DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico II

Tirate un qué, tirate un ranking...

Métodos Numéricos
Segundo Cuatrimestre de 2014

Integrante	LU	Correo electrónico
Aldasoro Agustina	86/13	agusaldasoro@gmail.com
Bouzón María Belén	128/13	belenbouzon@hotmail.com
Cairo Gustavo Juan	89/13	gjcairo@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

Habiéndonos siendo dada la problemática de cómo posicionar una página web alta en una búsqueda online, utilizando tres métodos de ordenamiento de páginas basados en links, este trabajo desarrolla distintas experimentaciones para la observación del comportamiento de los métodos y de este modo saber cuál sería la tarea a llevar a cabo.

Palabras claves

- Matriz Esparsa
- PageRank
- HITS
- In-deg

Índice

1. Introducción Teórica

A la hora de diseñar un Motor de Búsqueda hay varios aspectos a tener en cuenta, tales como: contar con acceso a las páginas disponibles en la red, tener una base de datos donde almacenarlas e indexarlas para su procesamiento posterior y ser capaces de ordenarlas de acuerdo a su importancia relativa dentro de dicha red. Nuestro trabajo se centrará en este último aspecto.

Existen varios métodos que priorizan distintas características de las relaciones entre las páginas para dar cierto orden a una red a partir de determinada búsqueda. Un criterio válido consiste en situar en una posición de mayor jerarquía a aquellas páginas que contengan mayor cantidad de coincidencias textuales con el concepto consultado. Éste podría no resultar óptimo en ciertos casos. Por ejemplo, si se buscara el string *"Red Social"*, intuitivamente se esperaría que entre los principales representantes de esta búsqueda se encontraran determinadas sitios web tales como Facebook o Twitter. Sin embargo, la cantidad de veces que estas páginas contienen al string *"Red Social"* puede no ser significativa y esto provocaría que no aparecieran en los primeros lugares las páginas genuinamente más vinculadas al concepto buscado.

Todos los métodos a desarrollar en este trabajo partirán del registro, la comparación y el análisis de los Links Salientes/Entrantes de la Red provista para ponderar el valor relativo de cada sitio en dicho sistema.

1.1. PageRank, HITS, In-deg

El trabajo consistirá en el estudio de distintos aspectos de los siguientes métodos: PageRank, HITS e In-deg. Los mismos se detallan a continuación:

PageRank *- Modelo del Navegante Aleatorio*

Este método consta de tres fases: exploración de la web y localización todas las páginas de acceso público; indexado de los datos desde el primer paso, de manera que se pueda acceder eficientemente a palabras claves o frases relavantes; y valoración de la importancia de cada una de las páginas en la base de datos. A nivel de nuestro desarrollo, sólo nos encargaremos de la última de las etapas mencionadas.

Teniendo un grafo dirigido, se le otorga a cada componente X_k del mismo un valor dado por la siguiente ecuación:

$$X_k = \sum_{j \epsilon L_k} \frac{X_j}{n_j}$$

Donde L_k es el conjunto de links entrantes a la página k y n_j es el número de links salientes desde la página j.

Luego, se construye una matriz A donde se encuentran -por filas- las respectivas ecuaciones para cada X_i, definidas como fue realizado anteriomente.

La resolución de este método se logra al hallar el autovector con autovalor asociado 1 para la matriz A. De acuerdo al trabajo de Bryan y Leise [4], este cálculo se computa mediante el método de la potencia.

Dicha matriz cuenta con ciertas mejoras que proporcionan ventajas en casos específicos. Por un lado, si alguna página p no tuviera ningún link saliente se considera que el navegante aleatorio saltará con equiprobabilidad a cualquiera de las otras páginas. De esta forma, se le otorga al vector columna p de la matriz A el valor de $\frac{1}{n}$ para cada componente. Por otro lado, existe un fenómeno denominado *"Teletransportación"* que considera la posibilidad de que dicho navegante se mueva de una página a otra pero no mediante los links existentes, sino tipeando la URL. Para modelar de manera óptima este suceso, se reemplaza a la matriz A por la matriz M definida bajo la siguiente ecuación: *M = (1-m)A + m.S* siendo m la probabilidad de que un navegante se *teletransporte* y S una matriz cuyos valores S_{ij} tienen todos el mismo valor: $\frac{1}{n}$ representando así una matriz donde la probabilidad de ir a cualquier página del grafo es uniforme.

El *Método de la Potencia* se realiza de manera iterativa, lo cual permite reducir el tiempo de cómputo para elevar a la k la matriz M. Si tenemos en cuenta el trabajo de Kamvar [5], presenta una herramienta de cálculo que permite encontrar el principal autovector de M en una serie menor de pasos modificando la ecuación de la matriz M.

Hyperlink-Induced Topic Search (HITS)

El método planteado por Kleinberg [6] consiste en partir de una consulta sobre σ y focalizarse en una colección de páginas S_σ tal que sea relativamente pequeña, rica en páginas relevantes sobre el tema y contenga la mayoría de las autoridades más fuertes sobre el mismo. Considerando autoridad a una página que tiene la mayor cantidad de links entrantes provenientes de páginas vinculadas al tema, esto se realiza del siguiente modo:
a) Acorde a un parámetro *t*, se coleccionan las primeras t páginas rankeadas bajo una búsqueda basada estrictamente por texto. A este conjunto se lo denomina R_σ.
b) Se incrementa el conjunto R_σ añadiendo las páginas que tienen links entrantes y salientes al mismo, formando así el conjunto S_σ. Para cada página de R_σ se permite añadir, a lo sumo *d* páginas que la apunten y *d* páginas a las cuales apunte.
c) Se eliminan de S_σ los links intrínsecos, es decir no se tienen en cuenta links que apuntan a una página del mismo dominio que la página saliente.
d) Se admiten hasta *m* páginas del mismo dominio apuntar a cualquier página p. Esta idea no fue utilizada por el autor.

El conjunto obtenido hasta esta instancia lo llamamos G_σ. Nuestro trabajo asume un conjunto G_σ bien formado.

Se construye una matriz de adyacencia que denominaremos A, bajo la siguiente fórmula:

$$a_{ij} = \begin{cases} 1 & \exists \textit{ link desde i hasta j} \\ 0 & \textit{caso contrario} \end{cases}$$

A cada página i de la Web se le otorga un peso como Autoridad y un peso de Hub:
Peso de autoridad:

$$X_j = \sum_{i:i\to j} Y_i$$

Peso de Hub:

$$Y_i = \sum_{j:i\to j} X_j$$

Este algoritmo devuelve dos arreglos: uno representa los pesos de Hub y otro los pesos de Autoridad, teniendo una coordenada para cada página perteneciente al conjunto G_σ.

In-deg

Consiste en definir el ranking de las páginas utilizando sólo la cantidad de ejes entrantes a cada una de ellas, ordenándolos en forma decreciente.

2. Desarrollo

2.1. Elección de las estructuras

Con el fin de elegir la estructura que representaría nuestra matriz esparsa, estudiamos tres tipos proporcionados por la cátedra: *Dictionary of Keys* (DOK), *Compressed Sparse Row* (CSR) y *Compressed Sparse Column* (CSC).

La primera consiste en un diccionario con doble clave (fila y columna) y su significado son los elementos de la matriz distintos de cero. De esta manera se saca provecho de la cantidad de elementos nulos de la matriz, garantizando una optimización en términos de espacio en memoria. Además esta implementación cuenta con la gran ventaja de que resulta simple construirla incrementalmente en un arreglo esparso y además puede ser traspuesta de manera sencilla (inviertiendo el orden de las claves). Sin embargo, el principal inconveniente reside en la necesidad de convertirla a otro formato para procesar los cálculos aritméticos. A causa de esto fue descartada la opción.

El modo de almacenamiento *Compressed Sparse Row* requiere la implementacion de tres arreglos (en nuestro caso vectores) que llamaremos val, ind_col y ptr_fila. El tamaño de los dos primeros está dado por la cantidad de elementos no nulos de la matriz. Mientras que el primero (val) almacena estos valores de izquierda a derecha y luego desde arriba hacia abajo, el segundo vector (ind_col) indica el número de columna para cada elemento. En otras palabras, el elemento almacenado en la posición i- ésima del vector ind_col representa la columna correspondiente al valor almanacenado en val_i. Por último el tercer vector (ptr_fila) tiene un tamaño equivalente a la cantidad de filas incrementada en uno, conteniendo los índices del comienzo de cada fila.

El modo de almacenamiento *Compressed Sparse Column* cuenta también con la implementación de tres arreglos llamados: val, ind_fila, ptr_col. El primero contiene todos los valores distintos de cero de la matriz, desde arriba hacia abajo y luego de izquierda a derecha. Ind_fila son los índices de fila correspondientes a dichos valores. Por último, ptr_col lista los índices donde comienza cada columna.

Por último, el tamaño del vector ptr_fila se encuentra determinado por la cantidad de filas incrementada en uno, y lista los índices que indican los valores de val que comienzan cada fila.

Ante a la disyuntiva acerca de cuál de estos últimos formatos escoger (CSR o CSC) decidimos realizar una serie de cálculos pequeños que nos permitieron notar que si nos situábamos en el formato de *Compressed Spare*, trasponer una matriz almacenada de manera CSC no sería más que interpretar sus mismos arreglos como CSC. Se incluye un ejemplo en *Apéndice C*. Fue decisión del grupo considerar el formato por defecto de la matriz el CSR (filas) y al trasponerlas sólo modificarle un bool que indique si está traspuesta y leerla y considerarla en adelante como CSC (columnas). Esta decisión fue tomada luego de que la cátedra nos confirmara que estaba permitido elegir una opción de las ofrecidas y adaptarla a nuestra provecho, siempre que se aclararan los cambios. Por este motivo, en el algoritmo de multiplicar una matriz por un vector se diferencia la manera en que la misma se encuentre almacenada y se obtiene el producto acorde a su formato. Se incluye el pseudocódigo de este algoritmo en la Subsección *Algoritmo multiplicación de una matriz por un vector*.

2.2. Algoritmo multiplicación de una matriz por un vector

Considerando la estructura elegida, nos vemos obligados a diferenciar dentro del algoritmo de multiplicación de una matriz por un vector de acuerdo al modo en que debe ser leído (CSR/CSC).

Para computar el cálculo de una matriz por un vector interpretándolo bajo la estructura *Compressed Sparse Row* se utilizó el siguiente algoritmo:

```
input  : Matriz m, Vector v
output: Vector res
for i ← 0 to cantidad de filas do
    inicio ← m.ptr_fil[i]
    fin ← m.ptr_fil[i+1]
    for j ← inicio to fin do
        col ← m.ind_col[j]
        res[i] ← res[i] + (m.val[j] * v[col])
    end
end
```

Se recorre el vector ptr_fil de la matriz, el cual indica en qué índice comienza cada fila. Para cada elemento de la fila actual, se asigna en el int col el número de columna correspondiente; y luego, se multiplica ese elemento con el correspondiente del vector v (v[columna actual]) y se suma en res[i], siendo i la fila actual.

Para computar el cálculo de una matriz por un vector leyéndolo bajo la estructura *Compressed Sparse Column* se utilizó el siguiente algoritmo:

```
input  : Matriz m, Vector v
output: Vector res
for i ← 0 to cantidad de filas do
    inicio ← m.ptr_fil[i]
    fin ← m.ptr_fil[i+1]
    for j ← inicio to fin do
        fil ← m.ind_col[j]
        res[col] ← res[fil] + (m.val[j] * v[i])
    end
end
```

Se recorre el vector ptr_fil de la matriz, el cual indica en qué índice comienza cada columna. Para cada elemento de la columna actual, se asigna en el int fil el número de fila correspondiente; y luego, se multiplica ese elemento con el correspondiente del vector v (v[fila Actual]) y se suma en res[fil].

2.3. Algoritmo de HITS

Nuestra tarea aquí es extraer del subconjunto G_σ sus autoridades analizando puramente la estructura de sus links. Ordenar las páginas, dándoles un puntaje de acuerdo a la cantidad de links de entrantes, trabaja mejor bajo el contexto de nuestro subconjunto, de todos modos un ranking de este tipo carece de una unidad temática. Las páginas con mayor puntaje de autoridad no solo tienen una cantidad significante de nodos entrantes sino que también van a tener muchas páginas en común que las apunten.

Hubs y Autoridades reprensentan una relacion de mutua dependencia, frente a esto es necesario un método para solucionar este ciclo como el siguiente:

input : Matriz m, double tol
output: Vector x, Vector y

Inicializar vectores *x* e *y* con 1 en todas sus posiciones
Vector xp, yp
while *(No se haya llegado a la cantidad máxima de iteraciones i)* **do**
  m.$trasponer()$
  xp ← a.$multMatVect(y)$
  xp.$normalizar()$
  m.$trasponer()$
  yp ← a.$multMatVect(x)$
  yp.$normalizar()$
  if *(xp* $\simeq$ *x* $\wedge$ *yp* $\simeq$ *y)* **then**
    i ← $Máxima\ Iteración$
  else
    i++
  end
  x ← xp
  y ← yp
end
print *x* e *y*

∘Este algoritmo devuelve los valores de Autoridad y Hub para todas las páginas, en X e Y respectivamente.

∘Los vectores X e Y arrancan inicializados en 1 porque PINTO (??? GUSSSSSSSSS).

∘El $\simeq$ considera la tolerancia (tol) pasada por parametro. Es decir, es equivalente a evaluar abs(x-xp)$\leq$tol.

∘La cantidad máxima de iteraciones la fijamos nosotros en 100.000 pero al existir la guarda del if cabe la posibilidad de salir del scope del while antes de las 100.000 iteraciones.

El modelo de cálculo es un algoritmo iterativo, el cual conserva y actualiza los pesos de Hub y de Autoridad para cada página. Se cuenta con dos vectores de un tamaño igual a la cantidad de nodos en la red, donde los pesos de autoridad de la página i se pueden ver en la posición i del vector X, mientras que los de Hub se encuentran en la posición i del vector Y. Ambos vectores son normalizados -bajo la Norma 2- en cada iteración. De este modo, las páginas con mayor valor en X son "mejores" autoridades y las que tengan mayor valor en Y son "mejores" Hubs.

Lo que se hace en cada iteración es actualizar primero los valores de X en base a los de Y, y luego actualizar los de Y en base a los nuevos de X. Se tiene una matriz A -*Matriz de adyacencia*- la cual posee un 1 en A(i,j) si existe un link i $\rightarrow$ j y un 0 en caso contrario. De este modo, trasponiéndola se puede observar la relación inversa. Esto explica que a la hora de actualizar los valores de X e Y se puede realizar asignando X $\leftarrow$ A^tY, Y $\leftarrow$ AX; acorde a lo visto en la *Introducción Teórica* sobre el *Algoritmo de HITS* se adaptan las ecuaciones a la forma matricial.

Basándonos en el trabajo de Kleinberg [6], podemos asegurar que este método converge bajo ciertas hipótesis (ESTOY MANDANDO FRUTA?) como que el grafo G_σ sea un grafo bien formado y que la matriz A tiene como principal autovector un valor único (CREO QUE NADA MAS. PREGUNTAR A GUS.). En el

mismo trabajo se puede ver la demostración de que los vectores X e Y convergen a X* e Y*, y además X* es el principal autovector de A^tA e Y* es el principal autovector de AA^t.

Si bien nosotros optamos por devolver todos los puntajes de Hub y de Autoridad, al momento de emplearlo en una búsqueda de páginas web se sitúan primero las k mayores Autoridades y luego los k mayores Hubs.

2.4. Algoritmo de PageRank

En este esquema, cada página se define como la suma de los valores de las páginas con links entrantes divididos por su cantidad de enlaces salientes (modelo visto en la Introducción teórica). La matriz a abordar se arma por columnas, para cada columna X_i se pone un 0 en los elementos que no los dirija ningún link y $1/k$ en los demás, siendo k la cantidad de links salientes de la página X_i. Algo debe hacerse para solventar el problema de las páginas con dangling nodes (páginas sin ningún link saliente).

Al contar con la matriz A -matriz de conectividad-, nuestro problema a resolver se limita a encontrar un autovector de A con autovalor 1. En el trabajo de Bryan y Leise [4] se demuestra que: Si la Web con la que trabajamos no presenta ningún danglin node, la matriz de conectividad A tiene al 1 como autovalor mediante las siguientes preposiciones: *La matriz A para alguna Web sin dangling nodes es estocástica por columnas* (una matriz cuadrada es estocástica por columnas si todos sus valores son no-negativos y los elementos de cada columna suman 1) y *Toda matriz estocástica por columnas tiene al 1 como autovalor*. Para asegurar la unicidad del ranking a armar es concluyente exigir que autovalor 1 tenga multiplicidad 1.

Pero nosotros vamos a reemplazar A por M, la cual está definida por la fórmula M = (1-m)A + mS, con m la probabilidad de moverme entre páginas no por medio de links y S una matriz con todos sus componentes iguales a 1/n. Donde así no debe efectuarse ninguna hipótesis fuerte sobre A ya que M queda estocástica por columnas y positiva la multiplicidad de λ_1 es 1.

A fines de emplear menos recursos de cómputo se optó por llevar a cabo el algoritmo desarrollado por Kamvar [5]:

input : Matriz m, double c, double tol
output: x

Se inicializa el vector x en todos unos
Vector xp
while *(No se haya llegado a la cantidad máxima de iteraciones i)* **do**
  xp ← m.$MultMatVec(x)$
  for $i \leftarrow 0$ **to** $tamaño\ de\ xp$ **do**
    xp[i] ← xp[i] * c
  end
  for $i \leftarrow 0$ **to** $tamaño\ de\ xp$ **do**
    xp[i] ← $xp[i] + \frac{norma(x)-norma(xp)}{n}$
  end
  if *(xp ≃ x)* **then**
    i ← *Máxima Iteración*
  else
    i++
  end
  x ← xp
end
normalizar el vector *x*
print *x*

○Este algoritmo devuelve el vector X el cual tiene Norma1 = 1 y representa para cada X_i el porcentaje de tiempo que el *Navegante Aleatorio* permanece en cada página i.

○En este algoritmo no hace falta normalizar en cada iteración ya que conserva la norma.

○El ≃ considera la tolerancia (tol) pasada por parametro. Es decir, es equivalente a evaluar abs(x-xp)≤tol.

○La cantidad máxima de iteraciones la fijamos nosotros en 1.000.000 pero al existir la guarda del if cabe la posibilidad de salir del scope del while antes del 1.000.000 de iteraciones.

○ el parámetro C de entrada corresponde a la probabilidad de *teletransportarse*.

Lo que se calcula en este algoritmo es $X_k = MX_{k+1}$ mediante el método de la potencia pero no se ejecuta la multiplicación de matrices sino que se hacen todos cálculos con vectores.

3. Resultados y discusión

3.1. Convergencia de PageRank

INCLUIR EXPERIMENTOS SOBRE LA PROPORCION DEL LAMBDA. Cuanto mas se acerca a 1 (xq lambda uno es uno) mas tarda en converger.

Estudio de la convergencia de PageRank, analizando la evolución de la norma Manhattan.
Hipótesis: Suponiendo que el método de la potencia converge (dado que la probabilidad de que no ocurra es minima), consideramos que la norma Manhattan de la diferencia entre iteraciones sucesivas debe converger a cero. Creemos que cuanto mayor sea el valor de c, más va a tardar en converger.

Nuestra hipótesis se basó en la ecuación $P_2 = cP_1 + (1 - c)E$, la cual hace que los datos cobren sentido. Dado un c, el segundo término queda constante; por lo tanto, podemos considerarlo despreciable para nuestros cálculos, restringiéndonos a analizar lo que pueda suceder sólo con el primero. cP_1 implica la multiplicación de una matriz por un valor c tal que $0 \leq c \leq 1$. No resulta relevante considerar los casos en que $c = 0$ ó $c = 1$, dado que esos valores no modelarían un aspecto de la realidad. Esto se debe a que si $c = 0$, la probabilidad de teletransportarse es 1 por lo tanto se moldearía una realidad donde nunca se viaja a través de Links. Y por otro lado, si $c = 1$ no se estaría considerando la *teletransportación*.

Dado que c se encuentra entre 0 y 1, el resultado va a ser un vector de valores muy pequeños, que va a achicarse progresivamente. Es más significativo lo que achica el c a P_2 que lo que aporta el segundo término.

Los siguientes gráficos están citados en orden creciente de tamaño de las redes:

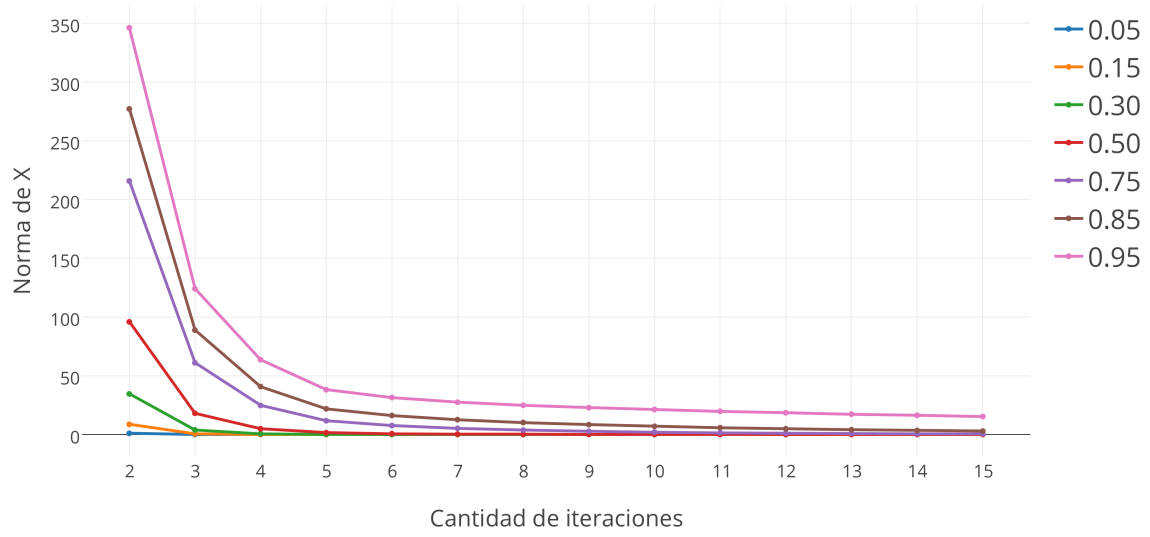


Figura 1: Variación de la norma cambiando el C para la Red Complexity

En la siguiente tabla se muestran la cantidad de iteraciones necesarias para distintos C bajo la misma Red:

	C=0.05	C=0.15	C=0.30	C=0.50	C=0.75	C=0.85	C=0.95
Cantidad de Iteraciones	5	8	12	21	49	86	268

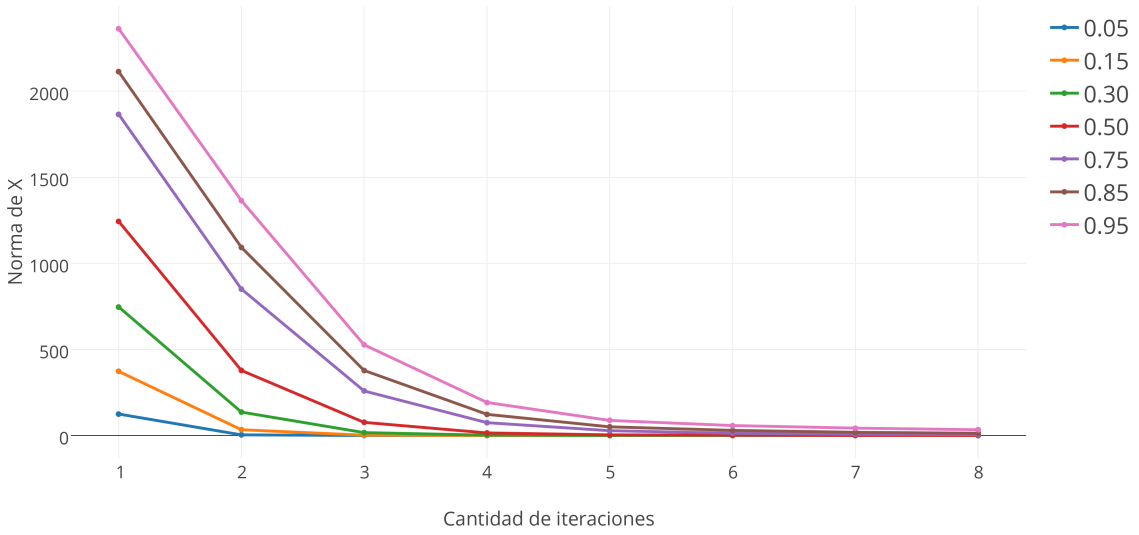


Figura 2: Variación de la norma cambiando el C para la Red Abortion

En la siguiente tabla se muestran la cantidad de iteraciones necesarias para distintos C bajo la misma Red:

	C=0.05	C=0.15	C=0.30	C=0.50	C=0.75	C=0.85	C=0.95
Cantidad de Iteraciones	5	7	11	19	46	80	251

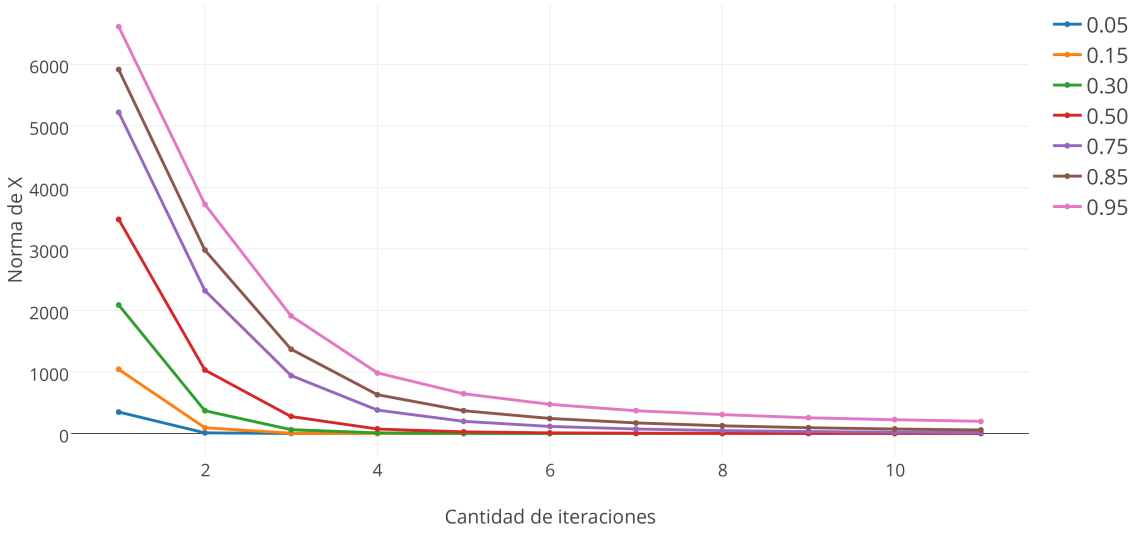


Figura 3: Variación de la norma cambiando el C para la Red Movies

En la siguiente tabla se muestran la cantidad de iteraciones necesarias para distintos C bajo la misma Red:

	C=0.05	C=0.15	C=0.30	C=0.50	C=0.75	C=0.85	C=0.95
Cantidad de Iteraciones	6	8	13	21	51	90	282

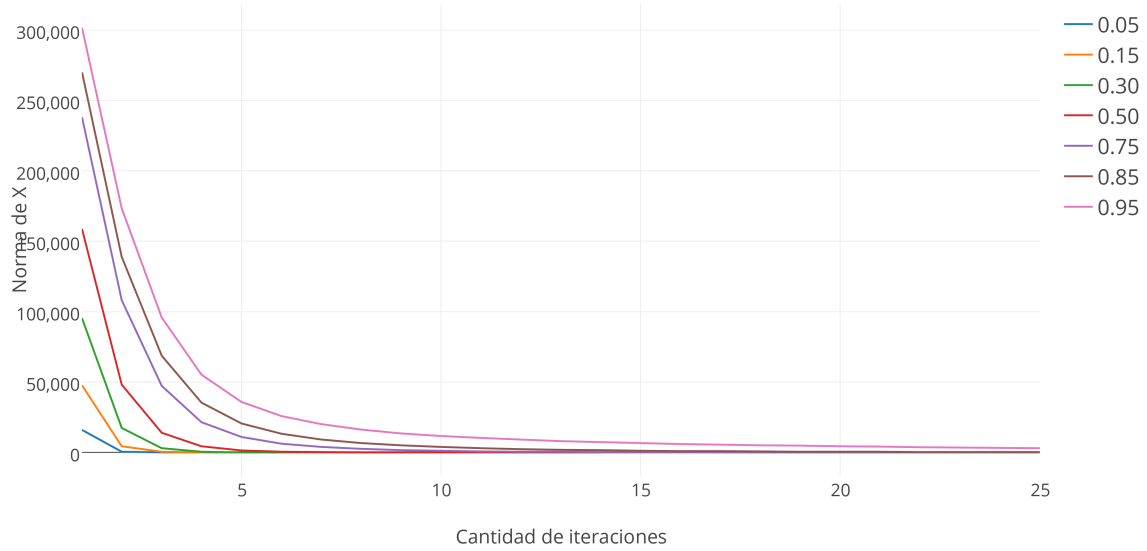


Figura 4: Variación de la norma cambiando el C para la Red Stanford

En la siguiente tabla se muestran la cantidad de iteraciones necesarias para distintos C bajo la misma Red:

	C=0.05	C=0.15	C=0.30	C=0.50	C=0.75	C=0.85	C=0.95
Cantidad de Iteraciones	6	9	14	25	59	104	329

En la siguiente tabla se muestran la cantidad de iteraciones necesarias para distintos C bajo la mis-

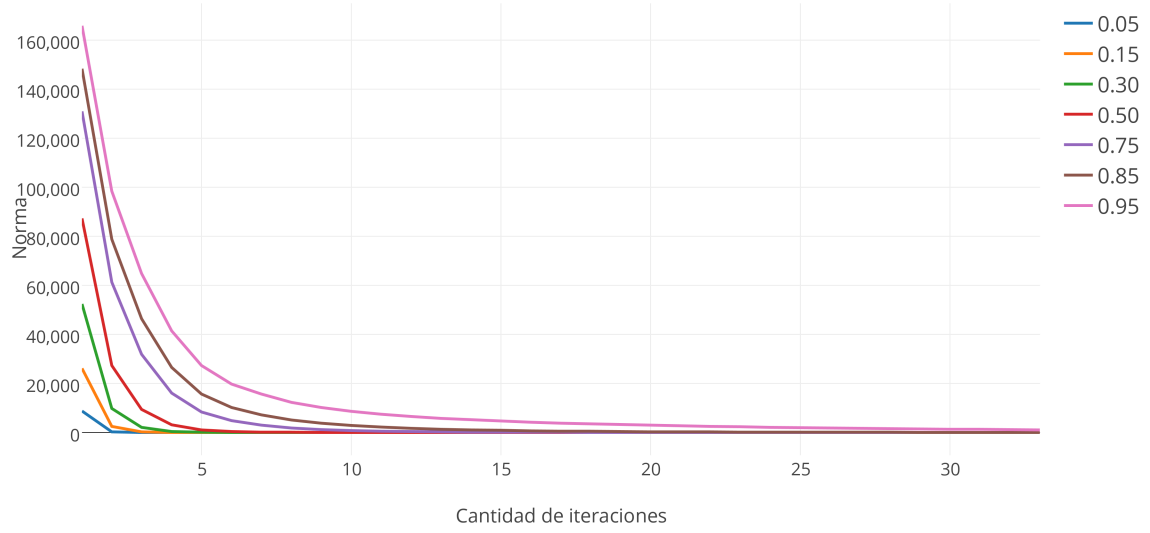


Figura 5: Variación de la norma cambiando el C para la Red Notre Dame

ma Red:

	C=0.05	C=0.15	C=0.30	C=0.50	C=0.75	C=0.85	C=0.95
Cantidad de Iteraciones	6	10	15	25	56	96	295

Experimentamos con Redes de distintos tamaños y podemos afimar que para todas ellas: *Cuanto mayor es el C, se necesita una mayor cantidad de iteraciones para concluir algorítmicamente que ya convergió el vector* x *a* $x*$. Es decir, que se aumenta la cantidad de veces que se ejecuta el ciclo hasta que la norma de la diferencia entre X-actual y X-anterior sea menor a la tolerancia. Lo cual confirma nuestra hipótesis.

3.2. Convergencia de HITS

Hipótesis: Nuestro planteo es que la norma 1 de la diferencia entre dos iteraciones consecutivas sobre el vector X y el vector Y va a converger al valor 0.

Esta idea surge de que el algoritmo implementado por HITS sea circular, es decir para actualizar los valores del vector X se utilizan los valores de Y y viceversa. Por este motivo, se podría decir que se "trasladan" los pesos de X e Y al multiplicar cada vector por la matriz, y como en cada iteración se normalizan los vectores esto lleva a que los valores a trabajar sean menores de iteración a iteración.

Los siguientes gráficos están citados en orden creciente de tamaño de las redes:

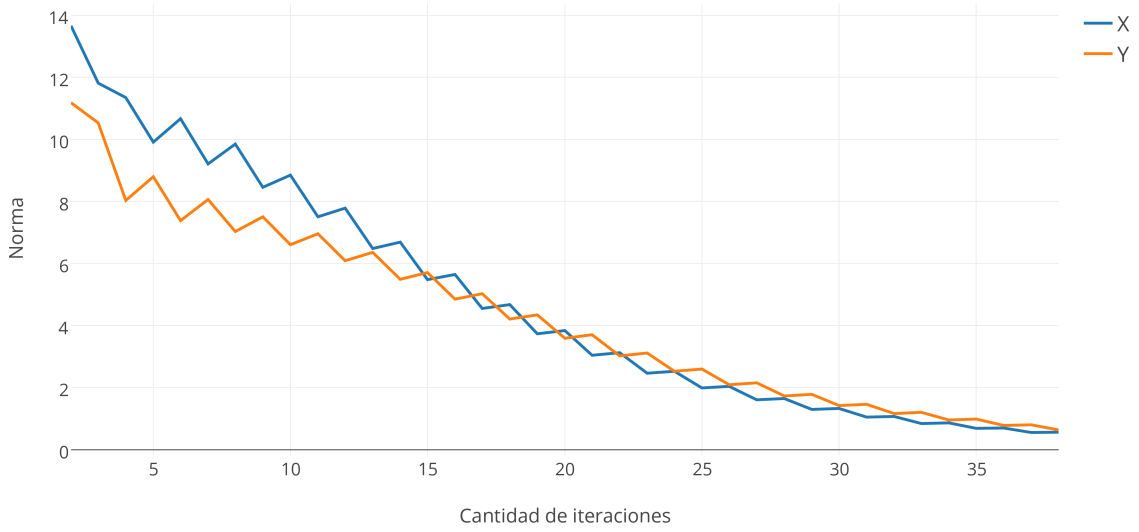


Figura 6: Variación de la norma de X e Y en cada iteración para la Red Complexity
La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 121.

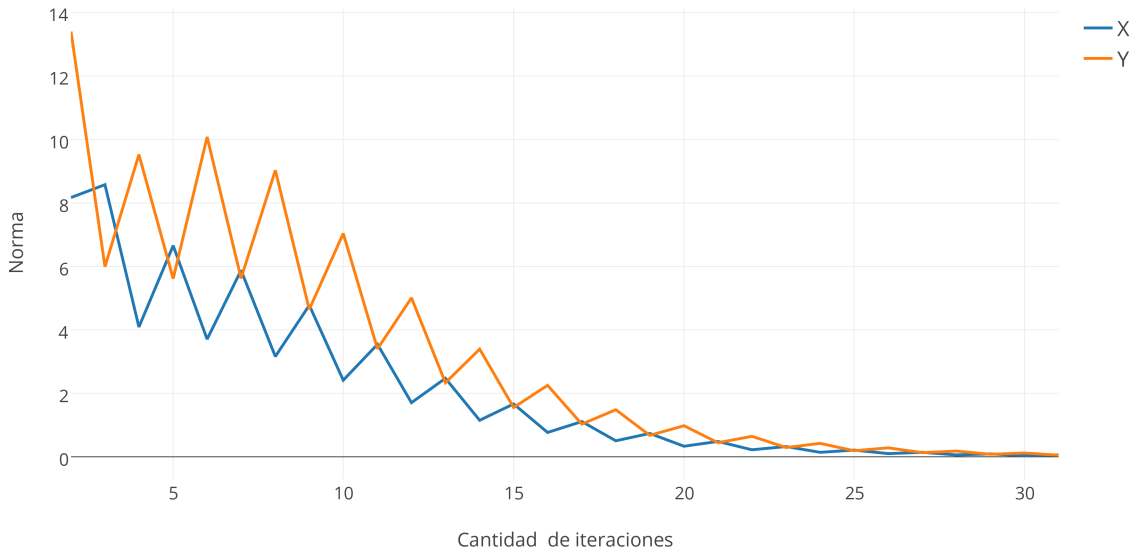


Figura 7: Variación de la norma de X e Y en cada iteración para la Red Abortion
La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 56.

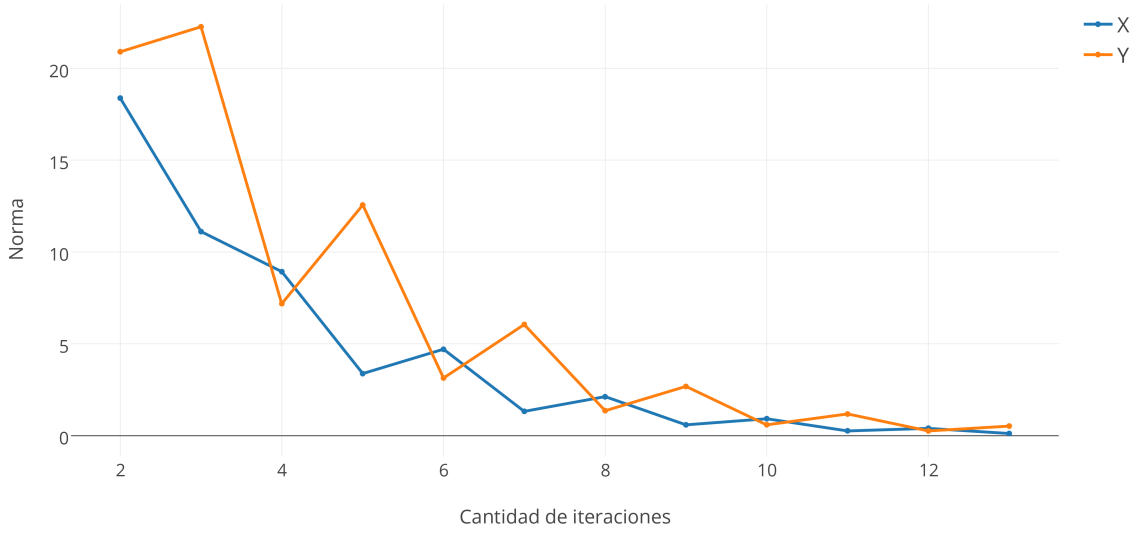


Figura 8: Variación de la norma de X e Y en cada iteración para la Red Movies
La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 29.

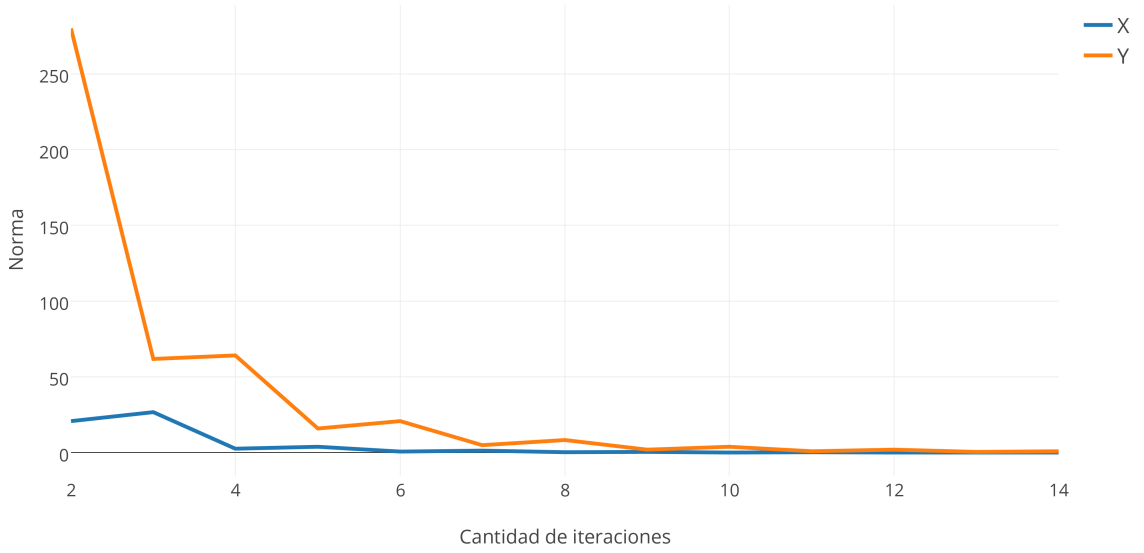


Figura 9: Variación de la norma de X e Y en cada iteración para la Red Stanford
La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 28.

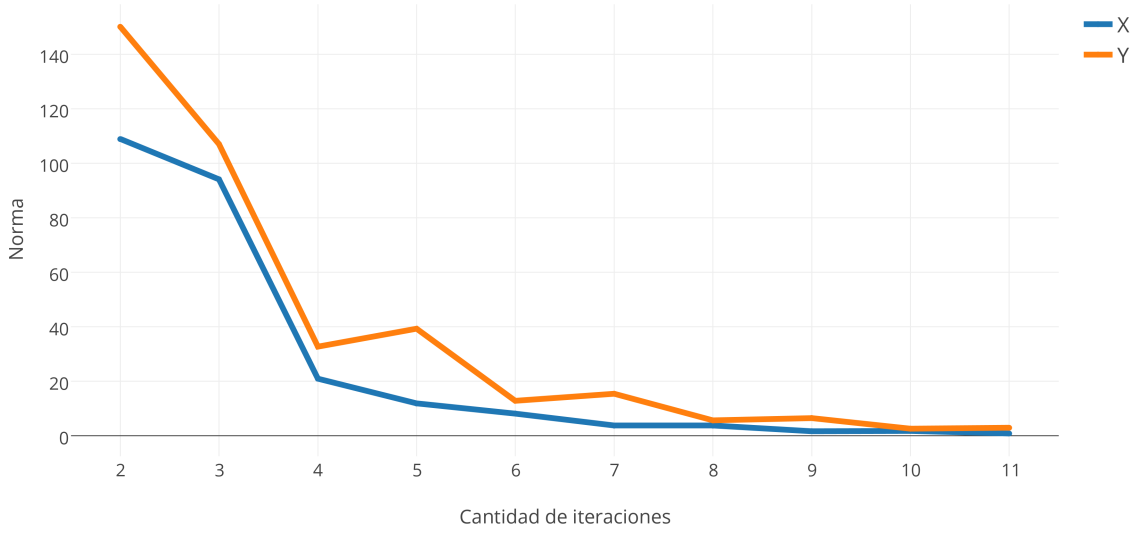


Figura 10: Variación de la norma de X e Y en cada iteración para la Red Notre Dame
La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 31.

En todos los gráficos presentados se puede apreciar que la norma de los vectores X e Y converge a cero. Cabe destacar un suceso llamativo: en las primeras iteraciones los valores de la norma de la diferencia de X con los de Y son alternadamente muy cercanos y muy lejanos. No se puede unificar esta observación para todos los casos, ya que cada uno presenta sus propias anomalías.

En el caso de red más chica (*Complexity*), si bien se puede apreciar esta alternancia entre la distancia del valor de X e Y, el vector X se ubica por encima del Y en las primeras iteraciones para luego situarse muy cercanos hasta converger a cero. En el caso de la segunda red (*Abortion*), se puede apreciar un comportamiento parecido donde el vector X se sitúa por encima del Y, sin embargo alternadamente van

adquiriendo un valor muy similar que hace que el valor de Y sea mayor al de X. La última red de tamaño mediano (*Movies*) reitera el comportamiento de las dos anteriores, sólo que en las iteraciones que los vectores X e Y antes estaban muy cerca ahora se cruzan ubicándose siempre el vector Y por encima del X en mayor proporción que antes.

En el caso de las redes grandes (*Notre Dame y Stanford*), sus comportamientos son muy similares entre sí. El vector Y se sitúa en todo momento por encima del X, y luego de una mayor cantidad de iteraciones se observa que ambos vectores convergen a cero.

Algo notorio que se observa es que: cuanto más grande sea la red, una mayor cantidad de iteraciones es necesaria para llegar al valor que converge.

3.3. Factor Temporal

PONER ACA LOS EXPERIMENTOS SOBRE EL VECTOR INICIAL Y LA CANTIDAD DE ITERACIONES Y para el algoritmo de PageRank, ver como dan los resultados para cada c, para la misma red y ver qué valor de c podría tener sentido. fruta++

El factor tiempo en el uso es una característica importante de los algoritmos si se van a computar online, ya que es tiempo que el usuario debe esperar hasta obtener su resultado. Si bien el algoritmo de PageRank es computado, generalmente, online; el de HITS no lo es ya que debe primero armar la red con la que va a trabajar.

El siguiente gráfico mide el tiempo de cómputo para redes de tamaño mediano.

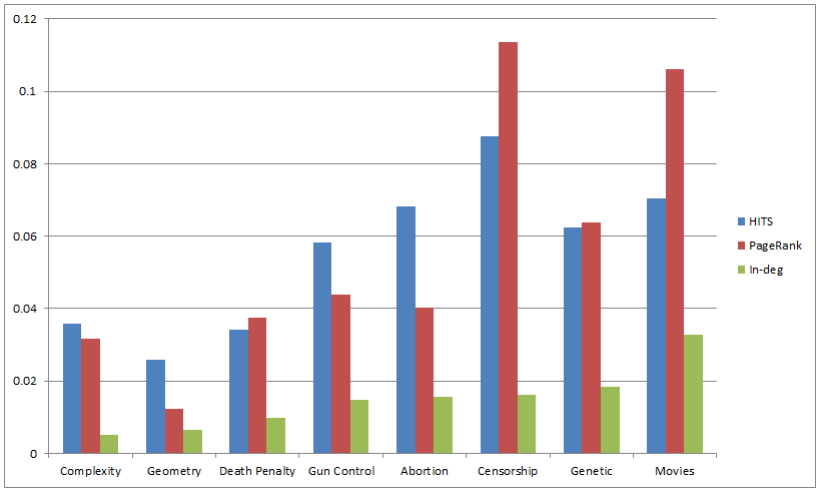


Figura 11: Medición de los tres métodos para redes medianas

El siguiente gráfico mide el tiempo de cómputo para redes de tamaño grande.

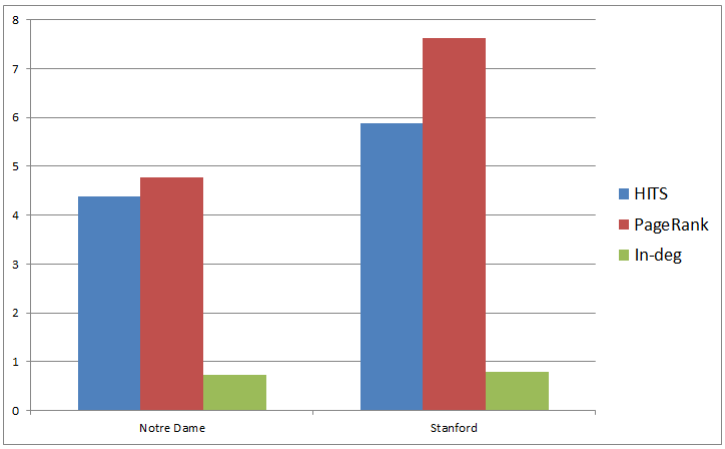


Figura 12: Medición de los tres métodos para redes grandes

Gus la tiene re clara aca, yo no tengo ni idea. perdon
NOS QUEDO PENDIENTE? o es lo de gus?(lo que sigue aca abajo) ...
Un trabajo que nos quedó pendiente es experimentar cuáles son los factores que influyen en el aumento del tiempo para el algorítmo de HITS entre la dimensión de la matriz, la cantidad de elementos o mismo la composición del grafo de la red.

3.4. Ejemplos ilustrativos y comparación de los tres Métodos

Con el fin de poner a prueba experimentalmente el conocimiento aprendido de manera teórica acerca de los algoritmos, destinamos esta sección a la presentación de las hipótesis que formulamos acerca de sus funcionamientos, a los ensayos dirigidos a partir de las mismas y a los resultados obtenidos a partir de aquellos junto con una serie de posibles explicaciones a los comportamientos observados.

Hipótesis 1: ***Si se toma un conjunto de páginas base (es decir, no apuntadas por ninguna otra de la red) que tengan links salientes hacia un conjunto de webs de menor dimensión cuyas direcciones se encuentran muy apuntadas, entonces mientras que el algoritmo de HITS va a considerar importantes dichas páginas, no encabezarán el listado de webs sugerido por un algoritmo de PageRank***

Para contrastar esta hipótesis diseñamos una red que cuya particularidad es que existen conjuntos de webs poco apuntadas que poseen links salientes a un grupo reducido de páginas que - en lo posible - no apunten recíprocamente a aquellas. La idea es generar una instancia en la cual existan un conjunto de autoridades destacables que concedan a las páginas que las apuntan una fuerte identidad de hubs, para poder anailzar cómo jerarquiza a estas últimas cada método.

Los resultados obtenidos para la Red 1 (citada continuación) son los presentados en las figuras 3.4, 3.4 y 3.4.

Red 1:

1 Empirismo
2 Racionalismo
3 Humanismo
4 Crítica de la razón pura
5 Metafísica
6 Discurso del método
7 Lógica de Port-Royal
8 Principio de razon suficiente
9 Juicio sintáctico a priori
10 Investigación sobre el entendimiento humano
11 Razonamiento inductivo
12 Instrumentalismo
13 Ecdótica
14 Robert G. Ingersoll
15 Antihumanismo

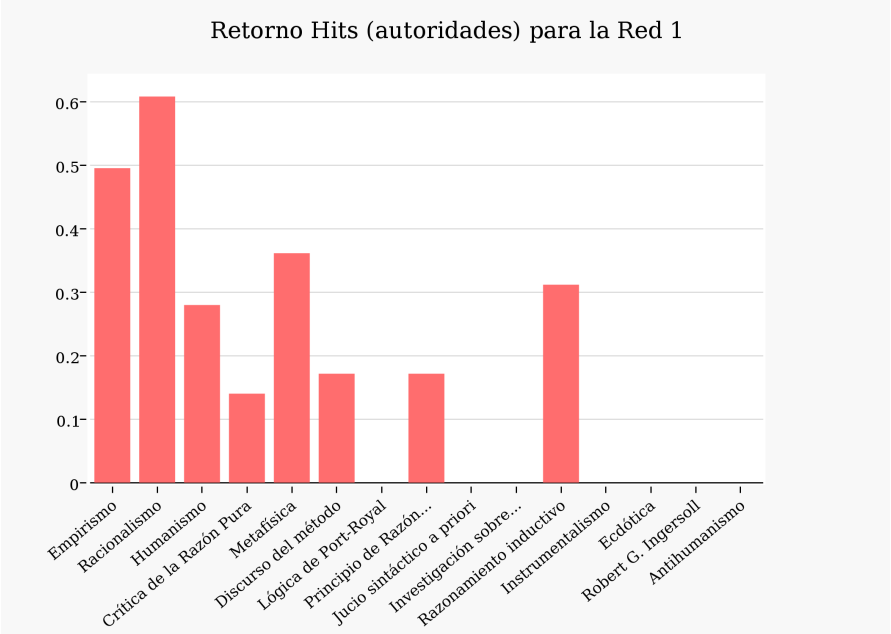
che, a todas las figuras las llama igual O_O

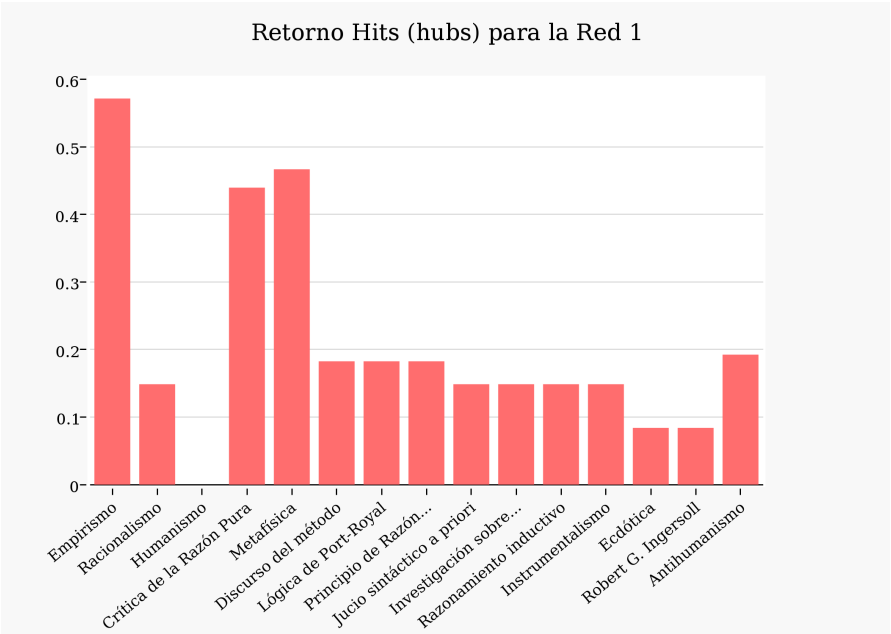
En los gráficos adjuntados se puede observar que las webs mejor puntuadas para PageRank resultaron ser
Empirismo
Racionalismo
Humanismo
Metafísica
Razonamiento inductivo
Principio de la razón

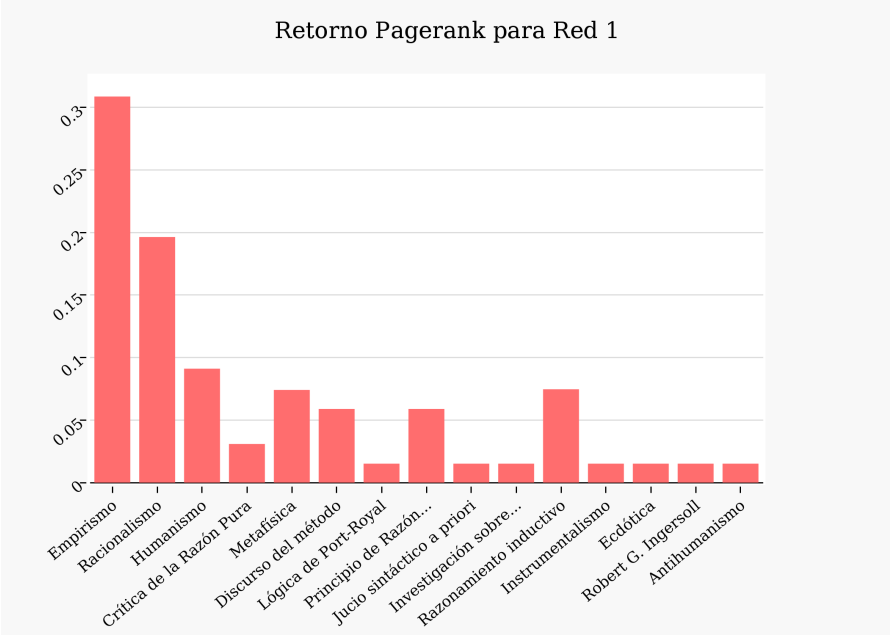
Si analizamos los resultados de Hits para organizar las autoridades, encontramos en este caso más páginas coincidentes con las priorizadas con el algoritmo de Pagerank. De hecho, las cinco páginas a las que éste último método concede las primeras posiciones, son las mismas que pueden encontrarse en los cinco principales puestos resultantes de Hits para autoridades.

Ahora bien, Si comparamos con la jerarquía sugerida por Hits (hubs), notamos que una de las webs anteriormente mencionadas (humanismo) tiene valor nulo, mientras que muchas otras sugeridas por este método ni siquiera figuran entre las que PageRank considera webs de importancia (crítica a la razón pura, antihumanismo, etc.).

Los acuerdos entre ambos gráficos se dan en los casos de las páginas “Empirismo” y “Metafísica”, las


Retorno Hits (autoridades) para la Red 1
0.6
0.5
0.4
0.3
0.2
0.1
0
Empirismo
Racionalismo
Humanismo
Crítica de la Razón Pura
Metafísica
Discurso del método
Lógica de Port-Royal
Principio de Razón...
Juicio sintáctico a priori
Investigación sobre...
Razonamiento inductivo
Instrumentalismo
Ecdótica
Robert G. Ingersoll
Antihumanismo

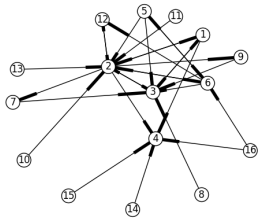

Retorno Hits (hubs) para la Red 1
0.6
0.5
0.4
0.3
0.2
0.1
0
Empirismo
Racionalismo
Humanismo
Crítica de la Razón Pura
Metafísica
Discurso del método
Lógica de Port-Royal
Principio de Razón...
Juicio sintáctico a priori
Investigación sobre...
Razonamiento inductivo
Instrumentalismo
Ecdótica
Robert G. Ingersoll
Antihumanismo



cuales poseen más links salientes que todas las demás (y apuntan a la mejor autoridad, que es racionalismo) y, a su vez, son apuntadas por más de dos webs. En otras palabras, estas dos páginas no nos interesan realmente a los fines analíticos de la hipótesis en cuestión, dado que al poder ser considerados "hibridos" entre hubs y autoridades, no nos permiten emitir conclusiones claras al respecto.

Dicho esto, si dercartaramos estos dos casos notaríamos que las coincidencias entre los resultados arrojados son muy disímiles.

Esto nos hace pensar que el nivel de relación que puedan tener los resultados de PageRank con los de HITS, dependerá diréctamente del criterio que se tome para ordenar jerárquicamente los resultados de HITS. Con este ejemplo podemos deducir que si los hubs fueran considerados más importantes que las autoridades, entonces las coincidencias entre los principales resultados de los métodos enunciados serán menores que en el caso en que el nivel de prioridades entre hubs y autoridades se intercambie.



Hipótesis 2: ***Si se toma la misma red y se le incluye una web que apunte a las mayores autoridades, entonces ésta pasará a ser el hub más importante de la nueva red.***

Para intentar contrastar esta proposición tomamos la Red 1 presentada en la $hip\acute{o}tesis\ 1$, le agregamos una web nueva ("$EscuelasFilos\acute{o}ficas$"), ejecutamos los algoritmos de PageRank y de HITS y luego analizamos los datos obtenidos en este apartado en conjunto con los arrojados en el inmediato anterior.

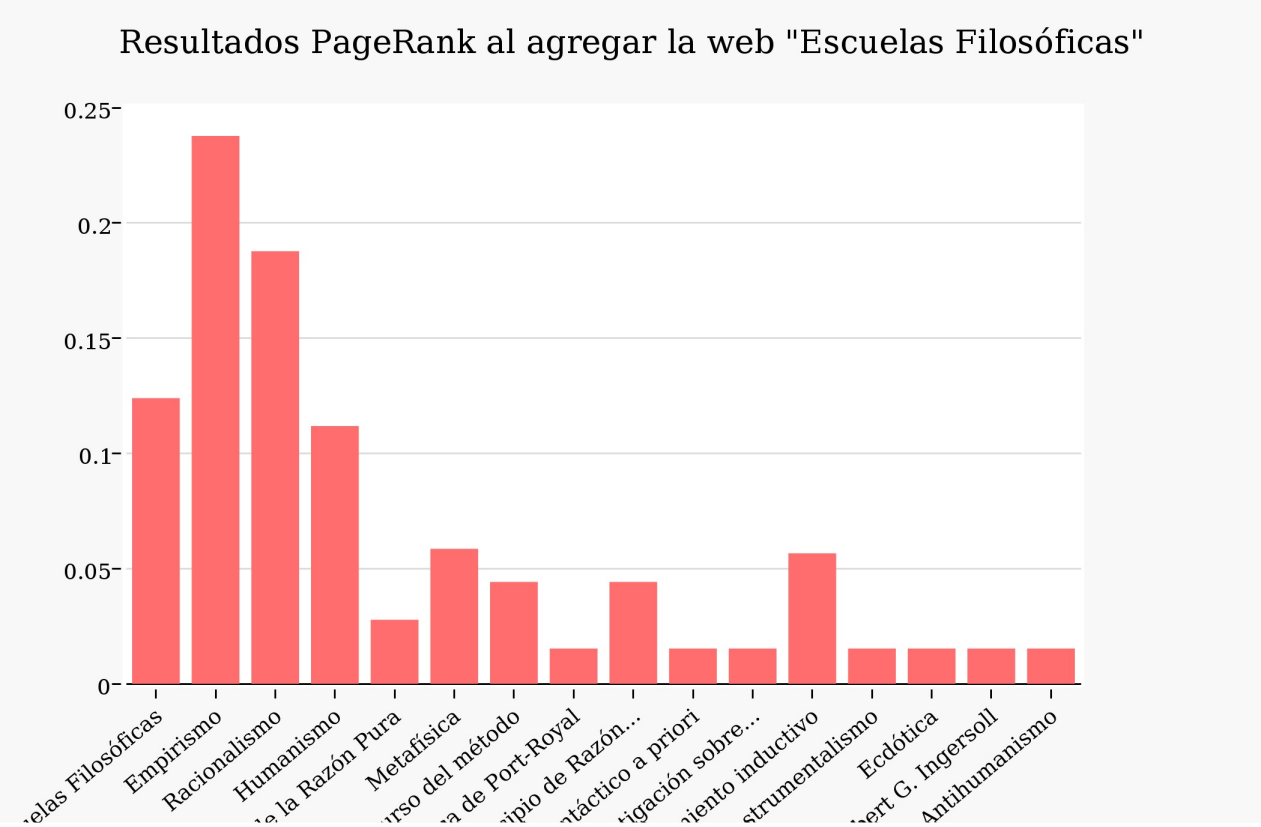
Los resultados son los presentados en las figuras **??**, **??** y **??**, mientras que en la figura 3.4 se puede observar el grafo de la nueva web.

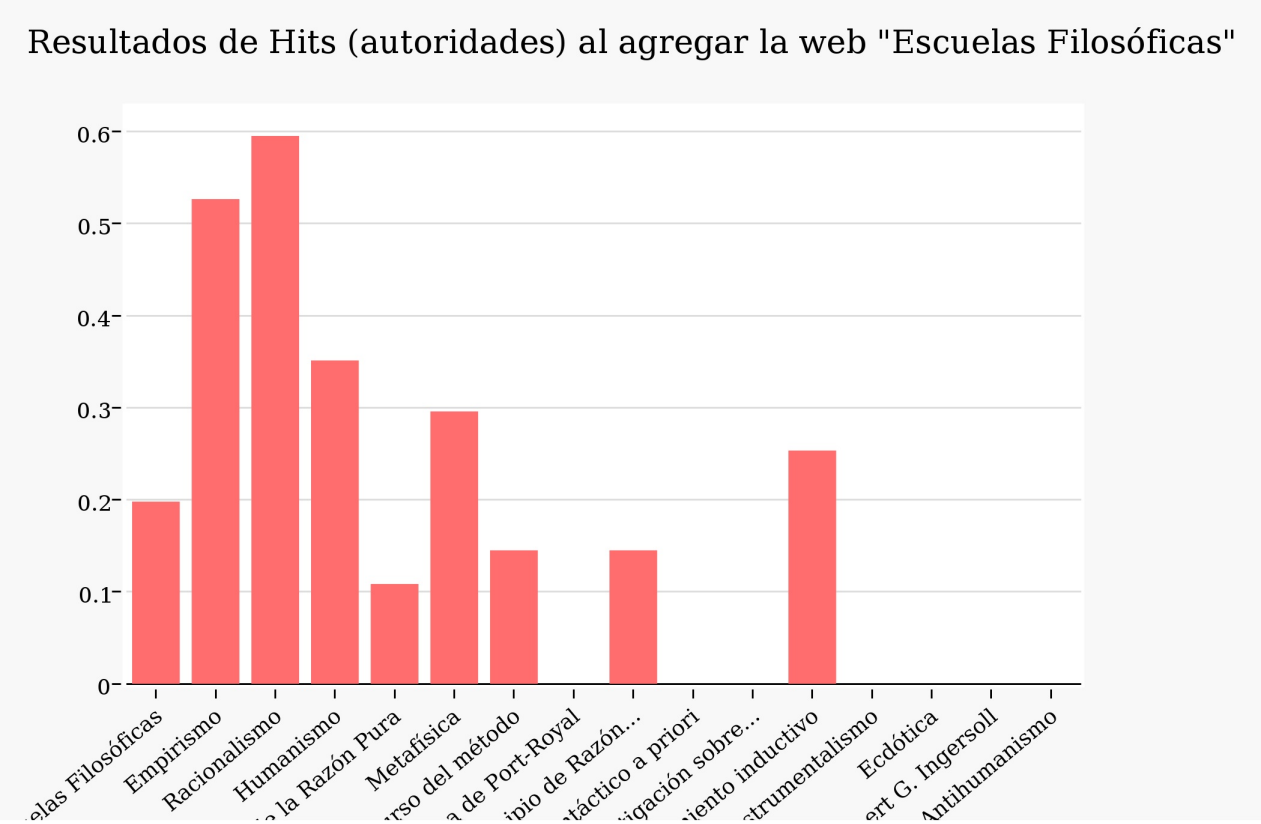
Red 2:

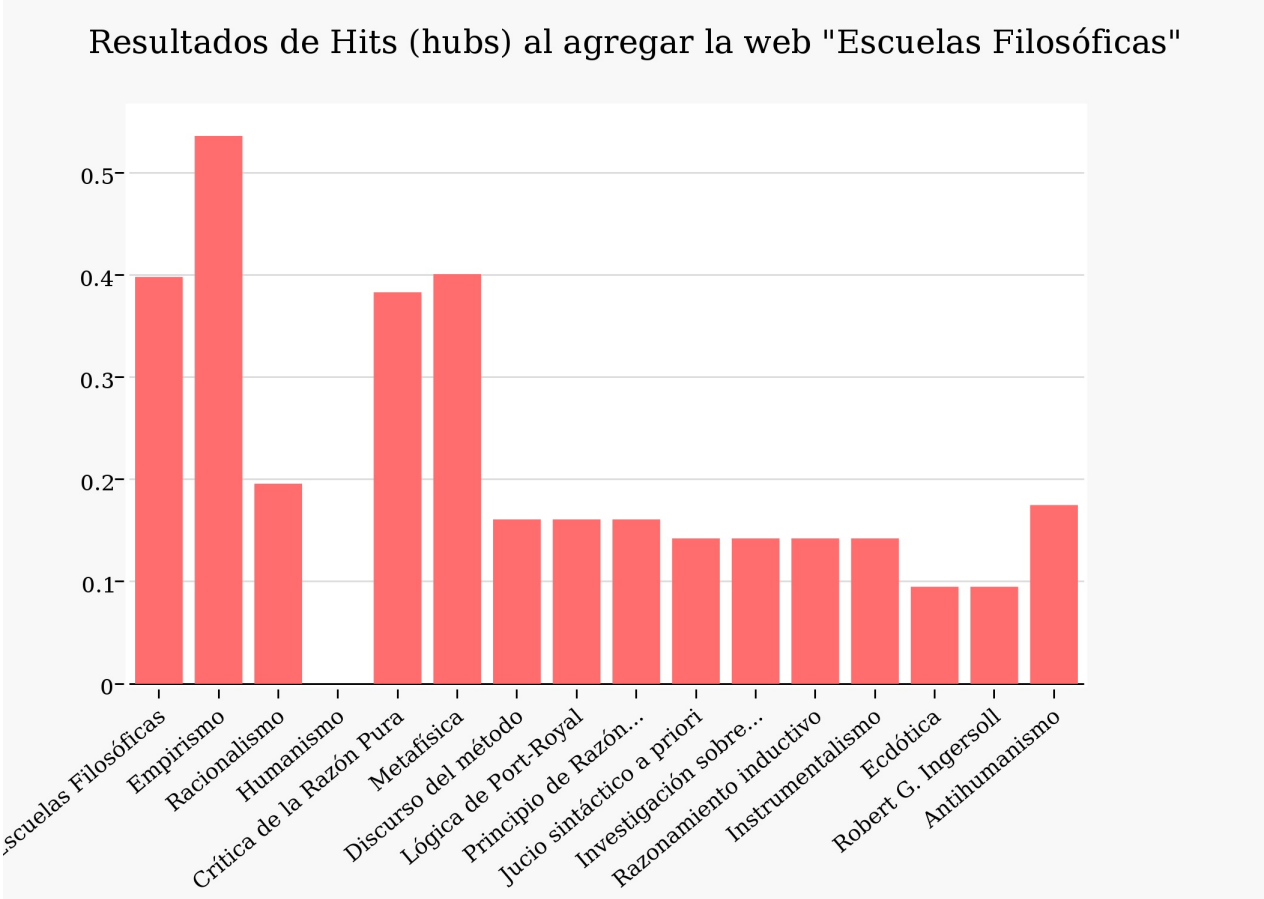
1 Escuela Filosófica
2 Empirismo
3 Racionalismo
4 Humanismo
5 Crítica de la razón pura
6 Metafísica
7 Discurso del método
8 Lógica de Port-Royal
9 Principio de razon suficiente
10 Juicio sintáctico a priori
11 Investigación sobre el entendimiento humano
12 Razonamiento inductivo
13 Instrumentalismo
14 Ecdótica
15 Robert G. Ingersoll
16 Antihumanismo

Si se contrasta la figura 3.4 con la figura 3.4 (propia de este experimento) se puede ver que el máximo valor de hub no se realiza en el peso del sitio agregado, sino en los que eran hubs destacados en la red original.
Si bien no obtuvimos los resultados esperados (la web agregada no resultó ser el hub más importante), esto pudo deberse a que el peso de hubs preexistentes era de gran magnitud. Al subestimar la dificultad de reemplazar un hub por otro y considerar que bastaría con que la nueva página apuntase a unas pocas pero fuertes autoridades, no tuvimos presente el hecho de que cada link saliente de una página incrementa o mantiene en la misma su valor de hub, sea cual fuere el peso de la página apuntada. Sin tener en cuenta esto, desestimamos la posibilidad de que se constituyera como hub prioritario una web que apuntara a muchas otras de pesos - en principio - poco relevantes, siendo que existía otra que aglutinaba en sus links a varias de las mejores autoridades de la red.

Nos parece importante destacar que a pesar de no haber obtenido estrictamente los resultados esperados, notamos que la nueva red preservó el orden relativo de los hubs de la red inicial a excepción de la nueva página agregada que, a pesar de no haber obtenido el mejor de los valores, fue posicionada entre los hubs de mayor relevancia.


Resultados PageRank al agregar la web "Escuelas Filosóficas"
0.25
0.2
0.15
0.1
0.05
0
Empirismo
Racionalismo
Humanismo
Metafísica
Ecdótica
Antihumanismo


Resultados de Hits (autoridades) al agregar la web "Escuelas Filosóficas"
0.6
0.5
0.4
0.3
0.2
0.1
0
Empirismo
Racionalismo
Humanismo
Metafísica
Ecdótica
Antihumanismo


Resultados de Hits (hubs) al agregar la web "Escuelas Filosóficas"
0.5
0.4
0.3
0.2
0.1
0
scuelas Filosóficas
Empirismo
Racionalismo
Humanismo
Crítica de la Razón Pura
Metafísica
Discurso del método
Lógica de Port-Royal
Principio de Razón...
Jucio sintáctico a priori
Investigación sobre...
Razonamiento inductivo
Instrumentalismo
Ecdótica
Robert G. Ingersoll
Antihumanismo

Hipótesis 3: ***Sea A una página de la red apuntada por otra página B perteneciente a la misma. En caso de que se agreguen a la red nuevos nodos a los que B apunte, entonces el nivel de importancia de la web A se verá dismunído en la segunda red de acuerdo al método de PageRank.***

Con el fin de contrastar la presente hipótesis, agregamos a la $Red\ 2$ un conjunto de links salientes de "$Antihumanismo$"(página que apuntaba a "humanismo"), resultando la siguiente red:

Red 3:

1 Escuela Filosófica
2 Empirismo
3 Racionalismo
4 Humanismo
5 Crítica de la razón pura
6 Metafísica
7 Discurso del método
8 Lógica de Port-Royal
9 Principio de razon suficiente
10 Juicio sintáctico a priori
11 Investigación sobre el entendimiento humano
12 Razonamiento inductivo
13 Instrumentalismo
14 Ecdótica
15 Robert G. Ingersoll
16 Antihumanismo
17 Razón
18 Muerte de Dios
19 Claude Levì-Strauss
20 Michel Foucault

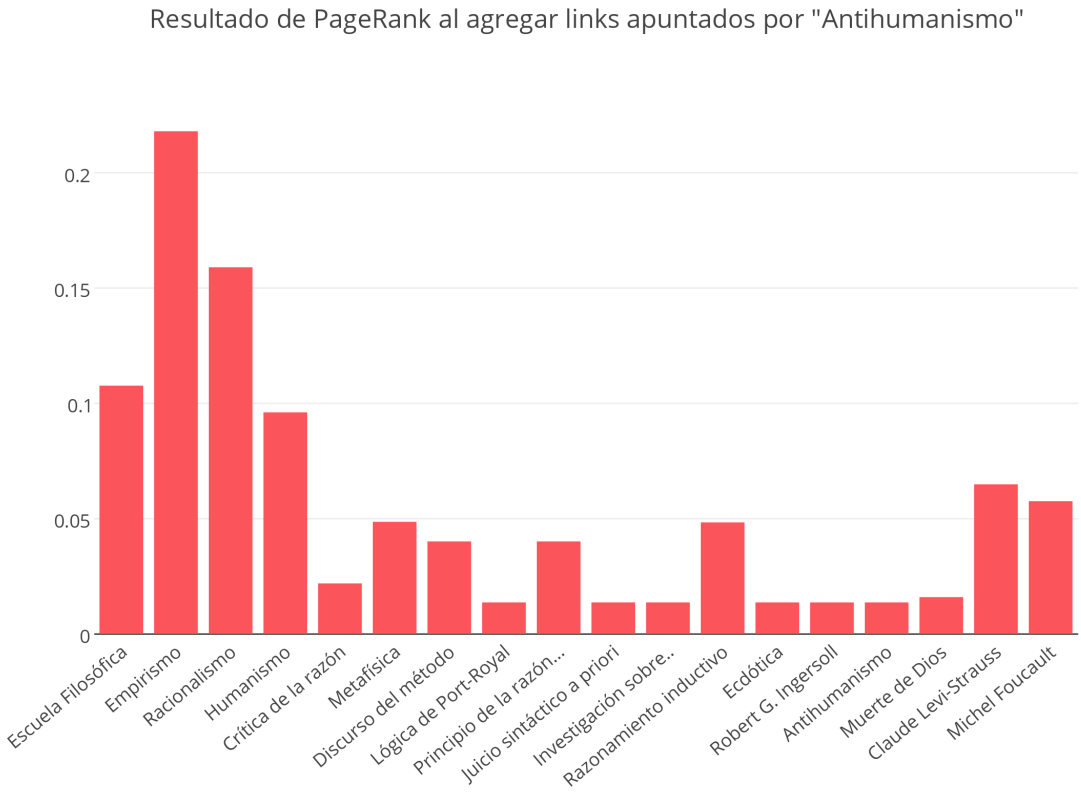
Los resultados obtenidos son los presentados en las figuras 3.4, 3.4, mientras que en la figuras 3.4 se puede observar el grafo de la web inicial.

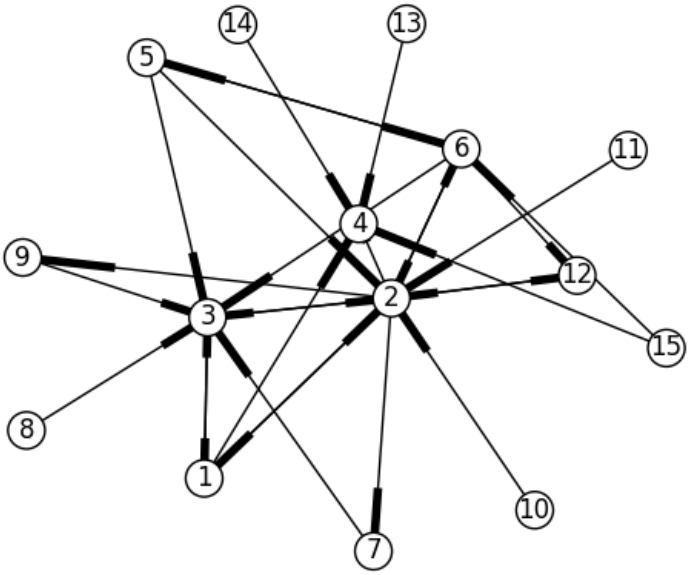
Al realizar este experimento, nuestro principal objetivo consistía en averiguar si el incremento de los links salientes de un nodo que apuntara a pocas páginas afectaría a las mismas, decrementando su importancia. Sin embargo, la fluctuación de dichos valores no fue lo más llamativo de la compraración entre el orden dado por PageRank en la red inicial y el dado por el mismo método para la red modificada.

Si bien se puede apreciar una diferencia negativa entre el valor de la página de "humanismo" entre la primera instancia y la segunda, también es cierto que la mayor parte de los nodos manifiestan este cambio. Esto puede deberse a la imposibilidad de garantizar que todos las modificaciones realizadas sean locales, puesto se cuenta con una red entramada en la cual cada cambio es potencialmente transitivo.

Por otro lado, el nodo cuyo valor se disminuyó notablemente fue aquel que incorporó nuevos links salientes. Esto cobra sentido cuando se piensa en la interpretación que se les da a los valores obtenidos en el método analizado: cada uno representa la proporción de tiempo que, en el largo plazo, el navegante aleatorio pasa en dicha página. De esta forma, si se incrementa la cantidad de links salientes de un nodo, se amplía la cantidad de sitios que el navegante aleatorio visitará sin teletransportarse, disminuyendo así el tiempo que el mismo pasará en el nodo en cuestión.

Figura 13: Descripcion de la figura




14
13
5
6
11
4
9
12
2
3
15
8
1
10
7

Hipótesis 4: ***Al aumentar la probilidad de que el navegante aleatorio se transporte, crece también la proporcion de tiempo que se espera que permanezca en paginas que inicialmente no se consideran de gran relevancia.***
COMPLETAR TODO

Hipótesis 5: ***BLA***

LA HIP 5 LA PENSAMOS INICIALMENTE PARA PROBAR MAS O MENOS COMO IBA TODO, SIN HIPOTESIS. QUE PONGO? (ES LA Q SE AGREGA ”MATRIZ.ª LO ULTIMO)

3.5. Comparación de la calidad de los Resultados para distintos métodos

A continuación se presentan los primeros 6 y 3 resultados de cada ranking.

Abortion

Al momento de correr los tres algoritmos sobre la red *Abortion* (con 2.293 nodos), los primeros elementos en figurar dentro de los ranking fueron los siguientes:

Pagerank (bajo un C=0.85):

(0.012534) **The John Birch Society**
http://www.jbs.org
Links entrantes:32 Links salientes: 5

(0.009202) **About - The Human Internet**
http://home.about.com
Links entrantes:30 Links salientes: 0

(0.008679) **AllExperts.com**
http://www.allexperts.com/about.asp
Links entrantes:55 Links salientes: 1

(0.007845) **American Opinion Book Services Online Store**
http://www.aobs-store.com
Links entrantes:25 Links salientes: 1

(0.006514) **National Right to Life Organization**
http://www.nrlc.org
Links entrantes:184 Links salientes: 1

(0.00647) **TRIMonline - Lower Taxes Through Less Government**
http://www.trimonline.org
Links entrantes:24 Links salientes: 4

In-deg:

(184) **National Right to Life Organization**
http://www.nrlc.org
Links entrantes:184 Links salientes: 1

(126) **Planned Parenthood Federation of America**
http://www.plannedparenthood.org
Links entrantes:126 Links salientes: 0

(115) **NARAL: Abortion and Reproductive Rights: Choice For Women**
http://www.naral.org
Links entrantes:115 Links salientes: 0

(114) **DimeClicks.com - Complete Web and Marketing Solutions**
http://www5.dimeclicks.com
Links entrantes:114 Links salientes: 0

(114) **Amazon.com–Earth's Biggest Selection**
http://www.amazon.com/exec/obidos/redirect-home/youdebatecom
Links entrantes:114 Links salientes: 0

(114) **HitBox.com - hitbox web site traffic counter - internet statistics and site promotion tools - WebSideStory**
http://rd1.hitbox.com/rd?acct=WQ590703J6FB45EN5
Links entrantes:114 Links salientes: 0

Hits:

Mayores Autoridades:

(0.333946) **DimeClicks.com - Complete Web and Marketing Solutions**
http://www5.dimeclicks.com
Links entrantes:114 Links salientes: 0

(0.333946) **Amazon.com–Earth's Biggest Selection**
http://www.amazon.com/exec/obidos/redirect-home/youdebatecom
Links entrantes:114 Links salientes: 0

(0.333946) **HitBox.com - hitbox web site traffic counter - internet statistics and site promotion tools - WebSideStory**
http://rd1.hitbox.com/rd?acct=WQ590703J6FB45EN5
Links entrantes:114 Links salientes: 0

Mayores Hubs:

(0.095693) **Abortion Books Pro and Con**
http://www.4greatbooks.com/abortion-books.htm
Links entrantes:0 Links salientes: 27

(0.09428) **Government Debates and Polls**
http://www.youdebate.com/government.htm
Links entrantes:0 Links salientes: 10

(0.09428) **Political Debates and Polls**
http://www.youdebate.com/POLITICS.htm
Links entrantes:0 Links salientes: 10

Es necesario destacar que la página *DimeClicks.com* queda en la posición 1 dentro del Ranking de Autoridades de HITS y en la posición 4 en el de In-Deg, cuando esta página no nos es de interés al momento de hacer una búsqueda con el string *"Abortion"* ya que consiste en una página de Marketing y soluciones Web. En cambio, para PageRank esta página se sitúa en la posición 27 con un puntaje de 0,00318. Además, la página que ocupa la tercera posición del Ranking de pesos de autoridad -*HitBox.com*- tampoco está relacionada con el contexto de *Aborto*, ya que habla de promociones y estadísticas en Internet. Mientras que HitBox se sitúa en la posición 6 para In-Deg y en la posición 27 para PageRank con un puntaje de 0,00318.

Por lo tanto, se puede concluir que para estar Red, PageRank sabe filtrar *"mejor"* las páginas que no nos son de ningún interés bajo el contexto de búsqueda.

Movies

A continuación se muestran los rankings obtenidos tras correr los tres algoritmos sobre la red *Movies* (con 5.757 nodos), los primeros elementos en figurar dentro de los ranking fueron los siguientes:

PageRank (bajo un C=0.85):

(0.007915) **On Wisconsin**
http://www.onwisconsin.com
Links entrantes:127 Links salientes: 8

(0.007829) **GuideLive: Movies in Dallas and Fort Worth**
http://www.guidelive.com/topic/movies.htm
Links entrantes:62 Links salientes: 51

(0.007015) **citysearch.com**
http://www.citysearch.com
Links entrantes:27 Links salientes: 9

In-Deg:

(393) **The Internet Movie Database (IMDb).**
http://www.moviedatabase.com
Links entrantes:393 Links salientes: 0

(277) **Hollywood.com - Your entertainment source for movies, movie showtimes, movie reviews, television and celebrity news.!**
http://www.hollywood.com
Links entrantes:277 Links salientes: 0

(143) **Paramount Pictures - Home Page**
http://www.paramount.com
Links entrantes:143 Links salientes: 0

HITS:

Mayores Autoridades:

(0.1412) **Empty title field**
http://chatting.about.com
Links entrantes:70 Links salientes: 0

(0.139835) **About.com A-Z**
http://a-zlist.about.com
Links entrantes:48 Links salientes: 0

(0.139793) **About - Arts/Humanities**
http://home.about.com/arts
Links entrantes:47 Links salientes: 0

Mayores Hubs:

(0.159812) **History of Classic Movies**
http://classicfilm.miningco.com/entertainment/classicfilm/msub19.htm
Links entrantes:0 Links salientes: 73

(0.159471) **Movie Reviews**
http://movieboxoffice.miningco.com/entertainment/movieboxoffice/msub7.htm
Links entrantes:0 Links salientes: 68

(0.159471) **Characters: Creatures**
http://starwars.miningco.com/entertainment/starwars/msubchar-crea.htm
Links entrantes:0 Links salientes: 68

En esta ocasión, también se puede apreciar que las páginas con mayor puntaje de autoridad no son las más acertadas para nuestro contexto. Sin embargo, lo más llamativo de este caso es que las páginas con mejor puntaje de Hub resultan ser páginas acertadas para la búsqueda, páginas que un usuario podría estar interesado en encontrarse al momento de buscar el string "movie". La página *History of Classic Movies* queda en la posición 3783 para PageRank y para In-Deg, con un puntaje de Autoridad que lo ubica en la posición 2888. Por otro lado, la página *Movie Reviews* se ubica en la posición 3771 del ranking de PageRank, también en la 3771 de In-Deg y ocupa la posición 2876 con su puntaje de Autoridad.

De este modo, se puede concluir que -para la Web Movies- priorizar las páginas con mayor puntaje de Hub por sobre las que tengan mejor puntaje de Autoridad es una buena idea, ya que los resultados obtenidos fueron más cercanos a lo esperado de este modo.

4. Conclusiones

A lo largo del presente trabajo hemos analizado distintos métodos cuyos propósitos consisten en jerarquizar un grupo de páginas interrelacionadas a partir de ciertos criterios y modelos de la realidad.

Habiendo formulado nuestras propias hipótesis acerca del comportamiento de dichos métodos y sus puntos fuertes y endebles y habiendo podido experimentar con ellos, nos proponemos en este momento aunar las interpretaciones de cada uno de los resultados obtenidos para elaborar una conclusión que sirva a la resolución del problema planteado.

Ante todo es importante destacar que la indicación sugerida por nosotros dependerá del método con el cual se ordenen las páginas web. Es por ello que se enumeran a continuación las estrategias sugeridas para cada uno de los métodos posibles:

In-deg: Como ya se vio, para puntuar alto en este modelo basta con conseguir que una gran cantidad de páginas apunten a la propia. Es por ello que en este caso nos remitimos a sugerir que se compren tantos links como sea posible, sin ahondar en profundidad en la importancia relativa de cada página dentro de la red.

HITS: En este caso, como la forma de presentar los resultados es devolviendo primero las páginas con mayor puntaje de Autoridad y luego las que tengan mayor puntaje de Hub, nos situamos en aumentar su puntaje de Autoridad. A través de una serie de experimentaciones y análisis notamos que un factor de gran influencia a la hora de aumentar el valor de un nodo como autoridad está dado por la cantidad y calidad de los hubs que lo apuntan. Por este motivo, nuestra recomendación es que el cliente consiga que su página sea apuntada por la mayor cantidad de webs de este tipo, cuyo orden se encuentre entre los principales.

PageRank: Considerando los distintos resultados obtenidos a partir de las pruebas y ensayos constituídos en torno a este método, concluímos que la forma más efectiva de garantizar que mejore la posición de la web del cliente en una red es realizando un análisis de la misma y consiguiendo que la apunten la mayor cantidad posible de sitios que poseen una posición destacada dentro de la misma y que a su vez no apuntan a demasiadas páginas.

Este criterio no es arbitrario, dado que el valor que una página transmite a otra depende no sólo de su propio valor sino también de la cantidad de sitios que también se están beneficiando de él.

Si esto no fuese posible (por ejemplo porque no existiera en la red una cantidad suficiente de páginas con las características mencionadas), entonces el criterio para discernir de qué manera proseguir debería ser lo suficientemente sutil y detallista como para escoger entre una serie de páginas cuyo peso no sea de una magnitud considerable pero satisfagan tener pocos links salientes, y otro grupo de mayor importancia pero con mayor cantidad de links salientes.

5. Apéndices

5.1. Apéndice A

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*. 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

[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.

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 $x_k = \sum_{j \in L_k} \frac{x_j}{n_j}, \quad k = 1, \ldots, n$. Definimos $P \in \Re$nxn tal que $p_{ij} = 1/n_j$ si $w_{ij} = 1$, y $p_{ij} = 0$ en caso contrario. Luego, el modelo planteado es equivalente a encontrar un $x \in \Re$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 $\in \Re$nxn, 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 $\bar{1} \in \Re$n es un vector tal que todas sus componenetes valen 1. La matriz resultante P_2 corresponde a un enriquecimiento del modelo formulado en (**??**). 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.[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 $\in \Re$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:

$$x_j = \sum_{i:i \to j} y_i. \tag{1}$$

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

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

Las ecuaciones (1) y (2) podemos expresarlas matricialmente de la siguiente manera:

$$x = A^t y \tag{3}$$

$$y = Ax, \tag{4}$$

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:

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

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.

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 (3) y (4). 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

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

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

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 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.

5.2. Apéndice B

Se adjunta aquí el algoritmo realizado para insertar, de a un elemento, los valores distintos de cero de una matriz en nuestra matriz esparsa:

input : Int fil, Int col, Double elem
output: Void

i ← índice donde comienza la fila fil pasada como parámetro
fin ← índice donde comienza la fila siguiente a la pasada como parámetro
Iterador itval ← crear iterador del vector val
Iterador itcol ← crear iterador del vector ind_col
while *(i<fin ∧ col>itcol)* **do**
  Avanzar los dos iteradores
  i++
end
if *(i == tamaño del vector ind_col)* **then**
  insertar al final de ind_col el valor col pasado como parámetro
  insertar al final de val el valor elem pasado como parámetro
else
  insertar en la posición correspondiente al iterador el valor col pasado por parámetro en el vector ind_col
  insertar en la posición correspondiente al iterador el valor elem pasado por parámetro en el vector val
end
for $i \leftarrow fil + 1$ **to** $cantidaddefilas$ **do**
  ptr_fil[i] ++
end

5.3. Apéndice C

El siguiente es un ejemplo de una matriz $A \in \mathbb{R}^{4x4}$ y su traspuesta con su forma de implementación.

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 3 & 4 & 0 & 8 \\ 0 & 0 & 9 & 0 \\ 10 & 11 & 0 & 0 \end{pmatrix} \tag{5}$$

CSR:
val = [1,2,3,4,8,9,10,11]
ind_col = [0,2,0,1,3,2,0,1]
ptr_fil = [0,2,5,6,8]
CSC:
val = [1,3,10,4,11,2,9,8]
ind_fil = [0,1,3,1,3,0,2,1]
ptr_col = [0,3,5,7,8]

$$A^t = \begin{pmatrix} 1 & 3 & 0 & 10 \\ 0 & 4 & 0 & 11 \\ 2 & 0 & 9 & 0 \\ 0 & 8 & 0 & 0 \end{pmatrix} \tag{6}$$

CSR:
val = [1,3,10,4,11,2,9,8]
ind_col = [0,1,3,1,3,0,2,1]

ptr_fil = [0,3,5,7,8]
CSC:
val = [1,2,3,4,8,9,10,11]
ind_fil = [0,2,0,1,3,2,0,1]
ptr_col = [0,2,5,6,8]

En este ejemplo, se puede apreciar que al trasponer la matriz los arreglos conservan sus mismos valores, sólo que cambia la forma de interpretarlos. Es decir: ind_fil pasa a ser ind_col, ptr_col pasa a ser ptr_fil y viceversa.

6. 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):107117, 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 261270, New York, NY, USA, 2003. ACM.

[6] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. J. ACM, 46(5):604632, September 1999.