The Web as a graph (2000)

R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, E. Upfal

Presentador: Andrés Abeliuk

Motivación

- Páginas web:
  - Distintos lenguajes, estilos, motivos …
  - Contienen verdades, mentiras, propaganda, tonterías ...
- **Hyperlinks** dotan la web con una estructura adicional.
  - Red de links son ricos en información latente
- Paper estudia el grafo inducido por los links entre páginas web.

Web Graph

- **Nodos** ⇔ páginas html estáticas.
- **Aristas dirigidas** ⇔ links.
- Estimaciones (2000):
  - $O(10^8)$ nodos
  - 7 links a otras páginas por nodo en promedio

  $\Rightarrow$ Billones de links.

Motivos de estudio

- Explotar estructura del grafo ha mejorado:
  - **Búsqueda en la Web**
  - Mejores algoritmos de clasificación de tópicos.
  - **Algoritmos para enumerar ciber-comunidades emergentes.**

- Hiperlinks representan una fuente de información sociológica.

Observaciones previas

- Para cada tópico:
  - Set **“authoritative”** de páginas enfocadas en el tópico.
  - Set **“hub”** de páginas que contienen links relevantes al tópico.

- Esta observación motivó:
  - Algoritmo de búsqueda que obtiene páginas de alta calidad.
  - Algoritmo que enumera todos los tópicos representados en sets densos de authority/hub.

The HITS algorithm

- **H**yperlink-**I**nduced **T**opic **S**earch
- Sampling step
  - Construye una colección de miles de páginas, con alta probabilidad de ser “authorities”
- Weight-propagation step
  - Rankea páginas por su calidad como hubs y authorities.
  - Retorna páginas con mayor puntaje.

Sampling Step

- Obtener 200 paginas de un buscador tradicional. (root set)
- Se expande el “root set”, incluyendo:
  - Todas las páginas que apuntan al “root set”
  - Todas las páginas a las que el “root set” apunta.
- “Base set” tipicamente tiene 1000-3000 páginas.
  - No se consideran links entre dos páginas del mismo sitio.

Weight-propagation step

- Cada página p tiene un par de pesos (x_p,y_p).
  - **x_p** es el peso de autoridad e **y_p** el de "hub".
  - Inicialmente todos parten e 1

- Si una página es apuntada por varios hubs, se debe incrementar su peso de autoridad

$$x_p = \sum_{q|q\to p} y_p$$

- Su dual

$$y_p = \sum_{q|p\to q} x_q$$

Weight-propagation step

- De forma mas compacta e ilustrando el algoritmo de puntajes:
  - Se define la matriz de adyacencia A de *n x n* cuya (i,j) entrada es 1 sii i->j, y 0 de lo contrario.
  - Los pesos se escriben como vectores, $x=(x_1,\ldots x_n)$ e $y=(y_1,\ldots y_n)$.
- Luego las reglas de actualización son:

$$x \leftarrow A^T y$$

$$y \leftarrow Ax$$

Weight-propagation step

- Reiterando la actualización se tiene:

$$x \leftarrow A^T y \leftarrow A^T Ax \leftarrow (A^T A)x$$

$$y \leftarrow Ax \leftarrow AA^T y \leftarrow (AA^T)y$$

- Vector x después de múltiples iteraciones es aplicar potencias de (A^TA).
  - Normalizado converge a los vectores propios de (A^TA).

- **Pesos son intrínsecos de la colección de páginas**

Resultados

- Propiedades interesantes:
  - Después de encontrar “root set” con la consulta, se ignora completamente el contexto textual.
  - HITS es puramente basado en los links.

- Consulta=“search engine”
  - Responde= Yahoo!, Excite, Magellan, Lycos, AltaVista.
  - Ninguna de las páginas anteriores contiene (al realizar el experimento) la frase “search engine”

"search engine" Search

About 221,000,000 results (0.20 seconds) Advanced search

Everything
Images
Videos
News
More

Any time
Latest

All results
Related searches
Wonder wheel
Page previews

More search tools

Dogpile Web Search ☆
Dogpile.com makes searching the Web easy, because it has all the best search engines piled into one. Go Fetch!
SearchSpy - White Pages - Preferences - Yellow Pages
www.dogpile.com/ - Cached - Similar

AltaVista ☆
AltaVista provides the most comprehensive search experience on the Web!
Images - English - Submit a Site - Background
www.altavista.com/ - Cached - Similar

Web **search engine** - Wikipedia, the free encyclopedia ☆
A web **search engine** is designed to search for information on the World Wide Web and FTP servers. The search results are generally presented in a list of ...
en.wikipedia.org/wiki/Web_**search_engine** - Cached - Similar

Google Custom Search - Site search and more ☆
Enable autocompletions for your **search engine** to help your users get to the right results faster. Learn more with our developer documentation, FAQs and ...
www.google.com/cse/ - Cached - Similar

Bing ☆
Bing is a **search engine** that finds and organizes the answers you need so you can make faster, more informed decisions.
www.bing.com/ - Cached - Similar

Ask.com - What's Your Question? ☆
Ask.com is the #1 question answering service that delivers the best answers from the web and real people - all in one place.
www.ask.com/ - Cached - Similar

DuckDuckGo ☆
This site requires JavaScript. About | Settings | Goodies. © 2010 - Privacy & Terms.
duckduckgo.com/ - Cached - Similar

Mejoras

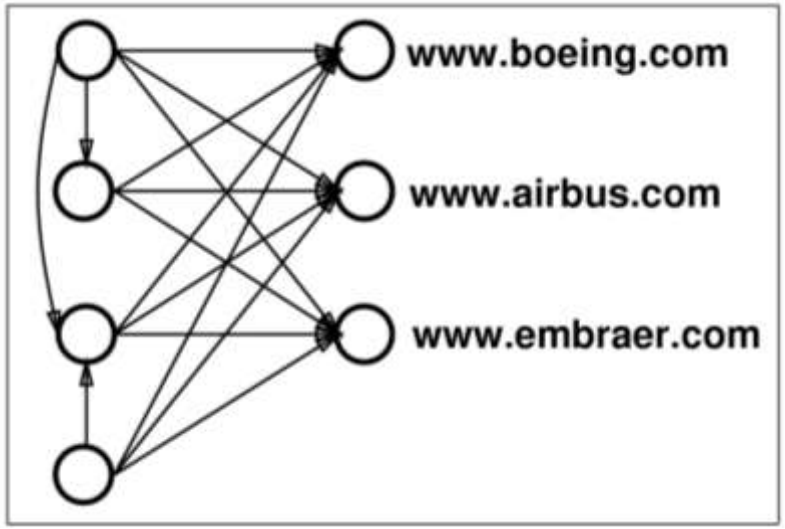
- El algoritmo se ha generalizado permitiendo que la matriz A no siga siendo booleana.
  - Con esta modificación se puede considerar el contenido de las páginas, los dominios donde reside, etc.

- Se mantiene representación hub/authority como componentes de los vectores propios.

Trawling algorithm

- Bipartite core $C_{i,J}$:
  - Grafo con i+j nodos que contiene al menos un clique bipartito $K_{i,J}$ como subgrafo.

- Ejemplo
  - $C_{4,3}$
  - Posible “ciber-comunidad” de aficionados de aviones



Comunidades

- Tales comunidades emergen cuando varios hubs apuntan a las mismas páginas authorities.
  - En mayoría de casos hubs no co-citan todos los authorities.

- Hipótesis débil:
  - Toda comunidad tiene un core bipartito $C_{i,j}$ para valores no triviales de i, j.

Experimentos

- Se enumeraron $C_{i,j}$´s
  - $3< i <9$, $3< j< 20$

- Resultados
  - La web tiene cientos de miles de esos cores
  - Aparentemente una fracción diminuta de estas son coincidencias.
  - La mayoría corresponde a comunidades con un tópico claro.

Mediciones

i	j	Cores	Diverse cores
3	3	89565	38887
3	5	70168	30299
3	7	60614	26800
3	9	53567	24595
4	3	29769	11410
4	5	21598	12626
4	7	17754	10703
4	9	15258	9566
5	3	11438	7015
5	5	8062	4927
5	7	6626	4071
5	9	5684	3547
6	3	4854	2757
6	5	3196	1795
6	7	2549	1425
6	9	2141	1206

Table 1: Number of cores enumerated during the pruning phase of trawling.

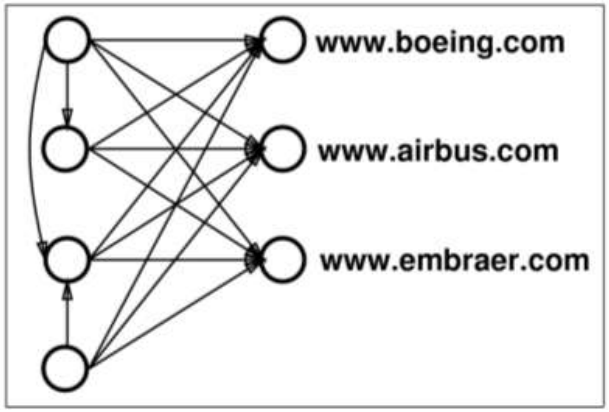
Trawling algorithm

- Implementación de fuerza bruta es impracticable.
  - 10^{40} posibilidades de $C_{2,3}$ en un grafo de 10^8 nodos
  - Se requiere acceso aleatorio, lo que significa traer todo el grafo a memoria principal.

Elimination-generation paradigm

- Iteraciones secuenciales sobre el grafo consisten en:
  - Guardar una nueva versión de los datos para la próxima pasada.
  - Colecta metadata que reside en memoria principal y sirve como un estado para la iteración.
  - Se hacen sorts en los datos para cambiar el orden de los escaneos.
    - Sorts son el principal costo del algoritmo.
  - Se realizan operaciones de **eliminación** y **generación**.

Elimination



- Condiciones necesarias pero no suficientes para que un nodo pertenezca a $C_{i,j}$
  - Cualquier nodo con grado-entrada menor a j no puede participar en el lado derecho del core
  - Nodos con grado-salida menor a i no pueden participar en el lado izquierdo del core.

- Nodos que no cumplen eso se eliminan.

Generation

- Obtiene los cores a los que pertenece un nodo
  - Si *u* es un nodo con in-degree i
    u pertenece a un $C_{i,j}$ sii los nodos que apuntan a *u* tienen una intersección de vecindad de tamaño al menos j.

- No es costoso verificar esta propiedad

- Identifica cores o prueba que estos no pueden existir.

- Terminado, se puede eliminar el nodo ya que se han enumerado todos los cores potencialmente interesantes.

Trawling algorithm

- Si los nodos aparecen en orden arbitrario no es posible aplicar el filtro de eliminación fácilmente.
  - Por eso se ordenan por destino o origen.
- Después de un paso de elimination/generation los nodos restantes tienen menos vecinos, presentando nuevas oportunidades en la próxima iteración.
- Se itera hasta que el progreso sea insignificante.

Complejidad

- Observaciones:
  - El in/out-degree de cada nodo disminuye o se mantiene en cada fase.
  - La fase de generación es lineal en el tamaño del grafo y en el número de cores enumerados.
  - En la practica, la fase de eliminación, rápidamente elimina un gran porcentaje de nodos.

Clasificación

- Dado un set de categorías, asignarle a un documento una de las categorías.
- Problema difícil en general.
  - Más aún en la web
- El desafío es ocupar la información de los links.

Hyperclass algorithm

- En vez de solo considerar la página *p*, se considera la vecindad de *p* para la clasificación.
- El algoritmo parte asignando etiquetas a toda página *p* del grafo, basado solamente en los términos de *p.*
- Luego las etiquetas se actualizan basándose en los términos de *p,* y los términos y etiquetas (parciales) de los vecinos de *p*.
- Las actualizaciones se basan en métodos estadísticos robustos.
- Experimentalmente hyperlink aumenta considerablemente la exactitud de clasificación.

Mediciones

- In/out-degrees sigue una ley de potencias.
  - El porcentaje de páginas con in/out-degree *i* es $1/i^x$ para algun x>1

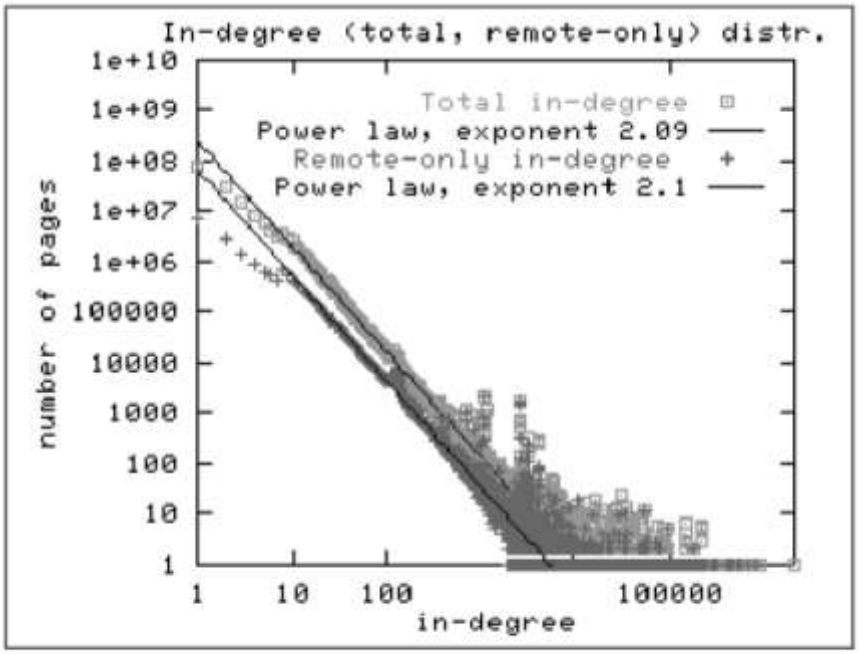


Figure 2: In-degree distribution.

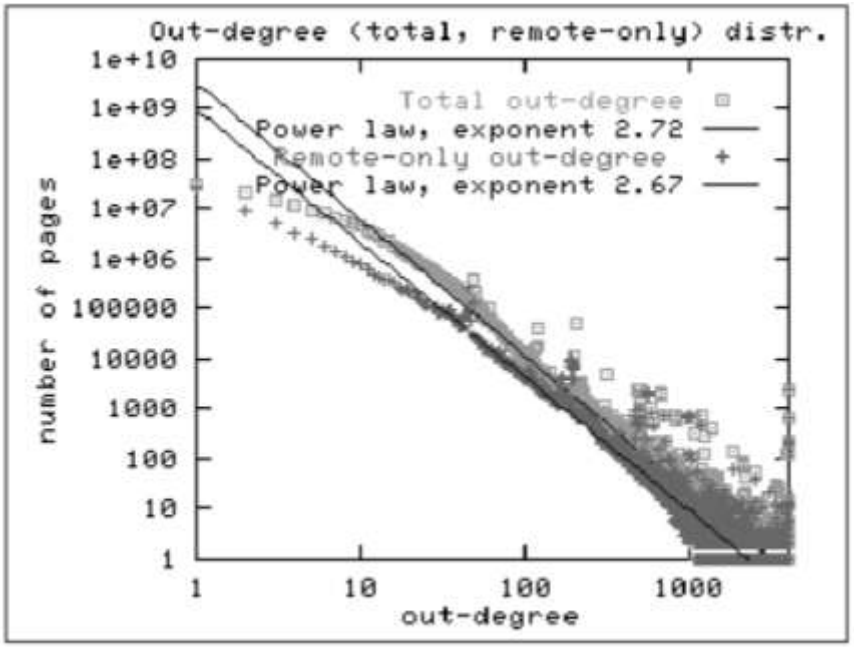


Figure 3: Out-degree distribution.

Mediciones

i	j	Cores	Diverse cores
3	3	89565	38887
3	5	70168	30299
3	7	60614	26800
3	9	53567	24595
4	3	29769	11410
4	5	21598	12626
4	7	17754	10703
4	9	15258	9566
5	3	11438	7015
5	5	8062	4927
5	7	6626	4071
5	9	5684	3547
6	3	4854	2757
6	5	3196	1795
6	7	2549	1425
6	9	2141	1206

Table 1: Number of cores enumerated during the pruning phase of trawling.

Componentes conexos

- Experimento realizado sobre 200 millones de nodos y 1,5 “billion” links.
- **Componente débilmente conexo** sii existe un camino de ida o vuelta entre pares de nodos.
  - El más grande obtenido fue de 186 millones nodos
- **Componente fuertemente conexo** sii existe un camino directo entre pares de nodos.
  - El más grande de 56 millones, seguido por 50 mil
- Número de componentes sigue una ley de potencias.

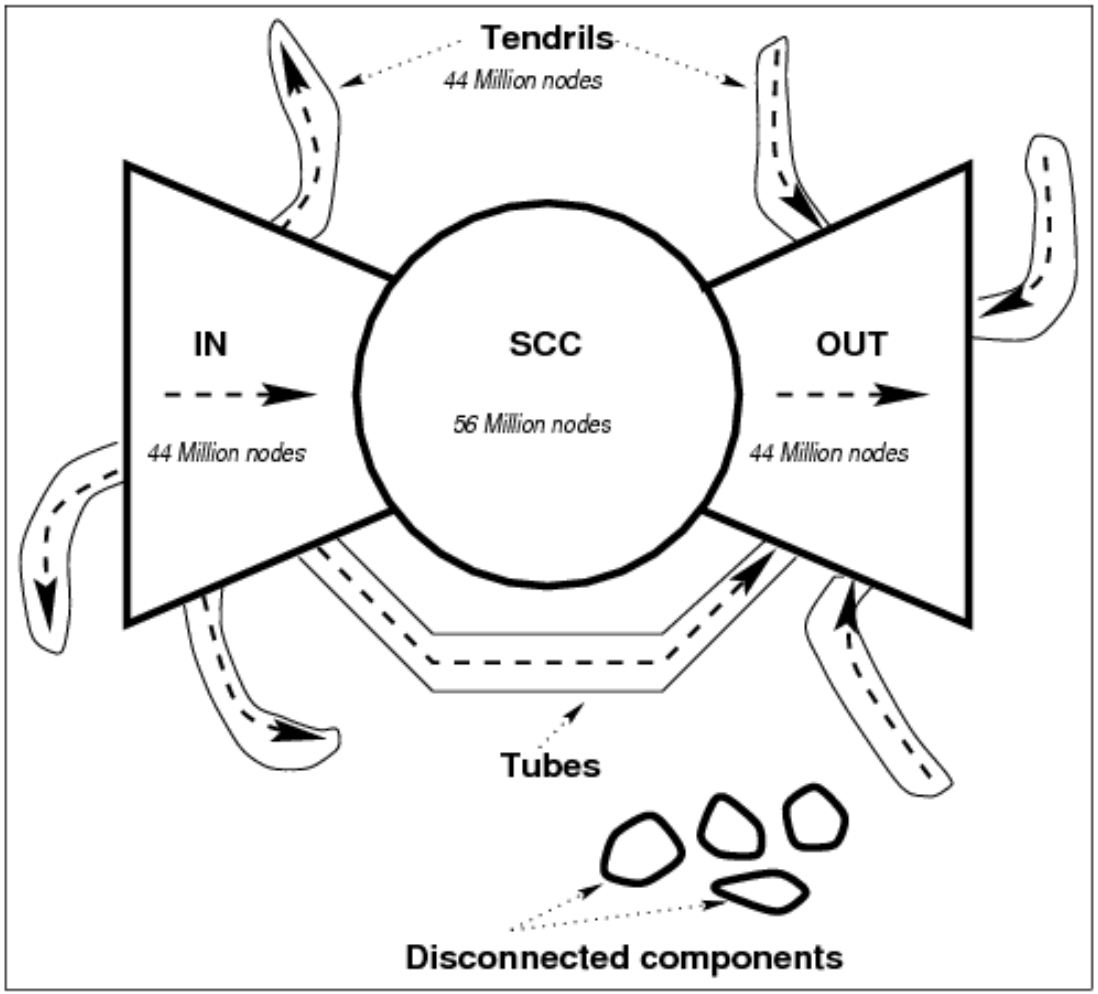


Figure 4: The web as a bowtie. SCC is a giant strongly connected component. IN consists of pages with paths to SCC, but no path from SCC. OUT consists of pages with paths from SCC, but no path to SCC. TENDRILS consists of pages that cannot surf to SCC, and which cannot be reached by surfing from SCC.

Diámetro de la web

- Si *u* pertenece a IN ∪ SCC, y *v* en SCC ∪ OUT un camino de *u* a *v* existe, de lo contrario con casi certeza no existe.
- La probabilidad de que *u* pertenece a IN ∪ SCC es aprox. ½, igualmente para para *v* en SCC ∪ OUT
- Luego la probabilidad que sucedan esto eventos independientes es ¼
- Por lo tanto para un 75% de páginas *u* y *v* no existe un camino.

Edge type	In-links	Out-links	Undirected
Avg. Conn. Dist.	16.12	16.18	6.83

Table 2: Average distances on the web, conditioned on existence of a finite path.

Modelo de la web

- Permite modelar varias propiedades estructurales de la web:
  - Distribución de grados, cores, etc.
- Un modelo de grafo de la red es una herramienta analítica.
  - Permite calcular complejidad de un algoritmo.
- Un buen modelo puede ser una herramienta aclaratoria del fenómeno.
- Puede actuar como una herramienta predictiva.
  - Sugerir propiedades emergente en la web de mañana.

Random Graphs

- $G_{n,p}$: Grafo aleatorio de *n* nodos donde cada posible arco es incluido con probabilidad *p*.

- Primero notar que $G_{n,p}$ no genera páginas extremadamente populares.

- La distribución de los grados sigue una binomial
  - Los experimentos muestran una ley de potencias.

Random Graphs

- En $G_{n,p}$ los links son independientes.
  - En la web, los links de una misma página tienen contenidos relacionados.

- Como consecuencia de eso, los números de cores tampoco coinciden con la web.
  - El número de cores $C_{i,j}$ esperado es:

$$\binom{n}{i}\binom{n}{j}p^{ij}$$

  - Insignificante para ij>i+j

Propiedades deseables

- Los ricos se enriquecen aún más.
  - Nuevos links apuntan a paginas con mayor in-degree
- Out-links correlacionados
  - Dado un link *h=(u,v),* los destinos de otros links desde *u* debiesen revelar información de *v.*
- In-links correlacionados
  - Dado un link *h=(u,v),* los orígenes de otros links con destino *v* debiesen revelar información de *u.*

Metas del modelo

- Fácil de describir y que sea natural.
- Reflejar el comportamiento de creación de contenido en la web.
  - Al menos capturar estructura de la información agregada .
- Estructura del grafo debiese reflejar las mediciones de la web.
- No debiese requerir set de “tópicos” estáticos a priori, sino que emerger naturalmente.
  - Importante, ya que es difícil caracterizar el set de tópicos en la web.

Random copying

- Se quiere capturar la siguiente intuición:
  - Algunos creadores de páginas notaran una interesante relación entre ciertas páginas y linkearan a ellas, pero
  - La gran mayoría, estará interesado en estos ya representados tópicos, y coleccionara links a paginas de estos tópicos.

El modelo

- Caracterizado por 4 procesos estocásticos responsables por la creación y eliminación de nodos y aristas (C_n, C_e, D_n, D_e).
- Cada uno es un proceso de tiempo discreto.

- $G_{n,p}$ es equivalente a:
  - C_v crea *n* nodos en el tiempo 0
    C_e crea cada arista con probabilidad uniforme *p.*
    D_n y D_e vacios.

Un ejemplo más real

- C_n en el tiempo *t* crea un nodo con probabilidad $\alpha_c(t)$ y D_n borra un nodo con probabilidad $\alpha_d(t)$.

- C_e en cada paso, agrega links todos los nuevos nodos agregados y algunos ya existentes.
  - Por cada nodo elegido se elige al azar el número de aristas *k* a agregar.
  - Con una probabilidad fija β se agregan *k* links a destinos elegidos uniformemente al azar.
  - Con la probabilidad restante se agregan *k* aristas simulando el random copying.

Un ejemplo más real

- Random copying
  - Se elige un nodo *v* con alguna distribución
  - Se copian *k* links(destinos) elegidos aleatorios de *v* y se ponen en la página como origen.
  - Si *v* contiene menos de *k* links, se elige otro página hasta completar *k*.

- D_e en el tiempo *t* con probabilidad δ(t) borra una arista con alguna distribución.

Ley de distribuciones

- El modelo anterior levemente simplificado sigue una ley de potencias
  - La fraccion de páginas con in-degree *i* convergue a $1/i^x$
- Sea $p_{i,t}$ la fraccion de nodos en el tiempo t con in-degree *i*

$$\forall i \ \lim_{t\to\infty} E[p_{i,t}] = \hat{p}(i) exists,$$

$$\lim_{t\to\infty} i^{\frac{2-\alpha}{1-\alpha}} \hat{p}(i) = \frac{1}{1+\alpha}$$

Trabajo futuro

- Cores no necesariamente son las únicas clases de subgrafos interesantes en la web.
  - Cliques, stars, arboles, etc
- Propiedades y evolución de los grafos modelados anteriormente.
- Como se analizan algoritmos en estos grafos?
- Que se puede inferir del proceso sociologico de crear paginas de forma distribuida?
- Que otras estructuras más finas, se pueden encontrar en el mapa de la web?