a ser un número muy grande, es importante tener en cuenta este hecho a la hora de definir las estructuras de datos a utilizar. Luego, desde el punto de vista de implementación se pide utilizar alguna de las siguientes estructuras de datos para la representación de las matrices esparsas: *Dictionary of Keys* (dok), *Compressed Sparse Row* (CSR) o *Compressed Sparse Column* (CSC). Se deberá incluir una justificación respecto a la elección que consdiere el contexto de aplicación. Una vez definida la estructura a utilizar, se deberá implementar el algoritmo HITS utilizando las ecuaciones (4) y (5). Para el caso de PageRank, se debe implementar el método de la potencia para calcular el autovector principal.

En función de la experimentación, se deberá realizar un estudio particular para cada algoritmo (tanto en términos de comportamiento del mismo, como una evaluación de los resultados obtenidos) y luego se procederá a comparar cualitativamente los rankings generados. La experimentación deberá incluir como mínimo los siguientes experimentos:

1. Estudiar la convergencia de PageRank, analizando la evolución de la norma Manhattan (norma L_1) entre dos iteraciones sucesivas. Comparar los resultados obtenidos para al menos dos instancias de tamaño mediano-grande, variando el valor de c. Opcional: Establecer una relación con la proporción entre $\lambda_1 = 1$ y $|\lambda_2|$.

2. Estudiar la convergencia de los vectores de peso x e y para HITS de forma similar al punto anterior.

3. Estudiar el tiempo de cómputo requerido por PageRank y HITS. Si bien ambos pueden se aplicados sobre una red genérica, cada algoritmo tiene un contexto particular de aplicación. Estudiar como impacta el factor temporal en este sentido.

4. Estudiar cualitativamente los rankings obtenidos por los tres métodos. Para ello, se sugiere considerar distintos ejemplos de búquedas de páginas web[4]. Analizar los resultados individualmente en una primera etapa, y luego realizar un análisis comparativo entre los tres rankings obtenidos.

5. Para cada algoritmo, proponer ejemplos de tamaño pequeño que ilustren el comportamiento esperado (puede ser utilizando las instancias provistas por la cátedra o generadas por el grupo).

Finalmente, y en base a la experimentación realizada, buscamos resolver el problema planteado originalmente: dada una foto de la red, con sus interconexiones entre páginas, supongamos que tenemos los pesos (ranking) asignados por uno de los algoritmos estudiados. ¿Cuál sería la estrategia que le sugiere al cliente para mejorar su correspondiente ranking? Para este último punto, suponer que es posible *negociar* que una página apunte a nuestro sitio, y que la cantidad de estas negociaciones que podemos tener es acotada.

Parámetros y formato de archivos

[4]La cátedra adjunta casos de *benchmark* que representan sub-redes obtenidas en base a búsquedas temáticas

El programa deberá tomar por línea de comandos dos parámetros. El primero de ellos contendrá la información del experimento, incluyendo el método a ejecutar (`alg`, 0 para PageRank, 1 para HITS, 2 para IN-DEG), la probabilidad de teletransportación c en el caso de PageRank (que valdrá -1 si `alg` no es 0), el tipo de instancia, el *path* al archivo/directorio conteniendo la definición de la red (que debe ser relativa al ejecutable, o el path absoluto al archivo) y el valor de tolerancia utilizado en el criterio de parada impuesto a cada método. El siguiente ejemplo muestra un caso donde se pide ejecutar PageRank, con una probabilidad de teletransportación de 0.85, sobre la red descripta en `red-1.txt` (que se encuentra en el directorio `tests/`) y con una tolerancia de corte de 0,0001.

```
0 0.85 0 tests/red-1.txt 0.0001
```

Para la definición del grafo que representa la red, se consideran dos bases de datos de instancias con sus correspondientes formatos. La primera de ellas es el conjunto provisto en SNAP [2] (el tipo de instancia es 0), con redes de tamaño grande obtenidos a partir de datos reales. Además, se consideran las instancias propuestas en [1]. Estas instancias son de tamaño mediano, obtenidas también en base a datos reales, y corresponden a redes temáticas obtenidas a partir de una búsqueda particular. Para cada nodo de la red se tiene: la direccion URL, una breve descripción, y las páginas a las cuales apunta. Si bien algunas de las URL ya no son válidas, la descripción permite tener algo más de información para realizar un análisis cualitativo.

En el caso de la base de SNAP, los archivos contiene primero cuatro líneas con información sobre la instancia (entre ellas, n y la cantidad total de links, m) y luego m líneas con los pares i, j indicando que i apunta a j. A modo de ejemplo, a continuación se muestra el archivo de entrada correspondiente a la red propuesta en Bryan y Leise [4, Figura 1]:

```
# Directed graph (each unordered pair of nodes is saved once):
# Example shown in Bryan and Leise.
# Nodes: 4 Edges: 8
# FromNodeId    ToNodeId
1   2
1   3
1   4
2   3
2   4
3   1
4   1
4   3
```

Para la otras instancias, en [1] puede encontrarse una descripción del formato propuesto (el tipo de instancia será 1 en este caso).

Una vez ejecutado el algoritmo, el programa deberá generar un archivo de salida que contenga una línea por cada página (n líneas en total), acompañada del puntaje obtenido por el algoritmo PageRank/IN-DEG. En el caso de HITS, el archivo contendrá $2n$ líneas, las primeras n con el *peso de autoridad* y las segundas n con el *peso de hub* para los vértices $1, \ldots . n$.

Para generar instancias, es posible utilizar el código Python provisto por la cátedra. La utilización del mismo se encuentra descripta en el archivo README. 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:* Sábado 11 de Octubre de 2014, 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 `[TP2]` seguido de la lista de apellidos de los integrantes del grupo.
- *Formato físico:* Miércoles 15 de Octubre de 2014, a las 17 hs. en la clase teórica.

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

Referencias

[1] `http://www.cs.toronto.edu/~tsap/experiments/datasets/`.

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

[3] 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.

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

[5] 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.

[6] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, September 1999.