Trabajo Práctico 2

Tirate un qué, tirate un *ranking*...

Métodos Numéricos
Segundo Cuatrimestre de 2014

Integrante	LU	Correo electrónico
Fosco, Martin Esteban	449/13	mfosco65@gmail.com
Minces Müller, Javier Nicolás	231/13	javijavi1994@gmail.com
Chibana, Christian Ezequiel	586/13	christian.chiba93@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Resumen

En este trabajo se analizan tres métodos para crear un $ranking$ de páginas web: PageRank, HITS e In-Deg. Estos algoritmos utilizan matrices para modelar la Web mediante matrices y utilizan métodos numéricos, como el método de la potencia, para ordenar las páginas a partir de los links que las relacionan.

Índice

1. Introducción Teórica

El problema a resolver es encontrar una manera confiable de definir en qué páginas es conveniente comprar espacios de propaganda para garantizar la mayor publicidad posible a un grupo de música. El objetivo de este trabajo, por tanto, es resolver este problema analizado diferentes métodos que nos permitan definir un ranking de importancia de páginas web.

En un primer paso se debe acordar un método de selección de las páginas web sobre las cuales se trabaja, es decir que es necesario adquirir una subred de páginas relevantes al tema en cuestión

Se recurre en primer lugar a un buscador textual, pero este no asegura un resultado confiable en cuanto a la relevancia de las páginas recibidas. En el paper escrito por Kleinberg [1] se sugiere un conjunto inicial de páginas S_σ que cumple las siguientes condiciones:

- S_σ es relativamente pequeña.
- S_σ contiene una gran cantidad de páginas relevantes.
- S_σ contiene muchas de las autoridades de más peso.

El autor propone una forma de ampliar este conjunto inicial para asegurar que se cumpla la tercera condición: buscar autoridades de gran peso a las que haya links en la subred obtenida del buscador e incluir éstas tambien en S_σ.

Se construye luego un mapa de este conjunto de páginas en forma de grafo dirigido, observando los links entre ellas. Hasta este punto, el proceso escapa al objetivo del trabajo, que considera dado este mapa.

Terminado este paso, se procede a construir la matriz de conectividad $W \in n \times n$, tal que $w_{ij} = 1$ si existe un link de la página j a la página i y $w_{ij} = 0$ en caso contrario.

Usando esta matriz (con la misma notación), se pueden aplicar distintos métodos para procesar y definir un cierto *ranking* de las páginas. Esta matriz es esparsa, porque incluso en redes de varios cientos de miles de páginas, en promedio cada una tiene 7 u 8 links salientes. [2] Esto debe ser tenido en cuenta a la hora de guardar la matriz en memoria.

Los métodos a analizar son:

PageRank, el más conocido de los utilizados por Google. PageRank asigna un puntaje a cada página según el puntaje de las que apuntan a ella. De esta forma se evita que una página con muchos links de páginas sin valor obtenga un puntaje demasiado alto. Además, modifica W para tener en cuenta que un ”navegante aleatorio”puede con una determinada probabilidad saltar a cualquier página de la web, algo que también hace si encuentra una página sin links salientes.

Esto se traduce como buscar un vector x tal que $Wx = x$, es decir, encontrar el autovector para el autovalor 1. Si se sabe que 1 es el autovalor de mayor módulo, se puede encontrar su autovector con el método de las potencias.

HITS (Hyperlink-Induced Topic Search), este se basa en la existencia de páginas de gran valor como ’autoridad’ y como ”hub”. En cada iteración, se actualiza el puntaje de cada página como autoridad a partir del puntaje hub de las páginas que la citan. Luego, se actualiza el puntaje como hub de cada página a partir del puntaje como autoridad de las páginas a las que apunta. Esto se puede reducir a una operación matricial.

In-Deg, El tercer método, el más intuitivo, se incluye principalmente como control. In-Deg consiste simplemente en contar los links que apuntan a cada página.

[1] Kleinberg - 1999 - *Authoritative sources in a hyperlinked environment*
[2] Kamvar

2. Desarrollo

2.1. Criterio de almacenamiento

Debimos primero encontrar una forma de almacenar los datos de los links entre páginas provistos (independientemente del formato en que fueron dados). Estos se guardan en la matriz W, ya mencionada.

Entre las 3 opciones sugeridas en el enunciado se optó finalmente por el almacenamiento en forma de Compressed Sparse Column (CSC)[3] porque notamos que esto facilita aquellas operaciones que pueden ser necesarias para los algoritmos de mayor complejidad temporal (Pagerank y HITS) como:

- La suma de columnas (ya que se trata de sumar una sección de elementos contiguos del array en el que se guardan los valores de la matriz distintos de 0).
- La división de una columna por una constante, debido a que al momento de definir en pagerank la probabilidad de salir de una página a otra, debemos dividir las columnas de forma que sus elementos (las probabilidades de salir hacia alguna página) sumen 1.
- El producto de una matriz transpuesta por un vector, ya que puede recorrer una sección del vector de valores y multiplicarlos por una posición del vector. Para el producto de la matriz sin transponer el algoritmo es más complejo pero igual de eficiente: alcanza con recorrer los valores e ir acumulando en las distintas posiciones del vector resultado los valores parciales que se obtienen.

Además, tiene una eficiencia espacial óptima para el almacenamiento de matrices que cuentan con muchos ceros (como es el caso de las matrices con las que trabajamos). Más específicamente : la memoria usada es $2 \times O(m) + O(n)$.[4]

En comparación con las otras dos opciones de almacenamiento:

Dictionary of Keys cuenta con sólo la última ventaja del CSC, que permite obviar a las posiciones en las que se guardan ceros, pero no es capaz de mejorar en ninguna manera la eficiencia de las operaciones matriciales, que se efectúan en los métodos de ranking más complicados.

Compressed Sparse Row (CSR) cuenta con la última ventaja del CSC también, pero no es capaz de operar con columnas de manera tan eficiente como el CSC.
Es necesario mencionar, sin embargo, que es una estructura más eficiente para realizar las operaciones que el *In-Deg* requiere (por filas) por razones obvias, pero optamos por la *Compressed Sparse Column* debido a que parece mejor optimizar las operaciones por columnas que intervienen en los algorítmos mas complejos (es posible modificar, sin embargo, las operaciones entre columnas para que se puedan realizar con una eficiencia análoga a las efectuadas por el CSC).

[3] Una matriz esparsa que se recorre por columnas.
[4] m es la cantidad de elementos diferentes de 0 y n las filas de la matriz.

2.2. PageRank

El algoritmo PageRank asigna un puntaje de importancia a cada página entre 0 y 1. Este no depende únicamente de la cantidad de links entrantes, sino que pondera cada link según el puntaje de la página al que pertenece. Sea x el vector en $\mathbb{R}^n$ donde x_j es el puntaje de la página j y n_j el $grado$ de j, es decir, la cantidad de links salientes, entonces se quiere ver que:

$$x_k = \sum_{j=1, w_{kj}=1}^{n} \frac{x_j}{n_j}, \quad k = 1, \ldots, n. \tag{1}$$

Supongamos que se tienen cuatro páginas para hacer el ranking (1, 2, 3, 4) y que están conectadas de esta forma:

$1 \to 2$
$1 \to 3$
$2 \to 3$
$3 \to 2$
$3 \to 4$

Entonces el problema se reduciría a buscar la solución a este sistema de ecuaciones:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Llamamos P a la matriz, que se construye dividiendo cada columna de W por su suma. Se puede ver que la solución de este sistema de ecuaciónes es un vector x tal que $Px = x$.

Se define un autovalor λ de una matriz A a un valor tal que $Ax = \lambda x$, y a x como su autovector. En este caso, entonces, el problema se reduce a buscar un autovector para el autovalor 1. Dado que este autovector representa probabilidades, debe tener norma 1 igual a 1.

El método de la potencia permite encontrar el autovalor de mayor módulo de una matriz. Para poder aplicarlo en este caso, deben cumplirse las siguientes condiciones [5]:

- P tiene un autovector asociado al autovalor 1
- Los demás autovalores de la matriz cumplen $1 = \lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$
- La dimensión del autoespacio asociado al autovalor λ_1 es 1: esto garantiza la unicidad del autovector de norma 1 igual a 1 asociado al autovalor 1.

Si la matriz P es estocástica por columnas, se puede ver que se cumple las primeras dos condiciones: Sea A una matriz en $\mathbb{R}^{n \times n}$ estocástica por columnas y sea e un vector en R^n tal que $e_i = 1 \ \forall i$, sabemos que A y su transpuesta A^T tienen los mismos autovalores. Como A es estocástica por columnas, se puede ver que $A^T e = e$, por lo que 1 es un autovalor de A^T y, por tanto, de A. Además al estar dividida cada columna por su norma 1, la matriz es efectivamente estocástica por columnas, salvo que haya columnas cuya suma es 0. Esto se da en el caso de los llamados "dangling nodes", páginas sin links salientes.

Se puede interpretar la navegación en la Web de un $navegantealeatorio$ como un proceso de Markov, a P como su matriz de transición y a la componente x_j del vector solución de norma 1 del sistema $Px = x$ como la proporción del tiempo que el navegante pasa en la página j Entonces, se puede suponer que al encontrarse en una página sin links salientes ($danglingnode$), irá a cualquiera con probabilidad $1/n$. Se define entonces la matriz P_1, en la que se asigna 1/n, entonces, a cada fila de cada columna que representa un dangling node. En el ejemplo anterior:

$$P_1 = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{6} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{6} \\ \frac{1}{2} & 1 & 0 & \frac{1}{6} \\ 0 & 0 & \frac{1}{2} & \frac{1}{6} \end{bmatrix}$$

[5] Propuestas en *Bryan, Leise - 2006 - The Linear Algebra behind Google* $*$ y *Kamvar, Haveliwala - 2003 - Extrapolation methods for accelerating PageRank computations*

Sin embargo, la última condición no se cumple necesariamente. Si se tienen dos subredes aisladas, entonces la dimensión del autoespacio será 2. Por ejemplo, sean las páginas (1,2,3,4), con los siguientes links:

$1 \to 2$
$2 \to 1$
$3 \to 4$
$4 \to 3$

Su matriz de conectividad es:

$$W = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Se puede ver que existen dos vectores diferentes de norma 1 tales que $Wx = x$, $x = \left(\frac{1}{2}, \frac{1}{2}, 0, 0\right)$ y $x = \left(0, 0, \frac{1}{2}, \frac{1}{2}\right)$

Para evitar esto, se agrega un parámetro $c > 0$, donde $1 - c$ representa la probabilidad de que un navegante aleatorio vaya a una página cualquiera. Se define entonces P_2 tal que $P_{2ij} = cP_{1ij} + \frac{1-c}{n} \forall i$

P_2 cumple las tres condiciones, luego se puede buscar la solución de $P_2 x = x$ con el método de la potencia. Sin embargo, no es esparsa; de hecho, no tiene ceros.

En el último ejemplo, con c=$4/5$:

$$W = \begin{bmatrix} \frac{1}{20} & \frac{17}{20} & \frac{1}{20} & \frac{1}{20} \\ \frac{17}{20} & \frac{1}{20} & \frac{1}{20} & \frac{1}{20} \\ \frac{1}{20} & \frac{1}{20} & \frac{1}{20} & \frac{17}{20} \\ \frac{1}{20} & \frac{1}{20} & \frac{17}{20} & \frac{1}{20} \end{bmatrix}$$

Y el PageRank resultante sería:

1: 0,25
2: 0,25
3: 0,25
4: 0,25

2.2.1. Método de la Potencia

Este método permite encontrar al autovalor de mayor módulo de una matriz.

Sea $A \in \mathbb{R}^{n \times n}$, $\lambda_1 ... \lambda_n$ sus autovalores tales que $|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_n|$, v_1 ... v_n una base de autovectores y x un vector cualquiera. x se puede escribir como combinación lineal de los autovectores de A:

$$\begin{aligned} x &= \sum_{j=1}^{n} \alpha_j v_j \\ Ax &= \sum_{j=1}^{n} \alpha_j A v_j \end{aligned}$$

Como v_j es autovector de A, entonces:

$$\begin{aligned} Ax &= \sum_{j=1}^{n} \alpha_j \lambda_j v_j \\ A^2 x &= \sum_{j=1}^{n} \alpha_j \lambda_j A v_j \\ A^2 x &= \sum_{j=1}^{n} \alpha_j \lambda_j^2 v_j \end{aligned}$$

$$
\begin{aligned}
&\vdots \\
A^k x &= \lambda_1^k \cdot (\sum_{j=1}^{n} \frac{\lambda_j}{\lambda_1}^k v_j) \\
A^k x &= \lambda_1^k \cdot (\alpha_1 v_1 \sum_{j=2}^{n} \frac{\lambda_j}{\lambda_1}^k v_j) \\
\frac{||A^k x||}{||A^{k-1x}||} &= \frac{||\lambda_1^k \cdot (\alpha_1 v_1 \sum_{j=2}^{n} \frac{\lambda_j}{\lambda_1}^k v_j)||}{||\lambda_1^{k-1} \cdot (\alpha_1 v_1 \sum_{j=2}^{n} \frac{\lambda_j}{\lambda_1}^{k-1} v_j)||}
\end{aligned}
$$

cuando $k \to \infty$, $\frac{\lambda_j}{\lambda_1} \to 0$ porque $\lambda_1 > \lambda_j \ \forall j$, luego

$$\frac{||A^k x||}{||A^{k-1} x||} \to \frac{||\lambda_1^k \cdot (\alpha_1 v_1)||}{||\lambda_1^{k-1} \cdot (\alpha_1 v_1)||} = \lambda_1$$

$A^k x = ((...((AA)A)...A)A)x = A(A(A(...(A(Ax)))$
Es decir que se puede implementar un algoritmo que calcule Ax y asigne el resultado a x, siendo este el autovector buscado.

Para el contexto particular del PageRank, utilizamos el algoritmo 1 propuesto por Kamvar[6], que permite calcular $P_2 x$ a partir de Px. Esto permite ahorrar tiempo y memoria, ya que P es una matriz esparsa. El algoritmo consiste de tres pasos:

$$
\begin{aligned}
y &= c(Px) \\
w &= ||x||_1 - ||y||_1 \\
y &= wv
\end{aligned}
$$

Donde v es un vector en $\mathbf{R}^n$ tal que $v_i = 1/n \forall i$
Es decir, $y_i = c(Px)_i + (1 - c||Px||_1)/n$

2.3. Hits

La idea detrás de este método es, como se ha explicado brevemente antes, realizar un análisis de la red y sus páginas viéndolas como hubs y autoridades, nociones que son definidas por las relaciones (links) que tienen con el resto de la red.

Dada una búsqueda, el objetivo es retornar un subconjunto de páginas relevantes. Se asigna entonces sobre las páginas el rol de $autoridad$ en el tema que se busca. Su valor como $autoridad$ está definido por el valor de hub de las páginas que apuntan a ella mientras que el valor o peso de hub está definido, a su vez, por el valor de $autoridad$ de las páginas a las que apunta.

Recordemos que los links están almacenados en la matriz de conectividad W^t donde $w_{ij}^t = 1$ si existe un link de la página j a la página i, $w_{ij}^t = 0$ en caso contrario. Para cada página $i \in Web$ se define como peso de autoridad x_i y como peso de hub y_i. A partir de esto, podemos definir los vectores $x, y \in \mathbb{R}^n$ como los vectores de pesos de autoridad y hubs, y se puede suponer que se encuentran normalizados. Evidentemente, se desprende de esto que las páginas con mayores x_i o y_i son consideradas las mejores $autoridades$ o $hubs$.

Se puede expresar la transferencia de pesos de hubs y autoridad, respectivamente, de la siguiente manera:

[6]Kamvar, Haveliwala - 2003 - Extrapolation methods for accelerating PageRank computations

Peso de Autoridad $x_j = \sum_i y_i$

Peso de Hub $y_i = \sum_j x_j$

Consecuentemente, si expresamos con operaciones matriciales los vectores de hub y autoridad:

$$\begin{aligned} x &= Wy \\ y &= W^t x \end{aligned}$$

Se aplica luego el paso de normalización $y \leftarrow \frac{y}{||y||}$ y se vuelve a comenzar hasta llegar a la convergencia. Para la primera iteración es posible empezar con un y_0 inicial cualquiera, y el método convergerá si se cumplen las siguientes condiciones:

- W tiene un autovalor λ_1 tal que $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$
- y_0 no es ortogonal al autovector asociado a λ_1

2.4. In-Deg

Este método es considerablemente más simple y sigue una idea más intuitiva de ranking de páginas sin llegar a entrar en complejizaciones para tener en cuenta distintos casos, lo que consideramos que lleva a unos resultados menos fiables. Esto se debería notar especialmente cuando son contrastados con aquellos que obtenemos de analizar redes con métodos que dan más peso a detalles que potencialmente cambiarían los resultados (por ejemplo: una página apuntada por pocas páginas de gran valor caería muy por debajo en el ranking, mientras que probablemente PageRank y HITS le asignarían una posición más alta).

La idea básica es definir el valor de un página según la cantidad de páginas que la apuntan.

En otras palabras: el método consiste en sumar los links a cada página y, teniendo en cuenta que ya contamos con la matriz W, este método se reduce a sumar la fila correspondiente a cada página y obtener así un vector que nos muestra el valor de cada una (la iésima posición corresponde a la iésima página, según su posición asignada en la matriz W).

2.5. Experimentos realizados

Para los dos primeros métodos, se realizaron experimentos para ver la convergencia de la diferencia en cada iteración de la norma del autovector buscado. En PageRank, se utilizo la norma 1 del autovector x. En el caso de HITS, se analizó los vectores x e y tomando norma 2. Luego, se comparó el tiempo de ejecución de cada uno utilizando la librería time.h de C++. No se tomó en cuenta el tiempo que toma cargar la matriz. Como una iteración de HITS hace dos productos de una matriz por un vector, mientras que PageRank hace uno, y que esta operación es la que domina el costo temporal de la ejecución, se esperaba que el tiempo de ejecución de HITS fuera alrededor del doble del de PageRank, independientemente del valor de c usado. Este influye en el tiempo de ejecución[7] pero no esperábamos que modificara sustancialmente esta relación.

Dado que indexamos la matriz desde 1, reemplazamos cada página numerada como 0 por $n + 1$ para poder realizar los tests.

Las instancias de test usadas fueron las siguientes:

σ	nodos	links	$\frac{link}{nodo}$
Movie	5.757	24.451	4,24
Censorship	2.947	9.555	3,24
Genetic	3.468	12.689	3,65
NotreDame	325.729	1.497.134	4,59
Stanford	281.903	2.312.497	8,20

Los valores de c usados para el pageRank fueron los siguientes:

[7] Kamvar, Haveliwala - 2003 - Extrapolation methods for accelerating PageRank computations

- 0.5
- 0.7
- 0.85
- 0.95

Mientras que la tolerancia para asumir convergencia la fijamos en 10^{-4}

A continuación se experimentó con una muestra de links entre varias páginas obtenida mediante el script provisto por la cátedra. De esta forma esperábamos analizar los resultados obtenidos en función de qué páginas se esperaban con mayor ranking.

3. Resultados

3.1. Convergencia de Norma Manhattan de Pagerank

Estos son los resultados obtenidos de convergencia de la norma para los distintos tests.
En los gráficos de PageRank se compara la evolución de $||x^k - x^{k-1}||_1$ para distintos valores de c.

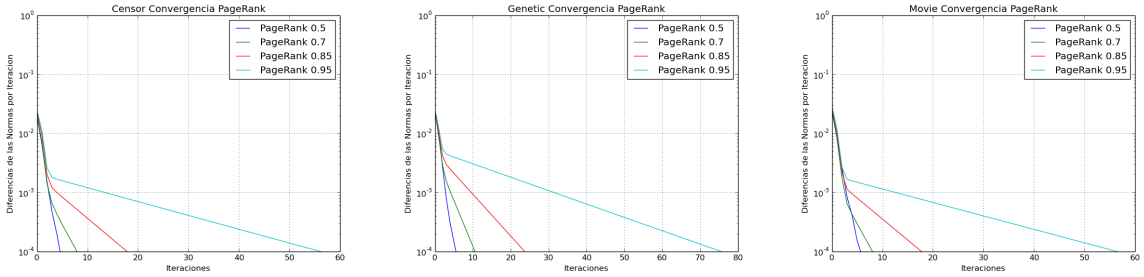

Figura 1: Normas (PageRank): escala logarítmica

	Censorship	**Genetic**	**Movies**
C	Iteración de Llegada	Iteración de Llegada	Iteración de Llegada
0.5	6	7	7
0.7	9	12	10
0.85	19	25	19
0.95	58	77	58

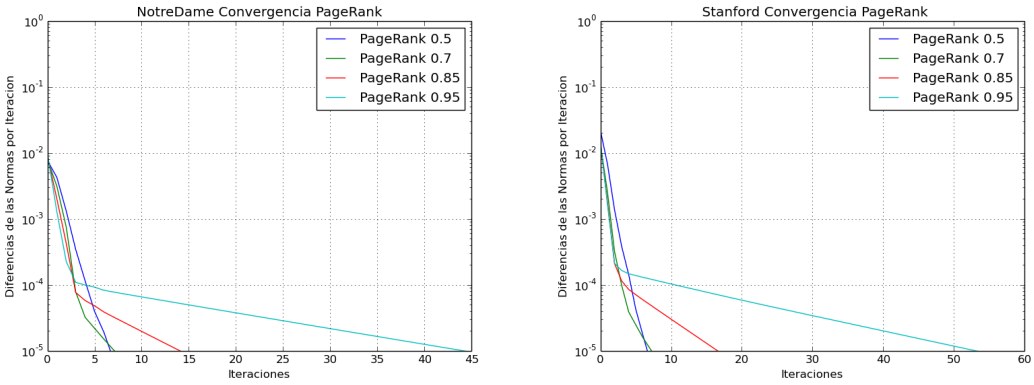

Figura 2: Normas (PageRank): escala logarítmica

	Notredame	**Stanford**
C	Iteración de Llegada	Iteración de Llegada
0.5	8	8
0.7	9	9
0.85	16	18
0.95	46	55

En los gráficos de HITS se compara la evolución de $||x^{(k)} - x_{(k-1)}||_2$ con la de $||y^{(k)} - y^{(k-1)}||_2$.

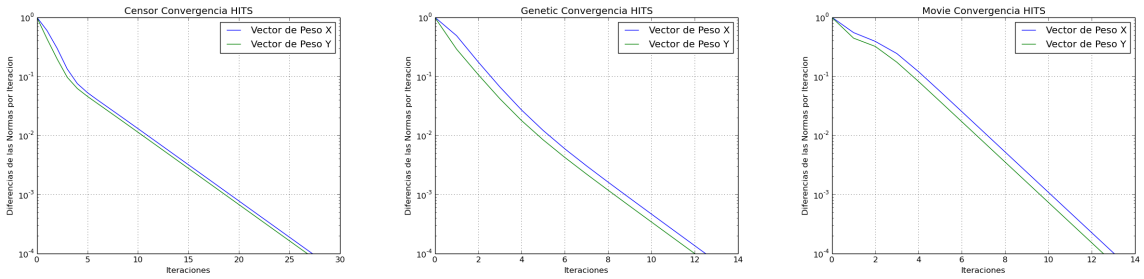

Figura 3: Normas (HITS): escala logarítmica

	Censorship	**Genetic**	**Movies**
Iteración de Llegada	29	14	15

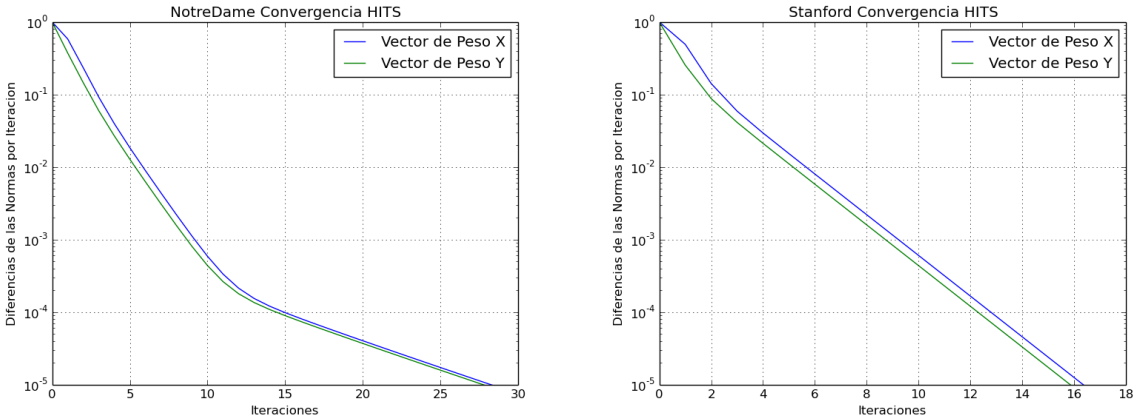

Figura 4: Normas(HITS): escala logarítmica

	NotreDame	**Stanford**
Iteración de Llegada	30	18

Estos son los resultados de tiempo de cómputo obtenidos, en ciclos de clock:

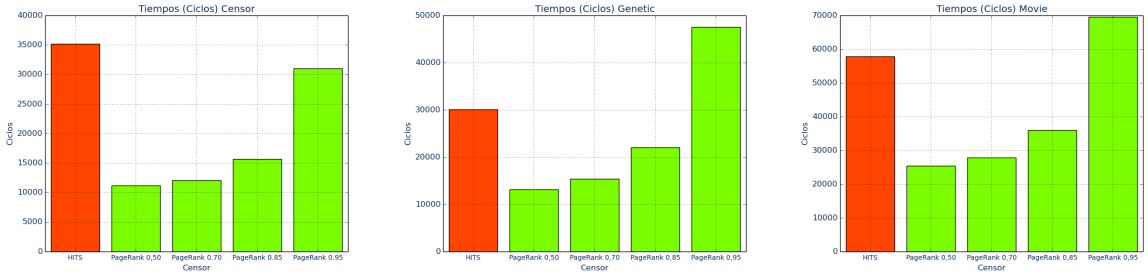

Figura 5: Tiempo de cómputo

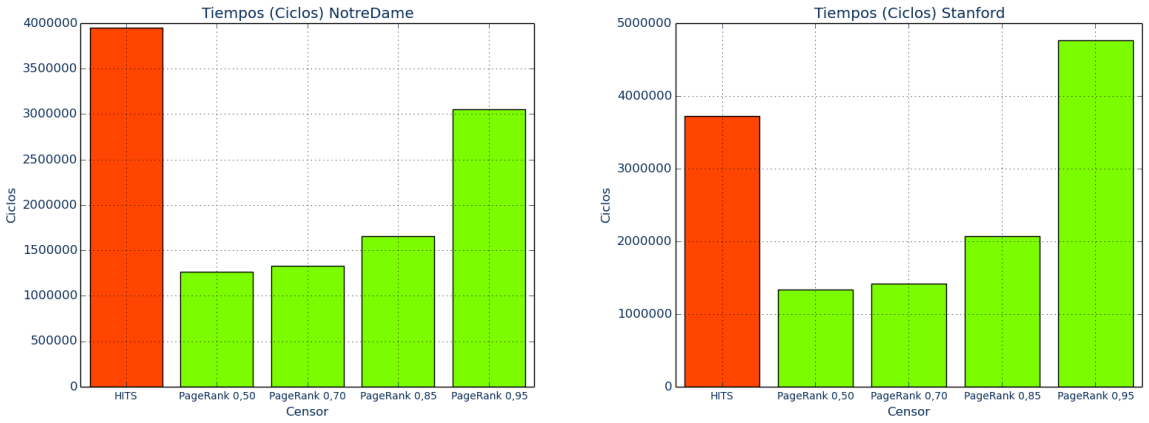

Figura 6: Tiempo de cómputo

	Movie	Censorship	Genetic	NotreDame	Stanford
Ciclos por Iteración PageRank	1896.1	822	882.2	103619.1	115099.8
Ciclos por Iteración HITS	3857.8	1213.4	2149.5	131540.1	206502.1

3.2. Puntajes obtenidos

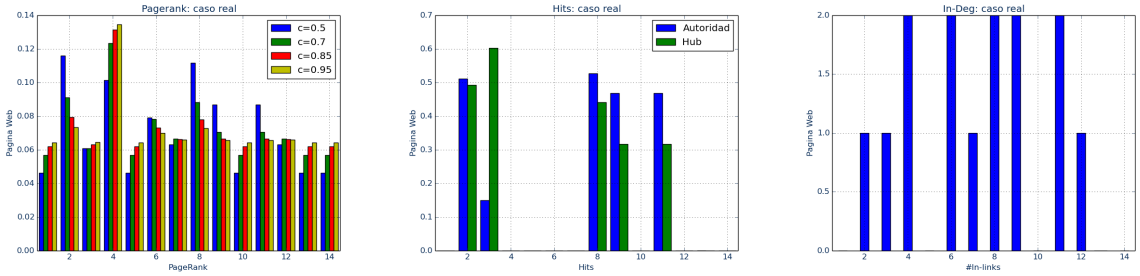

Figura 7: Puntajes obtenidos con los distintos métodos por distintas páginas

1	www.google.com
2	www.clarin.com
3	www.clarin.com/deportes
4	www.lanacion.com.ar
5	www.infobae.com
6	canchallena.lanacion.com.ar
7	www.rollingstone.com.ar
8	www.ole.com.ar
9	www.clasificados.clarin.com
10	www.mamapuntocero.com.ar
11	www.ciudad.com.ar
12	www.zonaprop.com.ar
13	www.pagina12.com.ar
14	www.yahoo.com

3.3. Redes pequeñas

Consideramos los siguientes casos:

- Una red en la que una página apunta a todas

- Una red en la que todas las páginas apuntan a una
- Una red en la que casi todas las páginas se apuntan entre sí
- Una red con muy pocos links

Para observar el comportamiento de PageRank y HITS. Obtuvimos estos resultados:

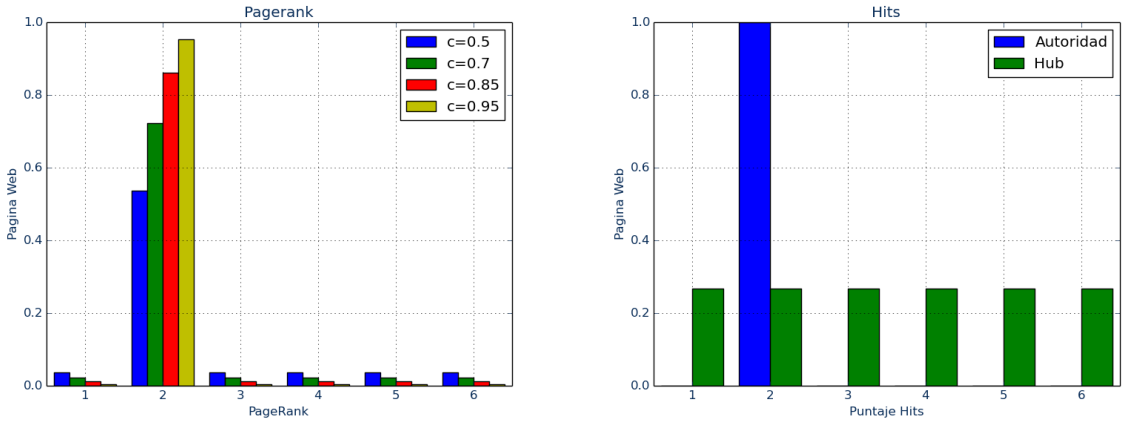

Figura 8: Puntajes obtenidos para una red en la que todas las páginas apuntan a la nº 2

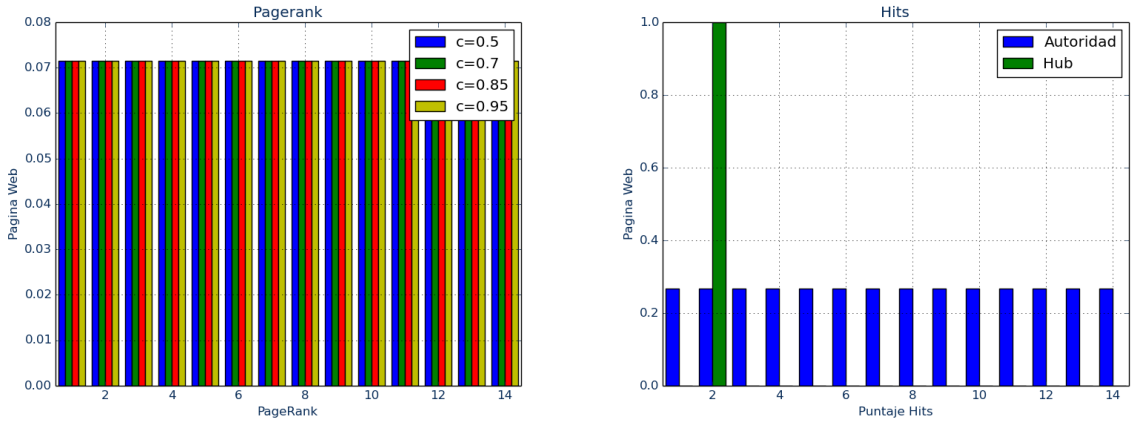

Figura 9: Puntajes obtenidos para una red en la que la pagina nº 2 apunta a todas

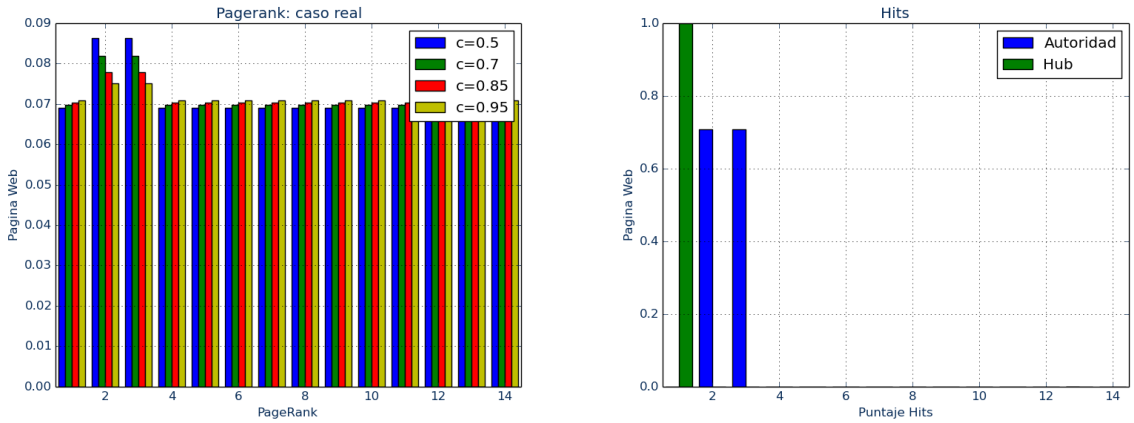

Figura 10: Puntajes obtenidos para una red con pocos links

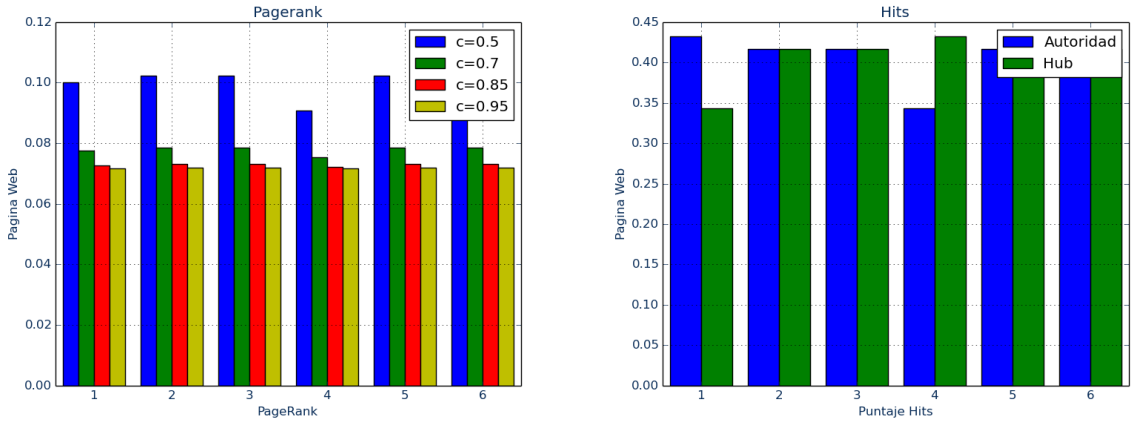

Figura 11: Puntajes obtenidos para una red con muchos links

4. Discusión

En PageRank, las normas convergen a 0 en forma exponencial luego de las primeras iteraciones. Cuanto mayor sea el valor de c, más tarda en alcanzarse el criterio de parada. Como ya se dijo, en un caso extremo, con $c = 1$, la norma podría no converger. En HITS, las normas también convergen a 0 en forma exponencial.

En todos los casos analizados se obtuvo que el tiempo de cómputo de PageRank aumentó exponencialmente con el valor de c.

El algoritmo HITS resultó más lento que PageRank para todos los valores de $c \leq 0,85$ que utilizamos, en todas las redes para las que medimos los tiempos. Sin embargo, consideramos que esta diferencia se debe principalmente a que el algoritmo HITS debe realizar dos productos entre una matriz y un vector, mientras que PageRank solo realiza una. Se puede comprobar que esta diferencia es proporcional al tamaño de la matriz que representa a la red. Es decir, la diferencia no llega a ser de órdenes de complejidad.

Además, se espera que si se eligen valores de c mayores a un valor, el tiempo de ejecución de Pagerank siempre supere al de HITS.

En los tests de mayor tamaño se puede ver que la cantidad de iteraciones necesarias para alcanzar convergencia es igual que en los tests menores o incluso menor. Si tienen mayor tiempo de cómputo, es porque la multiplicación entre la matriz y el vector que es necesaria en cada iteración tiene una mayor cantidad de elementos. Esto se puede ver en el tiempo de iteración. Sin embargo, dada la proporción de links por páginas, esto tampoco influye demasiado.

El valor de c, si bien influye en el puntaje, no lo varía demasiado. Un valor más alto de c implica darle una mayor importancia a los links que relacionan las páginas, resultando en un ranking más representativo de la estructura de la red. Sin embargo, no necesariamente será mejor como modelo para el comportamiento de un navegante aleatorio en la práctica.

Teniendo en cuenta que la cantidad de iteraciones, y por lo tanto el tiempo de ejecución, escalan exponencialmente con este parámetro, es conveniente elegir un valor no demasiado alto.

Al hacer un análisis cualitativo de los puntajes obtenidos se pueden obvervar que el puntaje no es el esperado. Por ejemplo, se esperaba que Google obtuviera el mayor puntaje de hub, cosa que no ocurrió.

Esto se debe a que la porción de la red elegida no es representativa. Se ve que lanacion.com obtuvo el mayor puntaje de PageRank, pero hay que tener en cuenta que en la lista hay sitios que pertenecen al mismo dominio.

5. Conclusiones

En el paper escrito por Brin y Page [8] se recomienda tomar un c (probabilidad de un navegante aleatorio de seguir un link de la página en la que está) de 0.85.

Consideramos sin embargo que, de contar con los recursos necesarios para realizar mediciones de la magnitud necesaria para obtener resultados representativos, la mejor manera de definir un valor de c fiable en la mayor cantidad de casos posibles sería acercarnos a la probabilidad real con la que el navegante aleatorio $salta$ a otra página sin seguir un link, para esto haría falta hacer experimentos con navegantes reales y realizar un promedio de los c que obtuvimos de estos navegantes; es decir que el mejor c a tomar sería la esperanza de los c obtenidos en estas mediciones.

El hecho de que la norma converja exponencialmente permite que la cantidad de iteraciones sea lo suficientemente acotada, teniendo en cuenta que cada iteración es una multiplicación de una matriz por un vector que puede tener varios millones de posiciones.
Luego de haber contrastado estos distintos métodos de ranking podemos decir con seguridad que:

In-Deg es completamente descartable como método para decdidir la relevancia de una página, se limita a realizar el análisis más previsible y propenso a caer en errores. Buscadores puramente textuales que siguieron este método, como AltaVista, fueron reemplazados eventualmente por otros más útiles, si bien fueron un paso necesario para la creación de buscadores mejores.

HITS es mucho más efectivo y confiable ya que realiza un análisis más inteligente de la red que le es proporcionada, confiriéndole distintos roles a las páginas y contrastándolas a partir de las relaciones entre estas páginas según el rol que estén cumpliendo.
Hemos comprobado que la cantidad de links a una página no son relevantes por sí solos para considerarla una buena autoridad sino que lo más importante es la relevancia como hubs de aquellas páginas que la apuntan, esto revela un análisis mucho más profundo que el de *Indeg*.
Por la naturaleza del análisis que realiza es, sin embargo, más susceptible de caer en engaños tales como que páginas acuerden apuntarse entre sí para ganar peso en el ranking que *HITS* les asigne, mientras que con *PageRank* este artilugio no daría frutos debido a que cuantos más links salientes una página tenga, menos valor le estará transfiriendo con cada link a las páginas apuntadas.

PageRank es el método que consideramos mejor, superando tanto a *HITS* como a *Indeg* debido a que realiza un análisis profundo de las relaciones entre páginas, distinto de aquel proporcionado por *HITS*. En ciertos casos parece que el procedimiento es opuesto. Por ejemplo, una página que apunta a todas le da poco valor a cada una en PageRank, pero al apuntar a muchas autoridades es un buen hub según HITS, luego le da valor a los que apunta. Un inconveniente que tiene este algoritmo es que trata una pagina con links a toda la web igual que un dangling node, mientras que HITS le da a un gran puntaje como hub.

Por ultimo, si el cliente quiere tener un mejor PageRank, debe conseguir que lo apunten páginas que tengan la mejor proporción entre PageRank y cantidad de links salientes, ya que el puntaje se distribuye entre todas las páginas. En cambio, si quiere conseguir un mejor puntaje como autoridad en HITS, debe buscar que la mayor cantidad posible de las mejores páginas lo apunten. Si desea tener un mejor puntaje de hub tendría que apuntar a las páginas con mayor autoridad en HITS.
Podemos finalmente concluir que la mejor estrategia a sugerir a clientes sería darle prioridad al ranking proporcionado por Pagerank, está además la prueba de la efectividad del Pagerank en el éxito que tuvo Google al usarlo. Sugeriríamos entonces apoyarse en el orden de resultados que proporcione este

[8] Brin, Page - 1998 - The anatomy of a large-scale hypertextual Web search engine

buscador (que es además el más usado) para decidir en qué páginas comprar espacio de publicidad.

6. Apéndice A

Tirate un qué, tirate un *ranking*...

Motivación

Luego de su repentina y efímera irrupción durante el año 2011, un grupo de la movida tropical[9] 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.[10]

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*. Esta, 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 distnguió 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 explicando en Brin y Page [**?**], 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.

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

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

[10] 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.

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

$$x_k = \sum_{j \in L_k} \frac{x_j}{n_j}, \quad k = 1, \ldots, n. \tag{2}$$

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 (2) 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 [**?**] y Kamvar et al. [**?**, 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 (2). 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 [**?**] y Kamvar et al. [**?**], 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 [**?**], 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.[11] 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 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:

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

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

$$y_i = \sum_{j:i \to j} x_j. \tag{4}$$

Las ecuaciones (3) y (4) podemos expresarlas matricialmente de la siguiente manera:

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

$$y = Ax, \tag{6}$$

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 INDEG, 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 [**?**] con los detalles sobre HITS, en particular las secciones 1, 2 y 3.

2. Considerar el trabajo de Bryan y Leise [**?**] donde se explica la intución y algunos detalles técnicos respecto a PageRank. Además, en Kamvar et al. [**?**] 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. [**?**, 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.

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

3. (Opcional) Completar la demostración del Teorema 3.1 de Kleinberg [**?**], 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 (5) y (6). 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[12]. 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

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 [**?**] (el tipo de

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

instancia es 0), con redes de tamaño grande obtenidos a partir de datos reales. Además, se consideran las instancias propuestas en [**?**]. 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 [**?**, 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 [**?**] 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.