



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

El resumen de no más de 200 palabras, deberá explicar brevemente el trabajo realizado y las conclusiones de los autores de manera que pueda ser útil por sí solo para dar una idea del contenido del trabajo.

## Palabras claves

- Matriz Esparsa
- PageRank
- HITS
- In-deg

## Índice

<b>1. Introducción Teórica</b>	<b>3</b>
1.1. PageRank, HITS, In-deg . . . . .	3
<b>2. Desarrollo</b>	<b>5</b>
2.1. Elección de las estructuras . . . . .	5
2.2. Algoritmo multiplicación de una matriz por un vector . . . . .	6
2.3. Algoritmo de HITS . . . . .	7
2.4. Algoritmo de PageRank . . . . .	9
<b>3. Resultados y discusión</b>	<b>10</b>
3.1. Convergencia de PageRank . . . . .	10
3.2. Convergencia de HITS . . . . .	14
3.3. Factor Temporal . . . . .	14
3.4. Ejemplos ilustrativos y comparación de los tres Métodos . . . . .	15
3.5. Comparación de la calidad de los Resultados para distintos métodos . . . . .	30
<b>4. Conclusiones</b>	<b>34</b>
<b>5. Apéndices</b>	<b>35</b>
5.1. Apéndice A . . . . .	35
5.2. Apéndice B . . . . .	36
5.3. Apéndice C . . . . .	36
<b>6. Referencias</b>	<b>38</b>

## 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 relevantes; 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:

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

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 anteriormente.

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 [ACA PONER EL NUMERO DE REFERENCIA DE LA BIBLIOGRAFIA DONDE ESTE ESTE PAPER.](#), 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 “Tele-transportación” 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 [ACA PONER EL NUMERO DE REFERENCIA DE LA BIBLIOGRAFIA DONDE ESTE ESTE PAPER.](#), 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 **ACA PONER EL NUMERO DE REFERENCIA DE LA BIBLIOGRAFIA DONDE ESTE ESTE PAPER**, consiste en partir de una consulta sobre  $\sigma$  y focalizarse en una colección de páginas  $S_\sigma$  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  $R_\sigma$ .
- b) Se incrementa el conjunto  $R_\sigma$  añadiendo las páginas que tienen links entrantes y salientes al mismo, formando así el conjunto  $S_\sigma$ . Para cada página de  $R_\sigma$  se permite añadir, a lo sumo  $d$  páginas que la apunten y  $d$  páginas a las cuales apunte.
- c) Se eliminan de  $S_\sigma$  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  $G_\sigma$ . Nuestro trabajo asume un conjunto  $G_\sigma$  bien formado.

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

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

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

Peso de autoridad:

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

Peso de Hub:

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

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  $G_\sigma$ .

### ***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 implementación 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 almacenado en `vali`. 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 Sparse*, transponer 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 transponerlas 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 nuestro 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  $i \leftarrow 0$  to cantidad de filas do
  inicio  $\leftarrow$  m.ptr_fil[i]
  fin  $\leftarrow$  m.ptr_fil[i+1]
  for  $j \leftarrow$  inicio to fin do
    col  $\leftarrow$  m.ind_col[j]
    res[i]  $\leftarrow$  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  $i \leftarrow 0$  to cantidad de filas do
  inicio  $\leftarrow$  m.ptr_fil[i]
  fin  $\leftarrow$  m.ptr_fil[i+1]
  for  $j \leftarrow$  inicio to fin do
    fil  $\leftarrow$  m.ind_col[j]
    res[col]  $\leftarrow$  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[fil Actual]*) y se suma en *res[fil]*.

## 2.3. Algoritmo de HITS

Nuestra tarea aquí es extraer del subconjunto  $G_\sigma$  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 significativa de nodos entrantes sino que también van a tener muchas páginas en común que las apunten.

Hubs y Autoridades representan una relación 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 \(??? GUSSSSSSSSSS\)](#).

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

◦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  $X \leftarrow A^t Y$ ,  $Y \leftarrow A X$ ; 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 ([PONER EL NUMERO DEL PAPER](#)), podemos asegurar que este método converge bajo ciertas hipótesis ([ESTOY MANDANDO FRUTA?](#)) como que el grafo  $G_\sigma$  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  $A^t A$  e  $Y^*$  es el principal autovector de  $AA^t$ .



## 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 **ACA PONER EL NUMERO DEL PAPER** 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  $\lambda_1$  es 1.

A fines de emplear menos recursos de cómputo se optó por llevar a cabo el algoritmo desarrollado por Kamvar **PONER NUMERO DE PAPER**:

```

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 ← 0 to tamaño de xp do
        | xp[i] ← xp[i] * c
    end
    for i ← 0 to tamaño de xp do
        | xp[i] ← xp[i] +  $\frac{\text{norma}(x) - \text{norma}(xp)}{n}$ 
    end
    if (xp  $\simeq$  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  $\text{Norma}_1 = 1$  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  $\simeq$  considera la tolerancia (tol) pasada por parametro. Es decir, es equivalente a evaluar  $\text{abs}(x - xp) \leq \text{tol}$ .

◦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.

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

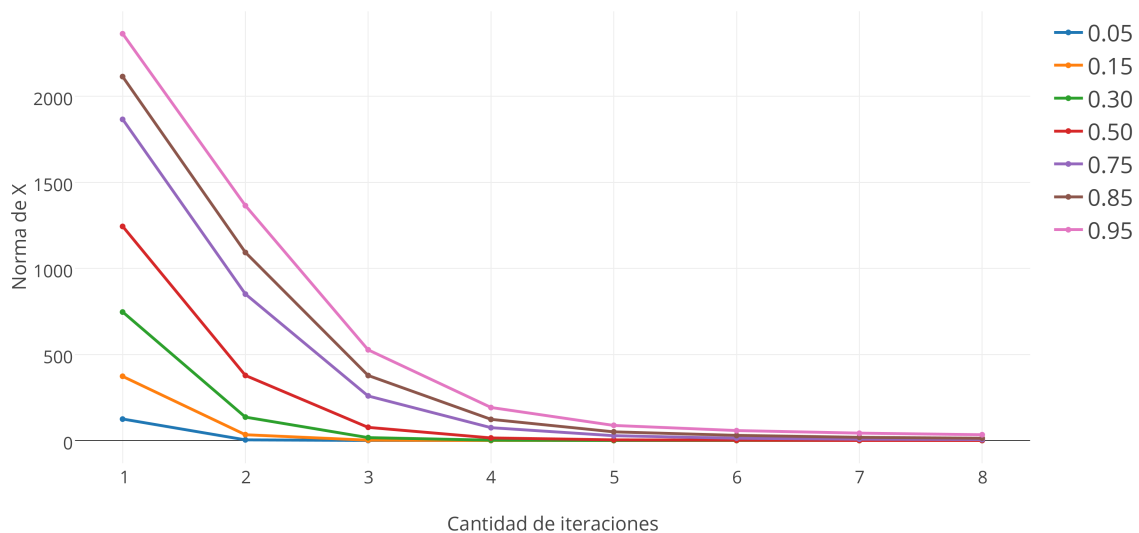


Figura 1: Variación de la norma cambiando el C para la Red Abortion

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

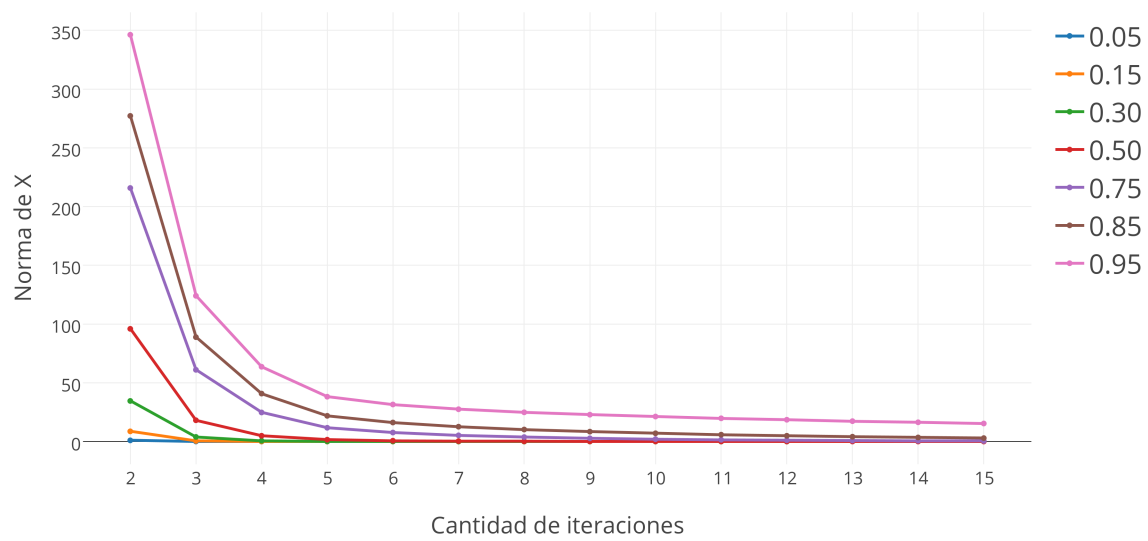


Figura 2: Variación de la norma cambiando el C para la Red Complexity

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

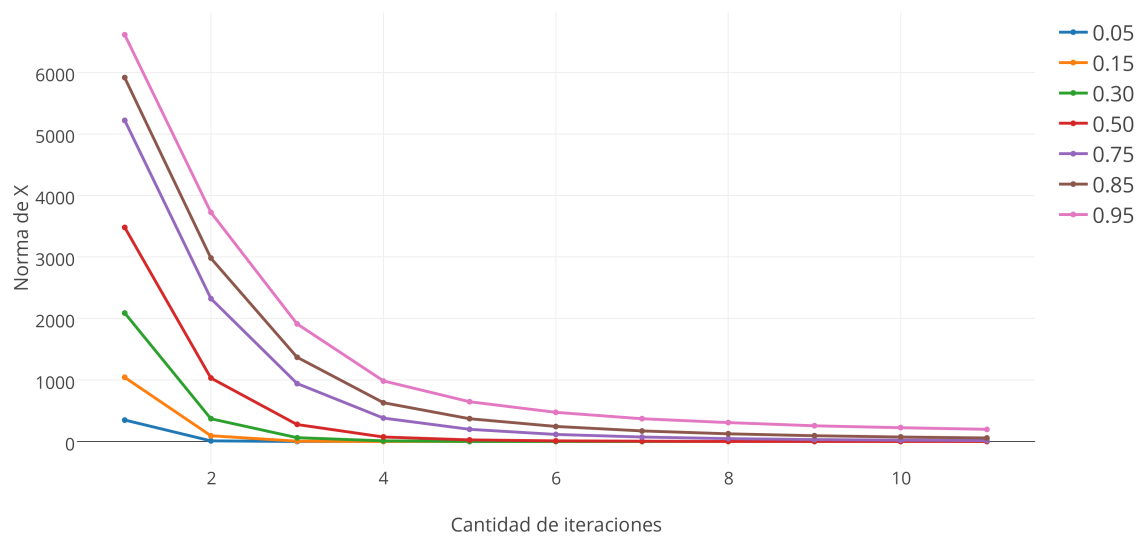


Figura 3: Variación de la norma cambiando el C para la Red Movies

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

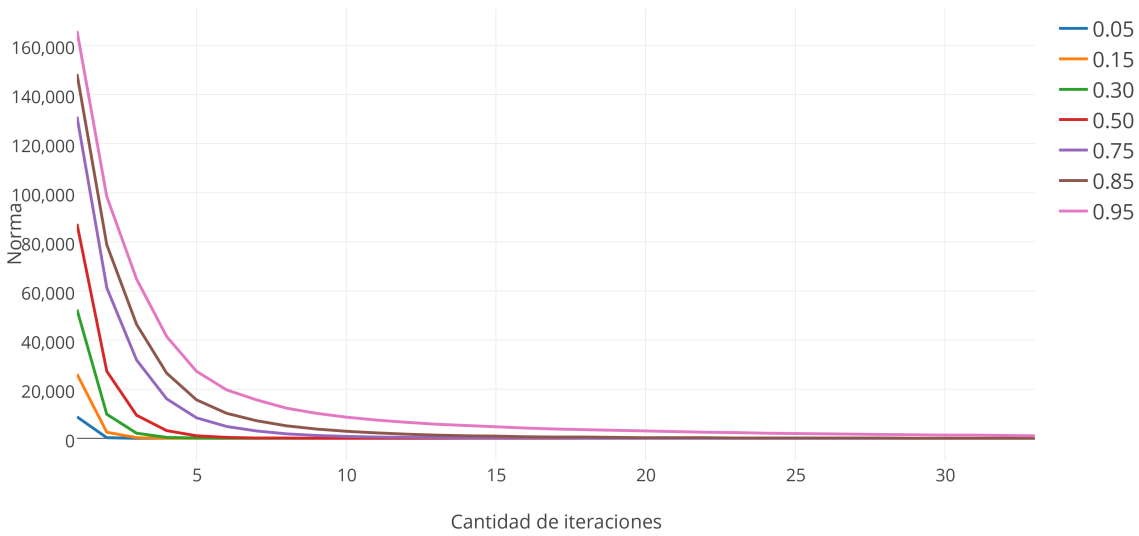


Figura 4: Variación de la norma cambiando el C para la Red Notre Dame

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	10	15	25	56	96	295

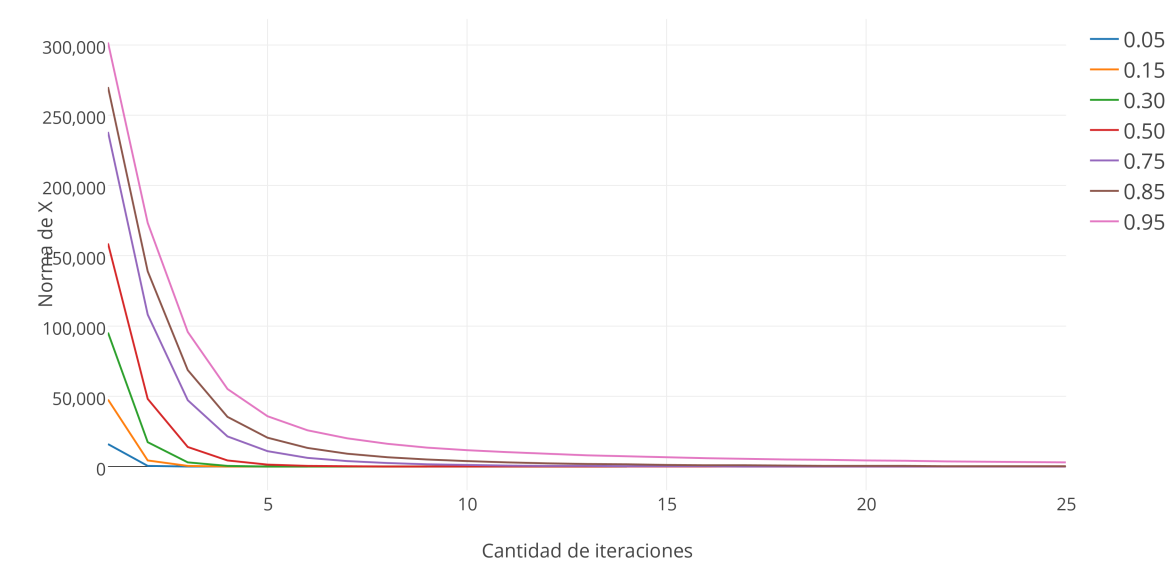


Figura 5: Variación de la norma cambiando el C para la Red Stanford

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

Experimentamos con Redes de distintos tamaños y podemos concluir 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.

### 3.2. Convergencia de HITS

Creemos que converge a 0.

Explicación: es un algoritmo circular. Se trasladan los "pesos." al multiplicar un vector  $x$  grande por la matriz, y haciendo que el otro - que es chico - tome ese valor. EN la iteración sig  $x$  sera menor...

—podemos incluir acá la cosa intermedia: hacer una comparación entre el tiempo de convergencia (o sea, ver cuál converge más rápido) de hits y PR para usarlo como fundamento en el exp 3.

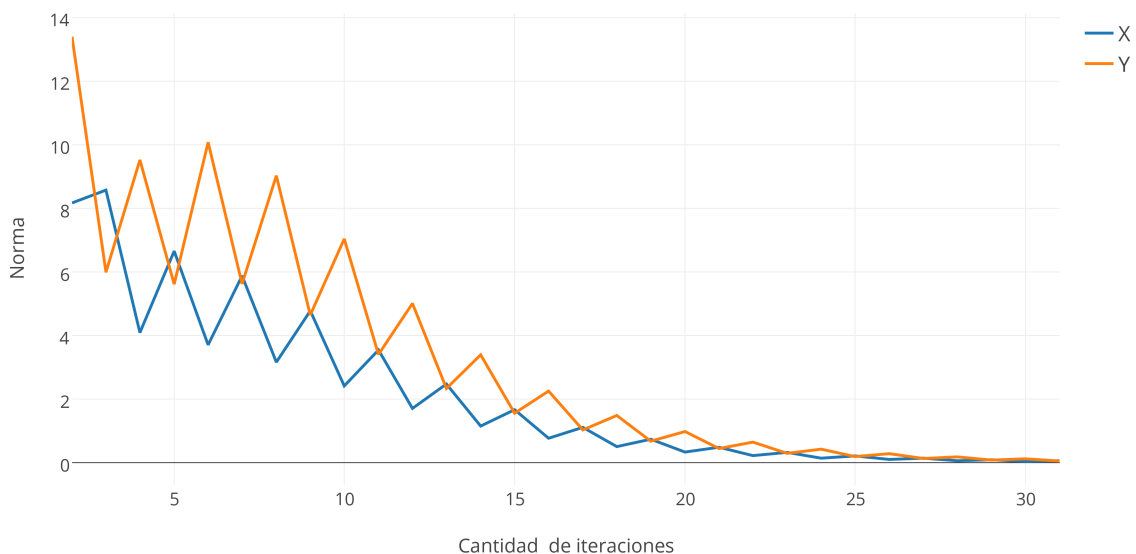


Figura 6: Variación de la norma de X e Y en cada iteración para la Red Abortion

La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 56.

La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 121.

La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 29.

La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 31.

La cantidad de iteraciones llevadas a cabo por el algoritmo fue de 28.

### 3.3. Factor Temporal

**PONER ACA LOS EXPERIMENTOS SOBRE EL VECTOR INICIAL Y LA CANTIDAD DE ITERACIONES**

\*Trabajo a futuro\* Ver si lo que influye en el tiempo que demora pr al lado de hits, es la dimension, la cantidad de elementos o la composicion de la red. Ver como dan los resultados para cada c, para la misma red y ver qué valor de c podría tener sentido. fruta+ +

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

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

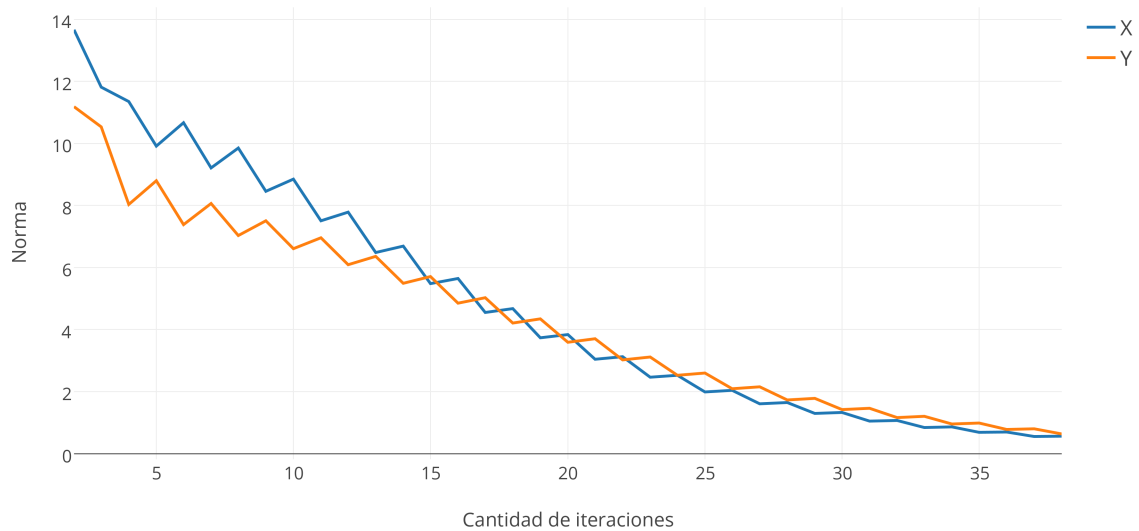


Figura 7: Variación de la norma de X e Y en cada iteración para la Red Complexity

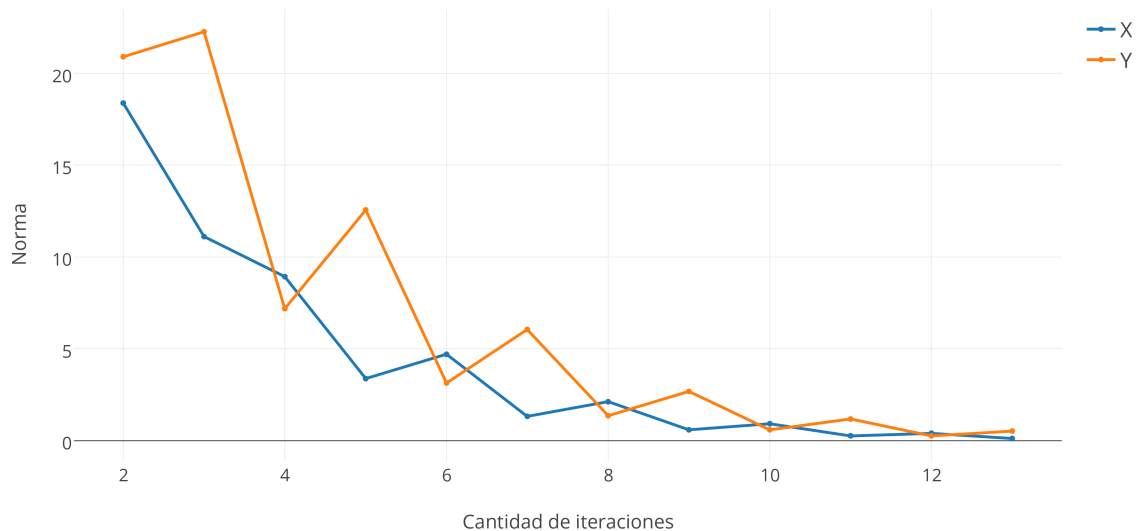


Figura 8: Variación de la norma de X e Y en cada iteración para la Red Movies

### 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*

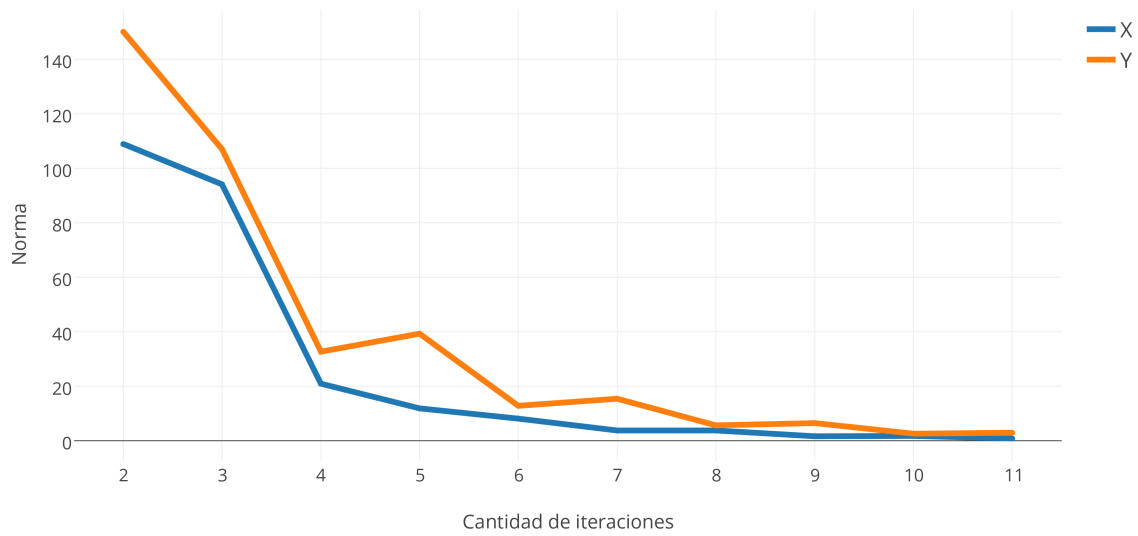


Figura 9: Variación de la norma de X e Y en cada iteración para la Red Notre Dame

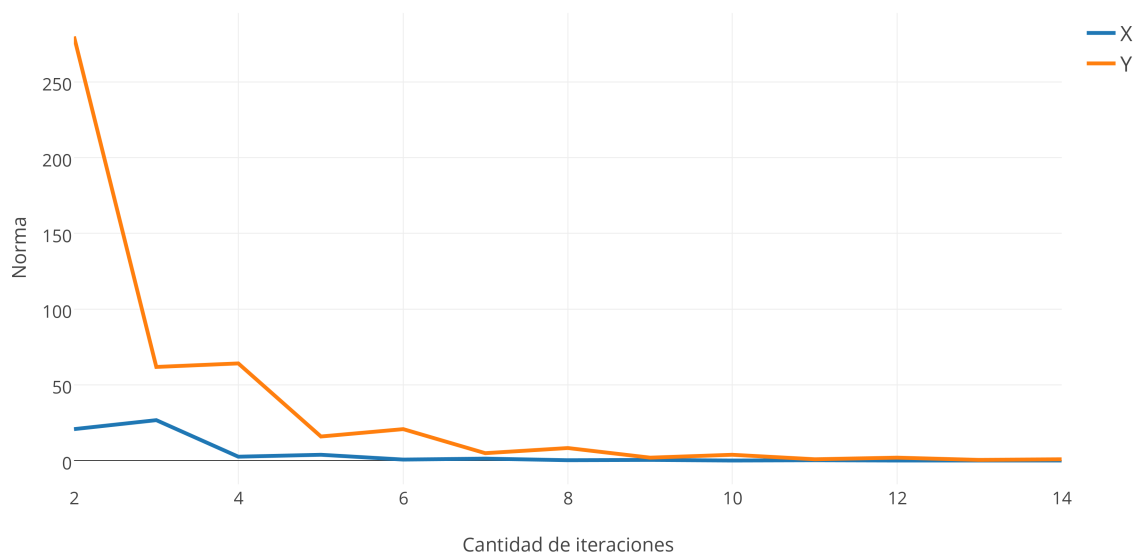


Figura 10: Variación de la norma de X e Y en cada iteración para la Red Stanford

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 analizar 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:**



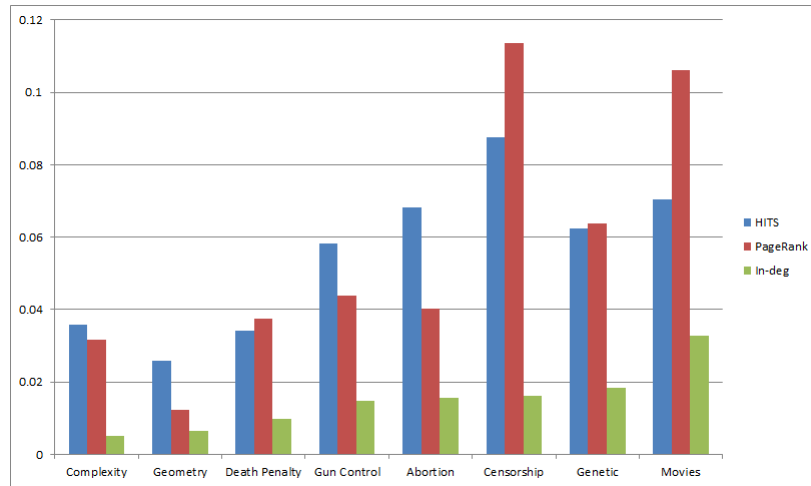


Figura 11: Medición de los tres métodos para redes medianas

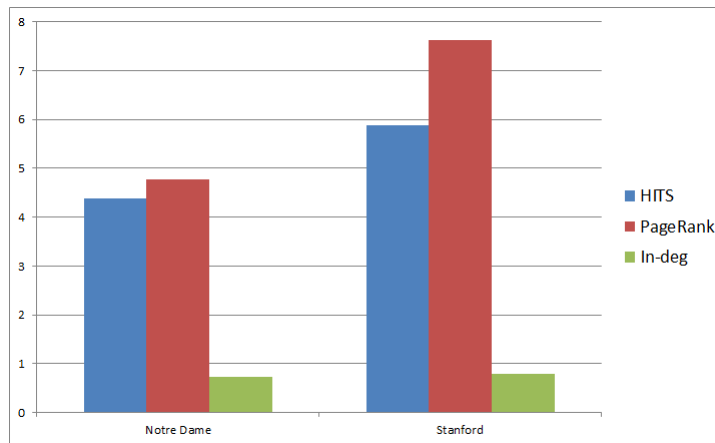
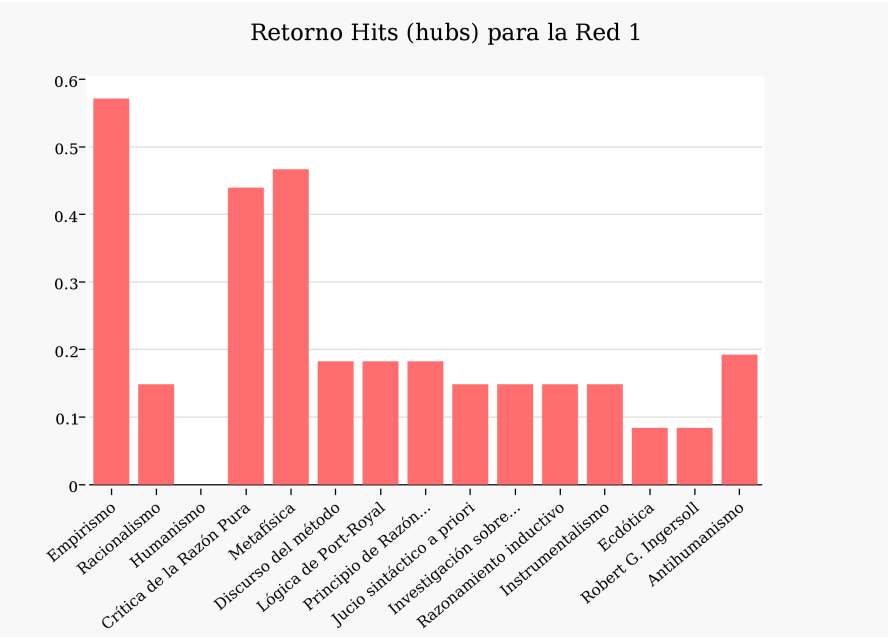
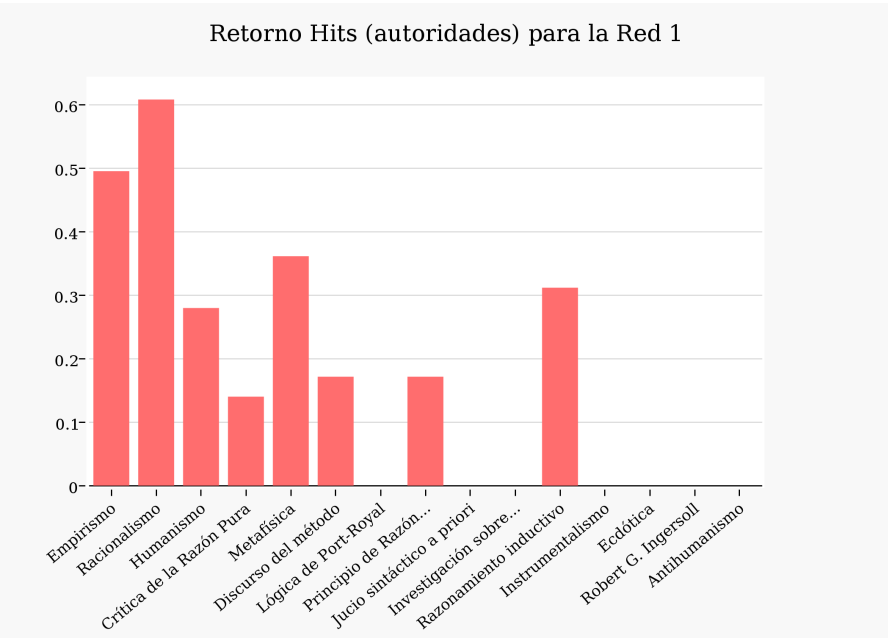


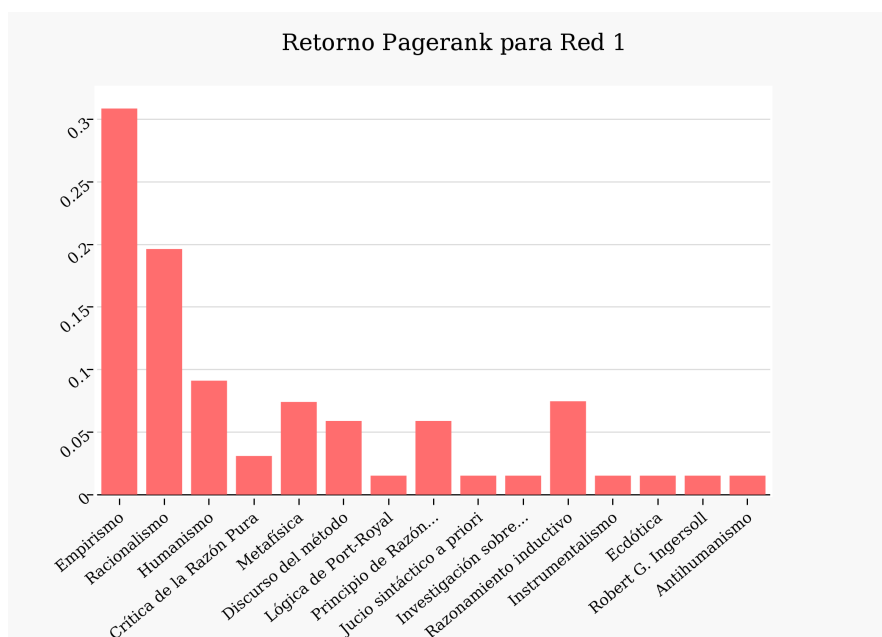
Figura 12: Medición de los tres métodos para redes grandes

- 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

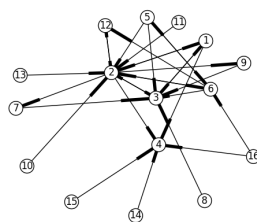
Si analizamos los resultados de Hits para organizar las autoridades, encontramos en este caso más páginas coincidentes con las priorizadas con el algoritmo de Pagerank. De hecho, las cinco páginas a las que éste último método concede las primeras posiciones, son las mismas que pueden encontrarse en los 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 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 “híbridos” entre hubs y autoridades, no nos permiten emitir conclusiones claras al respecto.

Dicho esto, si descartamos 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 directamente 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 *hipótesis 1*, le agregamos una web nueva ("*EscuelasFilosóficas*"), 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.

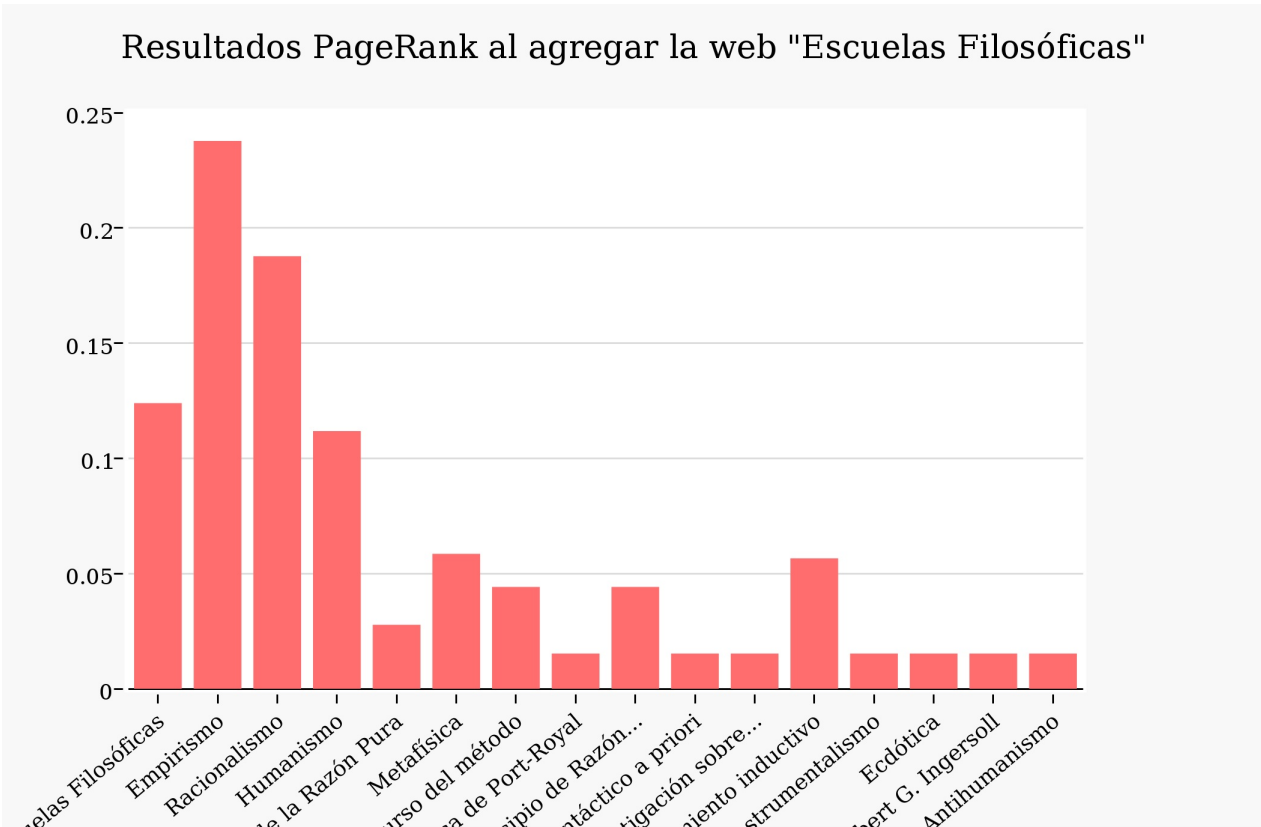
#### Red 2:

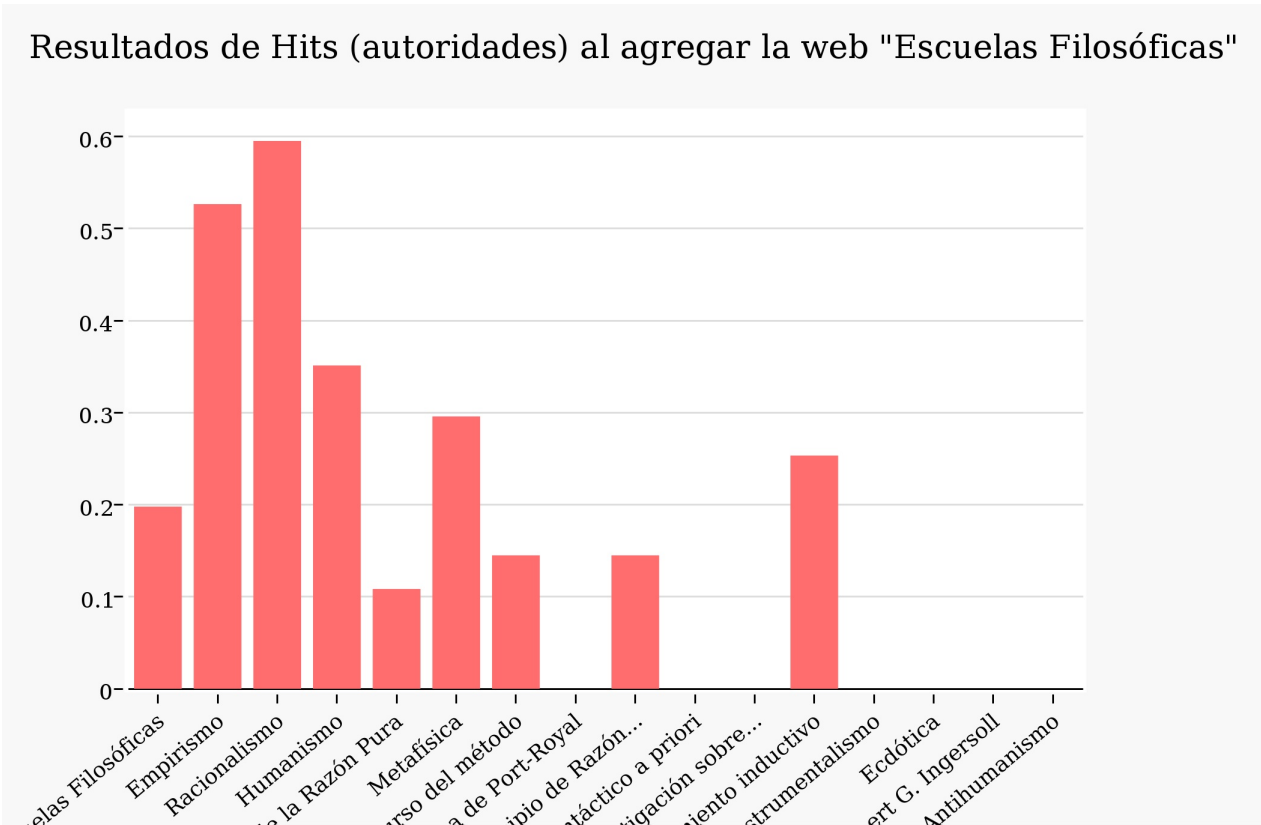
- 1 Escuela Filosófica
- 2 Empirismo
- 3 Racionalismo
- 4 Humanismo
- 5 Crítica de la razón pura
- 6 Metafísica
- 7 Discurso del método
- 8 Lógica de Port-Royal
- 9 Principio de razon suficiente
- 10 Juicio sintáctico a priori
- 11 Investigación sobre el entendimiento humano
- 12 Razonamiento inductivo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

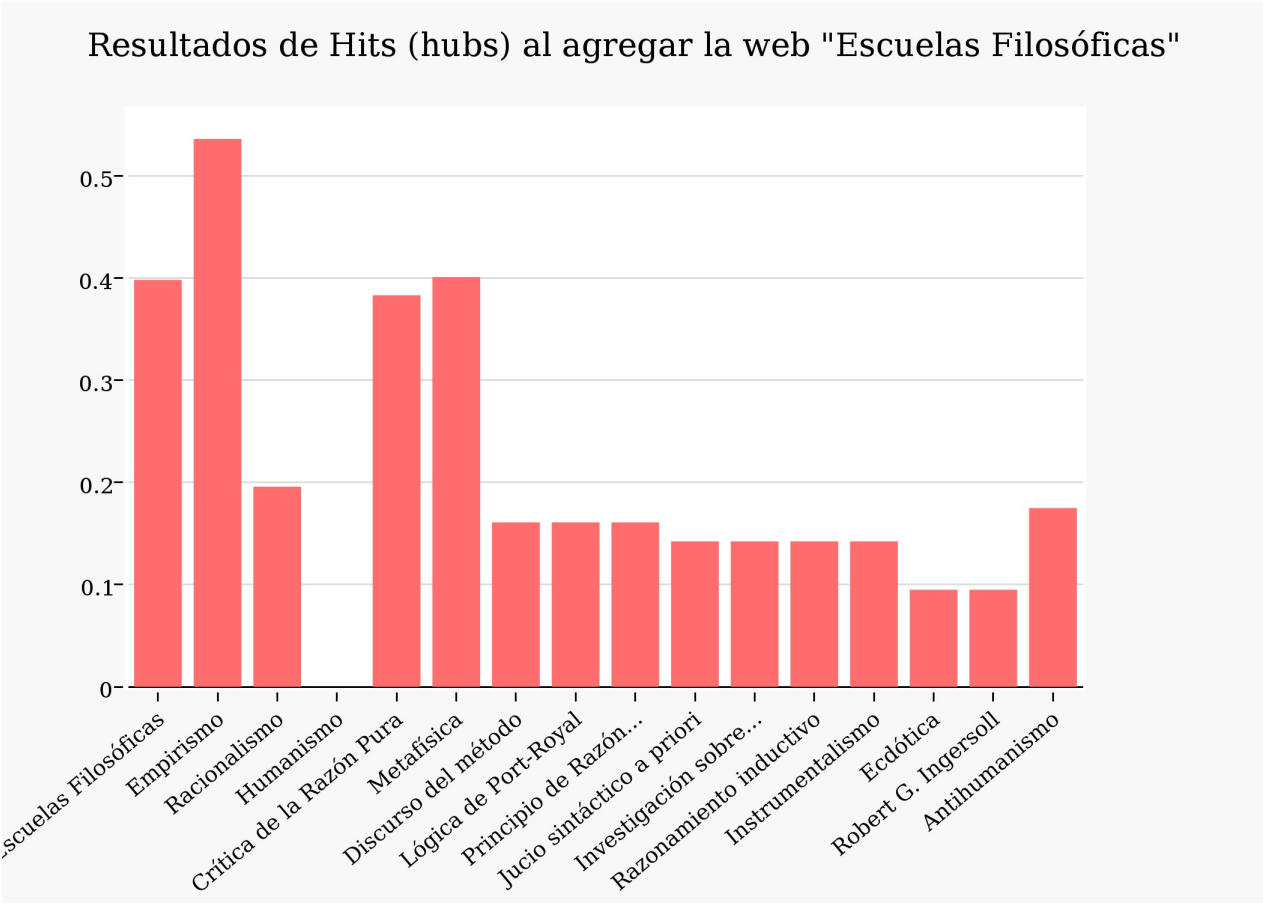
Si se contrasta la figura 3.4 con la figura 3.4 (propia de este experimento) se puede ver que el máximo valor de hub no se realiza en el peso del sitio agregado, sino en los que eran hubs destacados en la red original.

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.











**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 disminuido 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 Levi-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 comparación 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 todas 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.

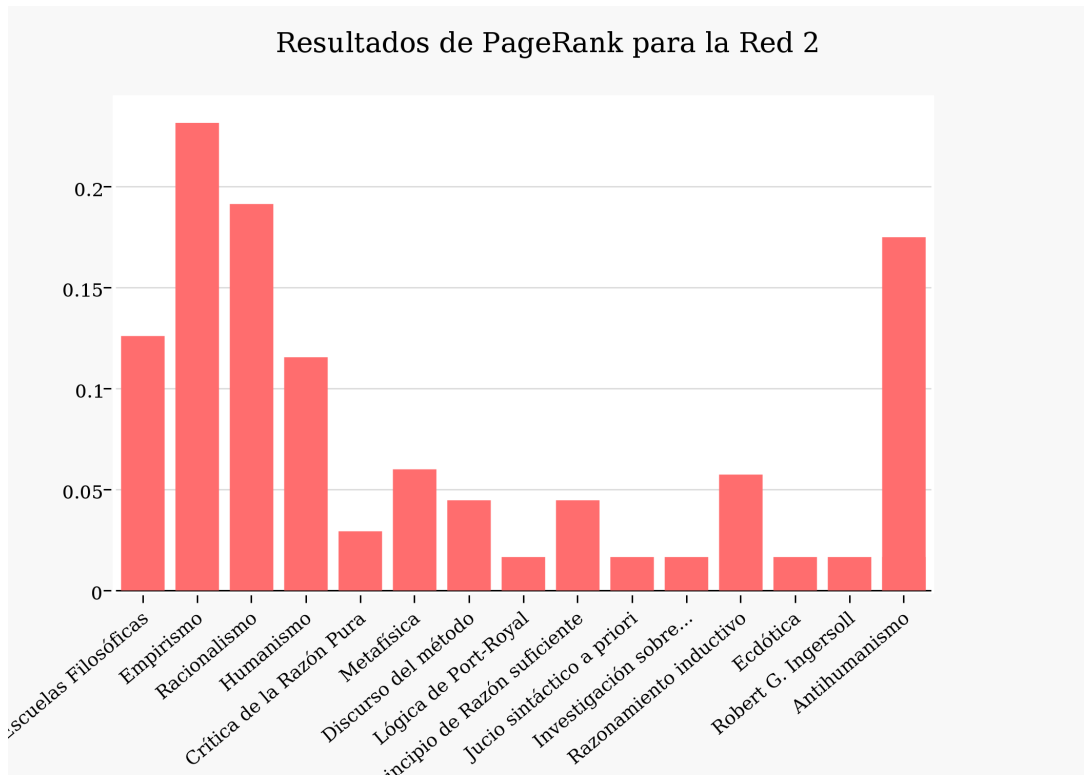
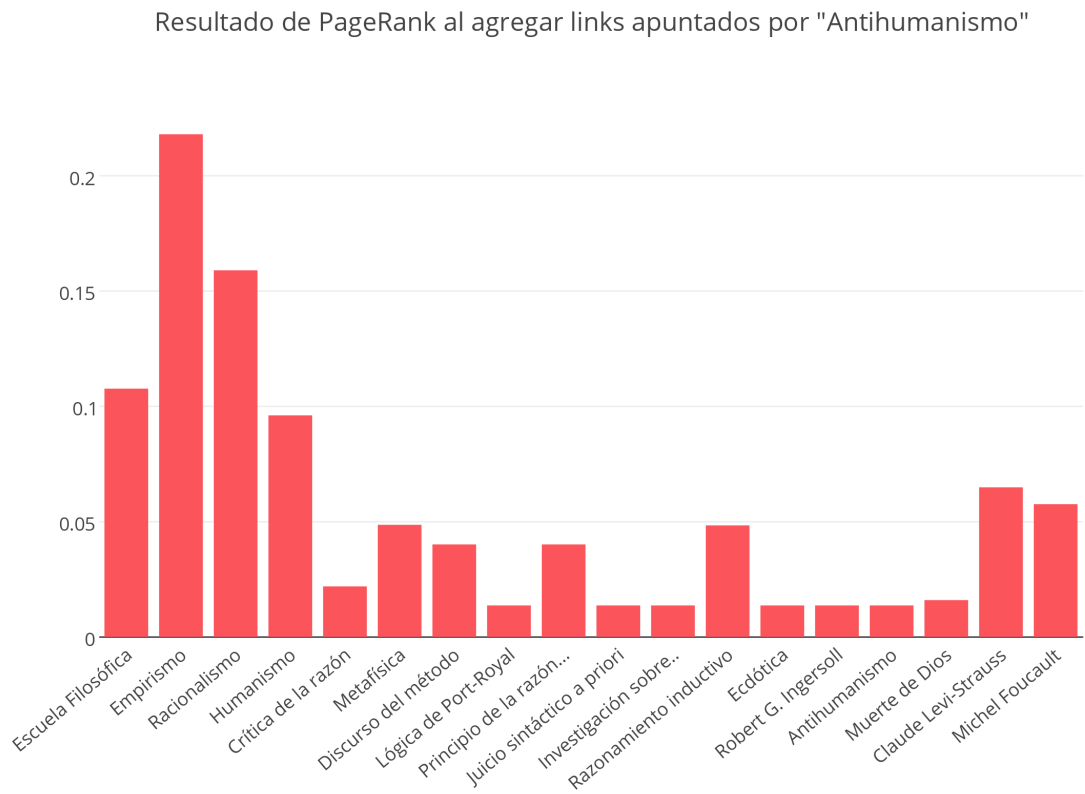
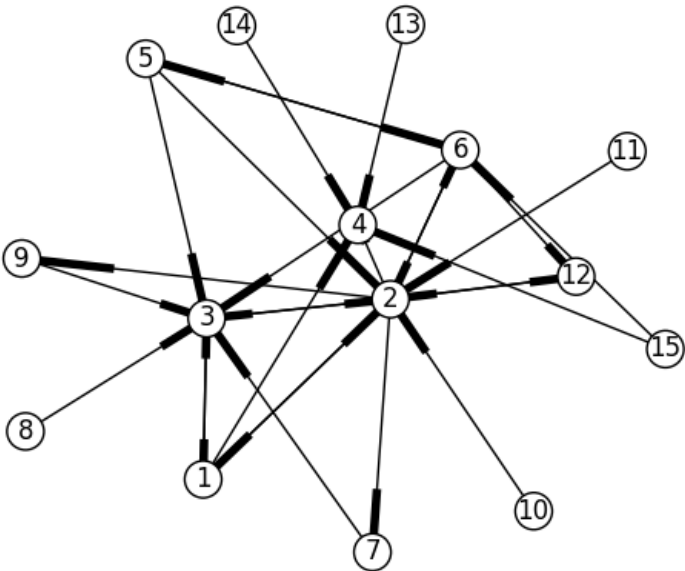


Figura 13: Descripción de la figura





**Hipótesis 4:** *Al aumentar la probabilidad de que el navegante aleatorio se transporte, crece también la proporción de tiempo que se espera que permanezca en páginas 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 HIPO-  
TESIS. QUE PONGO? (ES LA Q SE AGREGA "MATRIZ.<sup>A</sup> 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.movedatabase.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

Esta sección debe contener las conclusiones generales del trabajo. Se deben mencionar las relaciones de la discusión sobre las que se tiene certeza, junto con comentarios y observaciones generales aplicables a todo el proceso. Mencionar también posibles extensiones a los métodos, experimentos que hayan quedado pendientes, etc.

## **5. Apéndices**

### **5.1. Apéndice A**

## 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 ← fil + 1 to cantidaddefilas do
  | ptr_fil[i] ++
end

```

## 5.3. Apéndice C

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

$$A = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 3 & 4 & 0 & 8 \\ 0 & 0 & 9 & 0 \\ 10 & 11 & 0 & 0 \end{pmatrix} \quad (1)$$

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]

```

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

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

Es importante incluir referencias a libros, artículos y páginas de Internet consultados durante el desarrollo del trabajo, haciendo referencia a estos materiales a lo largo del informe. Se deben citar también las comunicaciones personales con otros grupos.

PONER ACA LOS PAPERSSSSSSSSSSSS