Gestión de Información en la Web
Máster en Ingeniería Informática
Parte II: Redes Sociales On-line

Seminario 3: Pagerank

Oscar Cordón García
Dpto. Ciencias de la Computación e Inteligencia Artificial
ocordon@decsai.ugr.es

INTRODUCCIÓN: Búsqueda de Información en la Web (1)

La búsqueda en la web implica ejecutar una consulta en un motor de búsqueda. Como resultado, ese motor devuelve una lista de páginas web que “casan” con la consulta realizada

Puesto que la WWW tiene un número enorme de páginas, esa lista de páginas devuelta suele ser un conjunto muy grande

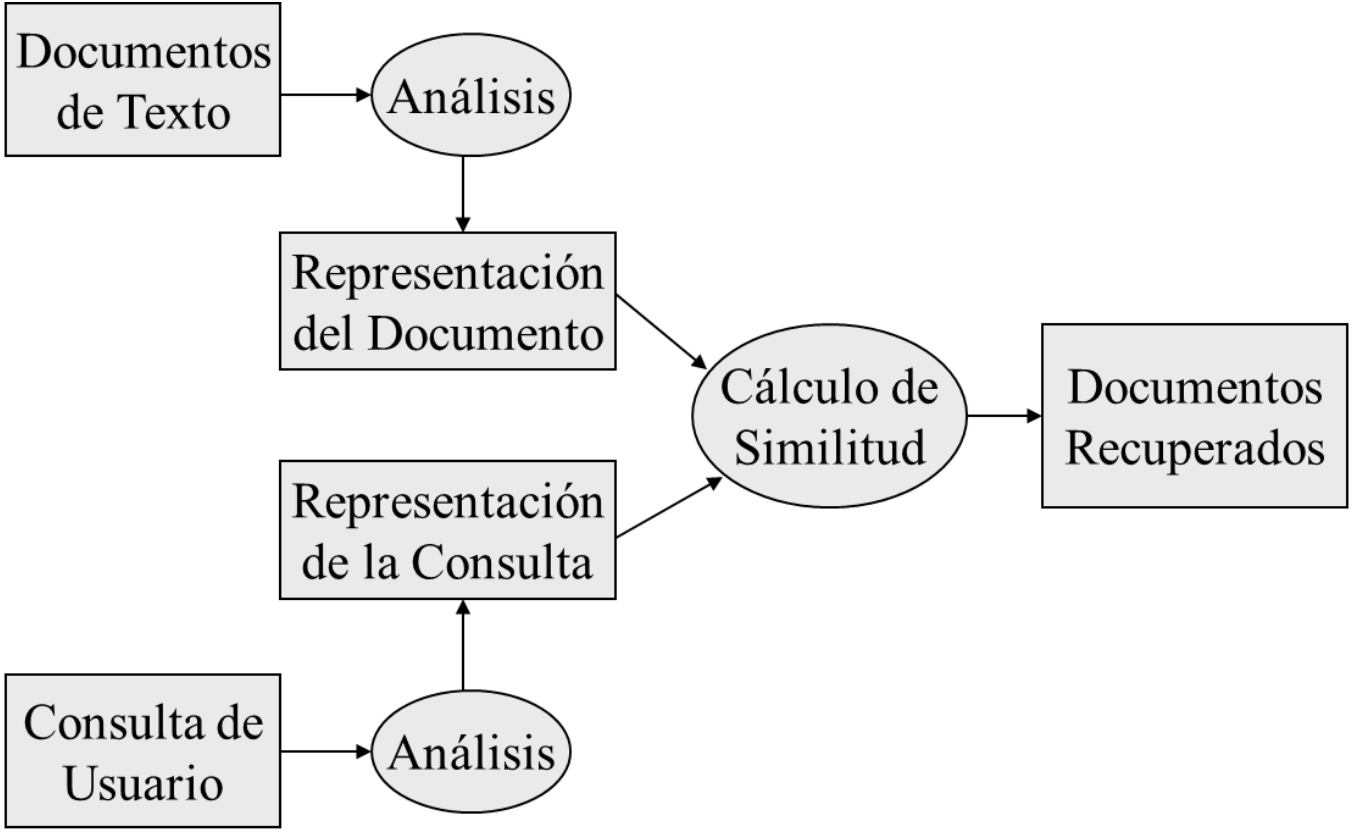
Por esta razón, es fundamental que **la lista esté ordenada**, proporcionando los resultados más relevantes en las primeras posiciones

Los métodos de recuperación de información clásicos solo se basan en el texto, ordenando los resultados por la similitud del **contenido**. La determinación de la importancia de una página web es una medida subjetiva

Los algoritmos de análisis de enlaces consideran la naturaleza hipertextual de la web (**los enlaces entre páginas**) para definir ese orden, complementando a los algoritmos clásicos de búsqueda

INTRODUCCIÓN: Búsqueda de Información en la Web (2)

La mayoría de los sistemas de búsqueda en la Web se basan en los Sistemas de Recuperación de Información clásicos (Booleano, **vectorial**, …)



INTRODUCCIÓN: Búsqueda de Información en la Web (3)

Estos sistemas no tienen en cuenta la diferencia estructural existente entre un documento web y uno de texto plano:

- Estructura interna: Un documento web presenta una estructura semántica mediante etiquetas HTML
- Estructura externa: Una página web puede estar vinculada a muchas otras y viceversa. Esa vinculación puede indicar una relación entre documentos

A finales de los 90 se propusieron varios sistemas de búsqueda web que aprovechan la estructura externa de un documento web como el **PageRank** de Google

Ese algoritmo se basa en el análisis de enlaces de las redes sociales. En concreto, implementa una **medida de Centralidad de vector propio** en el grafo dirigido de la WWW para medir el **Prestigio de rango** de las páginas

ANÁLISIS DE ENLACES: PageRank y HITS

En 1998 se propusieron los dos primeros algoritmos de análisis de enlaces y búsqueda en la web:

- **HITS**: Presentado por Jon Kleinberg en Enero de 1998 en el Ninth Annual ACM-SIAM Symposium on Discrete Algorithms
- **PageRank**: Presentado por Sergey Brin y Larry Page en Abril de 1998 en la Seventh International World Wide Web Conference (WWW7)

Ambos se basan en los conceptos básicos del **Análisis de Citas en publicaciones científicas**, un área de la **Bibliometría**

Las ideas principales de ambos algoritmos son similares. Mientras que PageRank define un orden estático entre las páginas web en función de su prestigio, HITS establece un orden dinámico que depende de la consulta concreta

PageRank está más extendido al ser el algoritmo empleado por Google

PAGERANK: Introducción (1)

PageRank fue desarrollado por Larry Page (de ahí su nombre) y Sergey Brin

Surgió del proyecto de ambos para proponer un nuevo tipo de motor de búsqueda que arrancó en 1995 y derivó en un primer prototipo funcional en 1998. Poco después fundaron Google

Wikipedia: "PageRank es una marca registrada y patentada por Google el 9 de enero de 1999 que ampara una familia de algoritmos utilizados para asignar de forma numérica la relevancia de los documentos (o páginas web) indexados por un motor de búsqueda"

Brin S., Page L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. Proc. 7th International Web Conference (WWW 98), 1998 y **Computer Networks and ISDN Systems 30:1-7 (1998) 107-117** (versión extendida: http://infolab.stanford.edu/~backrub/google.html)

Page L., Brin S., Motwani, R., Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report. Stanford InfoLab (1998-99)

PAGERANK: Introducción (2)

Pagerank define un orden estático en la WWW. Se calcula un valor *PR* basado en la medida de prestigio en redes sociales para cada página, que no depende de la consulta concreta

El PageRank confía en la estructura democrática de la web, considerando su vasta estructura de enlaces como un indicador de la importancia de una página individual

En esencia, interpreta un hiperenlace de una página *x* a otra *y* como un voto de la página *x* por la página *y*

Sin embargo, no sólo cuenta el número absoluto de votos sino que los pondera por la importancia de la página que efectúa el voto

Así, no es cuestión sólo de las páginas que te apuntan, sino de cuántas páginas apuntan a ellas y de su importancia: definición recursiva → Centralidad de vector propio = Prestigio de rango en la red de la WWW

PAGERANK: Análisis de enlaces web (1)

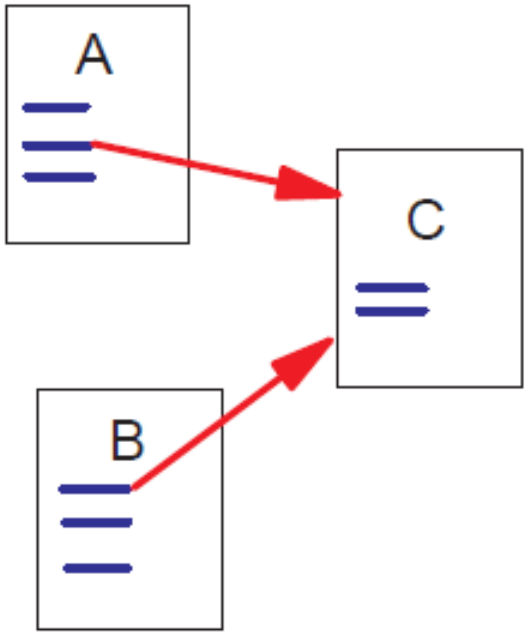


Figure 1: A and B are Backlinks of C

- La web tiene enlaces de entrada y de salida
- En 1998, la web ya presentaba 150 millones de páginas y 1.7 billones de enlaces
- PageRank se basa en considerar la red dirigida de la web y los conceptos de análisis de citas y centralidad de vector propio:
  - Las páginas que reciben más enlaces son más “importantes” que las que reciben pocos enlaces

PAGERANK: Análisis de enlaces web (2)

Las páginas web varían muchísimo en lo que respecta al número de hiperenlaces recibidos:

Ejemplo: www.joe-schmoe.com contra www.stanford.edu:

- www.stanford.edu recibe 23400 enlaces
- www.joe-schmoe.com recibe 1 enlace

Intuitivamente, una página es tanto más importante cuántos más enlaces recibe. Por ejemplo, en 1998 la página de Netscape recibía 62804 enlaces

Sin embargo, eso no es suficiente y puede ser contraintuitivo. ¿Qué pasa si una página recibe un solo enlace pero viene de Yahoo? ¿Es menos importante que otra que reciba varios enlaces de páginas “poco importantes”?

¡No todos los hiperenlaces son iguales!

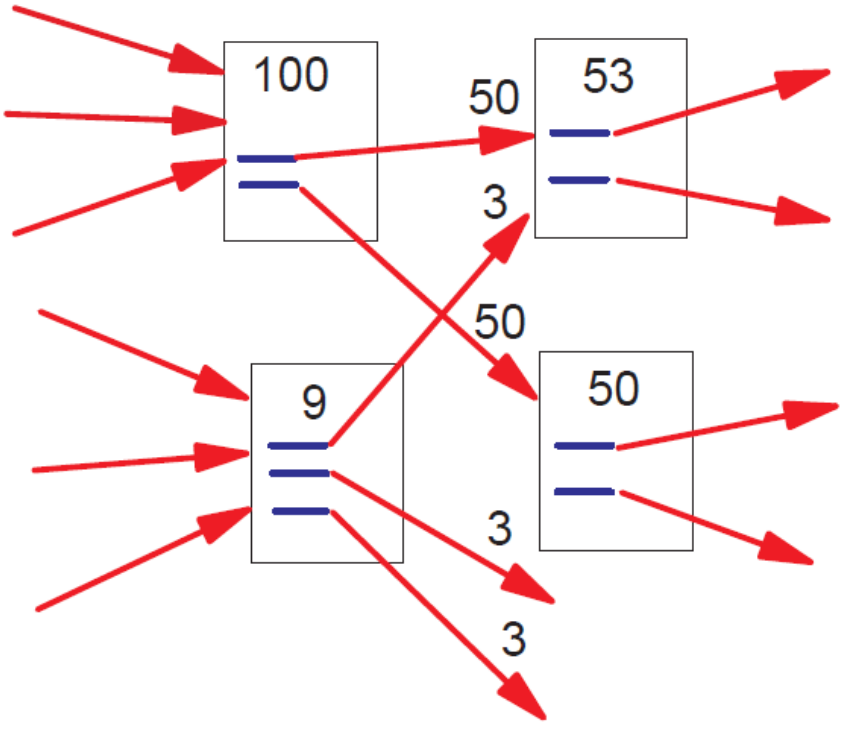
PAGERANK: Formulación básica recursiva (1)

Basada en la Centralidad de vector propio (prestigio de rango):

- El peso del voto de cada enlace es proporcional a la importancia (valor *PR*) de la página web que emite el voto
- Si una página *j* con importancia (prestigio) *PR(j)=x* tiene *n* hiperenlaces de salida, reparte su importancia entre ellos: cada enlace recibe *x*/*n* votos
- Así, el valor *PR(i)* de una página es una suma ponderada de los valores *PR(j)* de las páginas que apuntan a ella (la suma de los votos que recibe)

Con esta definición recursiva es mucho más difícil aumentar artificialmente la importancia de una página en un motor de búsqueda web (*search engine optimization, SEO*)

PAGERANK: Formulación básica recursiva (2)



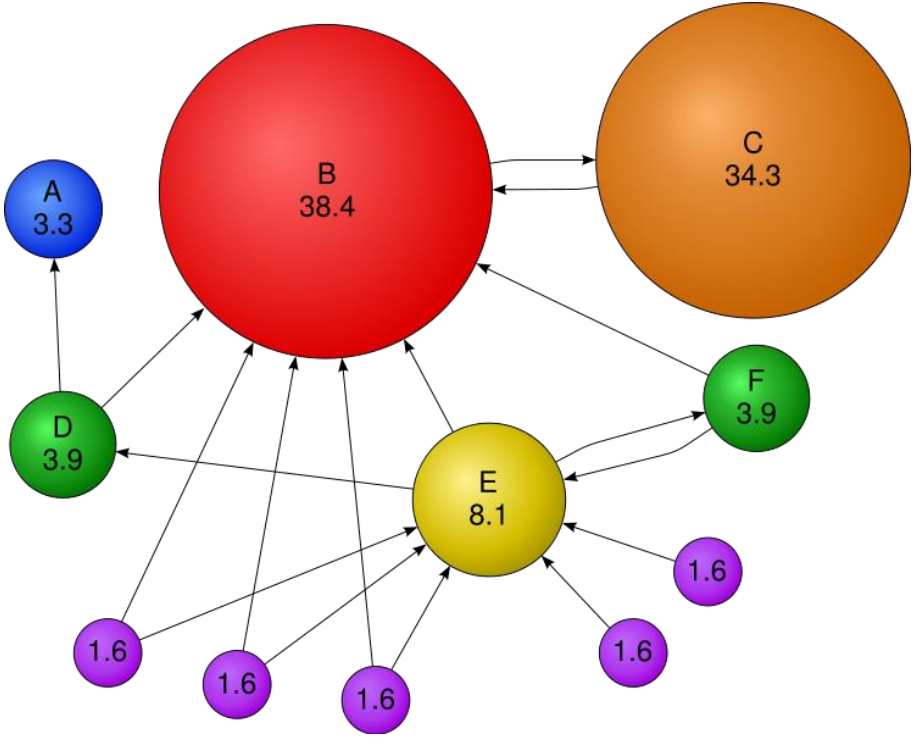


Figure 2: Simplified PageRank Calculation

PAGERANK: Formulación básica recursiva (3)

Para formular estas ideas se considera el grafo de la WWW $G=(V,E)$, donde V es el conjunto de páginas ($n=|V|$) y E el de hiperenlaces

El valor *PR* de una página i, $P(i)$, se calcula como:
donde O_j es el número de enlaces de salida de la página j

$$P(i) = \sum_{(j,i)\in E} \frac{P(j)}{O_j}$$

Matemáticamente, tenemos un sistema de n ecuaciones con n incógnitas, que se puede representar de forma matricial:

$\boldsymbol{P}=(P(1), \ldots, P(n))^T$

$$A_{ij} = \begin{cases} \frac{1}{O_i}, & \text{si } (i,j)\in E \\ 0, & \text{en otro caso} \end{cases}$$

$\boldsymbol{P} = \boldsymbol{A}^T \cdot \boldsymbol{P}$

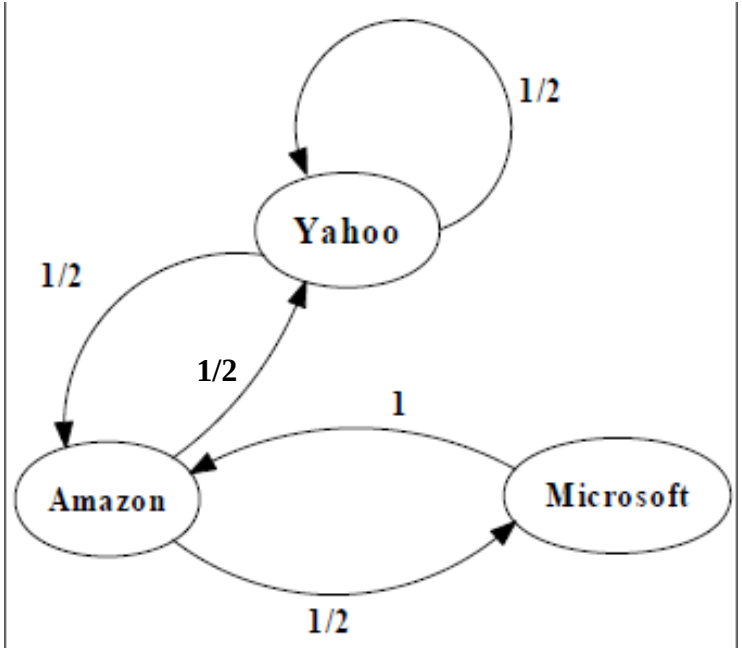
Esta ecuación coincide con la **ecuación característica para encontrar los vectores y valores propios de la matriz** $\boldsymbol{A}^T$. La solución $\boldsymbol{P}$ es un vector propio de $\boldsymbol{A}^T$, el vector propio principal, con valor propio 1

PAGERANK: Resolución con el Método de las Potencias

Método iterativo sencillo (funciona cuando la matriz es estocástica, es decir, si la suma de valores por filas vale 1):

1. $k \leftarrow 0$
2. Inicializar $\boldsymbol{P}^k = (1/n, \ldots, 1/n)^T$
3. Iterar $\boldsymbol{P}^{k+1} = \boldsymbol{A}^T \cdot \boldsymbol{P}^k$
4. Parar cuando $||\boldsymbol{P}^{k+1} - \boldsymbol{P}^k||_1 < \varepsilon$
   - $||x||_1 = \sum_i |x_i|$ es la distancia L_1
   - Se pueden usar otras como la Euclidea

PAGERANK: Ejemplo simple de cálculo del PageRank, primera iteración

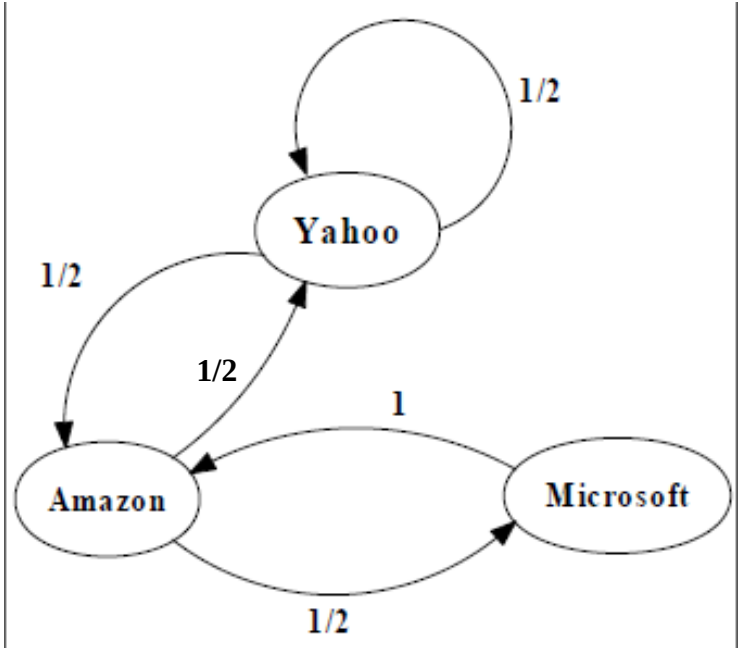


$$A^{T} = \begin{array}{c} \begin{matrix} Y & A & M \end{matrix} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \end{array} \begin{matrix} Y \\ A \\ M \end{matrix}$$

$$\begin{bmatrix} yahoo \\ Amazon \\ Microsoft \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = P$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = P$$

PAGERANK: Ejemplo simple de cálculo del PageRank, segunda iteración

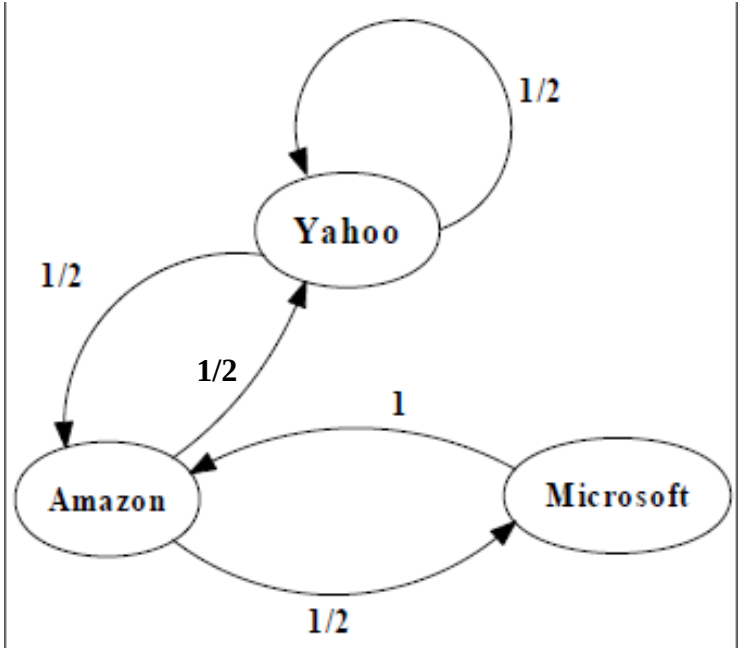


$$A^{T} = \begin{array}{c} \begin{matrix} \mathbf{Y} & \mathbf{A} & \mathbf{M} \end{matrix} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \end{array} \begin{matrix} \mathbf{Y} \\ \mathbf{A} \\ \mathbf{M} \end{matrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \boldsymbol{P}$$

$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \boldsymbol{P}$$

PAGERANK: Ejemplo simple de cálculo del PageRank, convergencia



$$A^{T} = \begin{array}{c} \begin{matrix} \mathbf{Y} & \mathbf{A} & \mathbf{M} \end{matrix} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \end{array} \begin{matrix} \mathbf{Y} \\ \mathbf{A} \\ \mathbf{M} \end{matrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \boldsymbol{P}$$

$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix} = \boldsymbol{P}$$

PAGERANK: Interpretación de caminos aleatorios en grafos. El “surfero” aleatorio

Patente de Google PageRank:

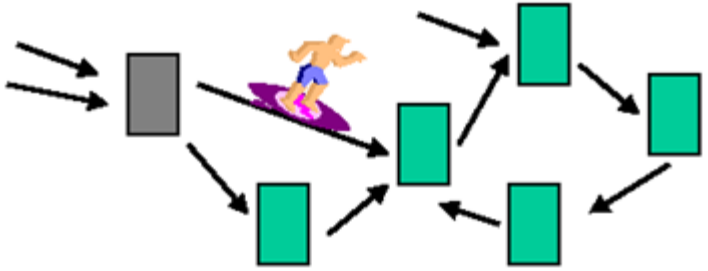
“The rank of a page can be interpreted as the probability that a surfer will be at the page after following a large number of forward links”

- Imaginemos un surfero que recorre el grafo de la web de forma aleatoria:
  - En un instante de tiempo t, el surfero está en alguna página web *i*
  - En el instante de tiempo t+1, el surfero escoge aleatoriamente un hiperenlace de salida de *i* y lo sigue, llegando a alguna página *j*
  - El proceso se repite indefinidamente
- Sea $\boldsymbol{p}(t)=(p_1(t), \ldots, p_n(t))$ un vector cuyo i-ésima componente indica la probabilidad de que el surfero esté en la página *i* en el tiempo *t*
  - $\boldsymbol{p}(t)$ representa una distribución de probabilidad sobre las páginas web

PAGERANK: Interpretación con caminos aleatorios en grafos: El “surfero” aleatorio (2)

El movimiento aleatorio del surfero en la web es un proceso estocástico modelado por una **cadena de Markov**:

- cada página web (**nodo**) *i* es un **estado** en el que puede estar el surfero,
- cada hiperenlace (**enlace**) es una **transición** que lleva de un estado *i* a otro *j* con una probabilidad (almacenada en la celda a_{ij} de la matriz de adyacencia $\boldsymbol{A}$),
- El surfero, localizado en una página web *i*, se mueve a otra página *j* aleatoriamente de acuerdo a las probabilidades existentes en $\boldsymbol{A}$ según la ecuación: $\boldsymbol{p}(t+1)=\boldsymbol{A}^{T}\cdot\boldsymbol{p}(t)$
- El vector $\boldsymbol{p}(t)=(p_1(t), \ldots, p_n(t))$ es una distribución de probabilidad estacionaria de la localización del surfero en una página *i* en el instante de tiempo *t* (del camino aleatorio realizado)
- Dicho vector satisface que $\boldsymbol{p}=\boldsymbol{A}^{T}\cdot\boldsymbol{p}$ (**vector propio**)
- **Una página tiene un prestigio de rango alto si su probabilidad de ser visitada es alta**



PAGERANK: Existencia y unicidad del vector propio *P*/*p*

Resultado central de los Procesos de Markov/Teoría de Caminos Aleatorios:

> En grafos que satisfagan unas ciertas condiciones, la distribución estacionaria ***p*** (es decir, el vector propio ***P***) es única y se alcanzará independientemente de la distribución inicial de probabilidad en el instante t=0

Para ello, se debe cumplir que la matriz de transición/adyacencia ***A*** sea **estocástica**, **irreducible** y **aperiódica**. Una matriz es estocástica si la suma de las transiciones por filas vale 1 (es una distribución de probabilidad): $\sum_{j=1}^{n} A_{ij} = 1$

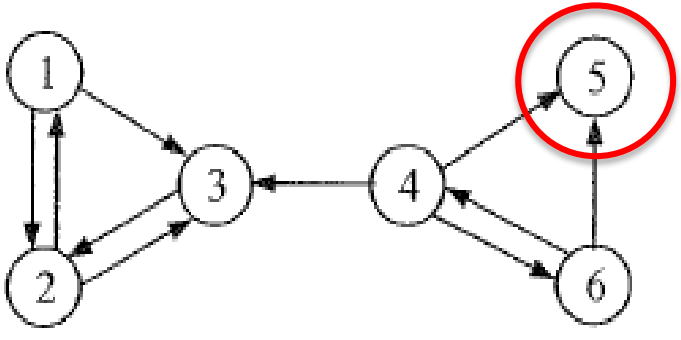
Por otro lado, es irreducible si el grafo es fuertemente conexo y aperiódica si no existe ningún estado que sea periódico, es decir, que tenga ciclos de vuelta para un estado en la cadena de Markov

Si se cumplen esas condiciones, ***p***/***P*** es el vector propio principal de $\boldsymbol{A}^{\mathsf{T}}$, tiene valor propio 1 y puede calcularse con el Método de las Potencias

PAGERANK: Problemática de la formulación básica recursiva (1)

La matriz de adyacencia del grafo de la WWW **no es una matriz estocástica** al presentar (muchos) enlaces que llevan a *páginas web descolgadas* ("*dangling links*")

Estas páginas son las que no tienen hiperenlaces de salida y, por tanto, presentan su fila de la matriz ***A*** con todos los valores a 0:



$$A = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

Afectan al modelo porque no está claro a quién se deben redistribuir sus votos. No afectan directamente al ranking de ninguna otra página

PAGERANK: Problemática de la formulación básica recursiva (2)

Existen dos soluciones distintas a este problema, ambas basadas en transformar la matriz de adyacencia ***A*** en una matriz estocástica:

- Eliminar los hiperenlaces y las páginas descolgadas antes de aplicar el PageRank y añadirlas de nuevo a posteriori

  Se puede calcular después su ranking de forma sencilla con la ecuación de los vectores propios. No serán los valores exactos pero serán muy similares

- Añadir un conjunto completo de hiperenlaces de salida en cada página web descolgada a todas las demás páginas de la WWW con probabilidad uniforme 1/*n*

En el ejemplo anterior:

$$\overline{A} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

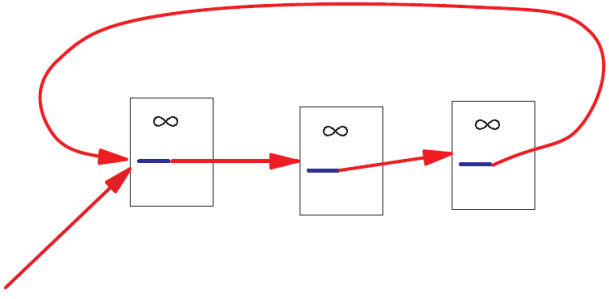
PAGERANK: Problemática de la formulación básica recursiva (3)

Por otro lado, la matriz de adyacencia del grafo de la WWW **tampoco cumple las propiedades de ser irreducible y aperiódica**

- Dicho grafo no es fuertemente conexo (en el ejemplo, no hay camino entre *3* y *4*), por lo que el surfero aleatorio podría no ser capaz de alcanzar algunas páginas

- Además, presenta ciclos que crean problemas para distribuir los votos, las llamadas **trampas de araña** (“*spider traps*”) o **callejones sin salida** (“dead ends”)

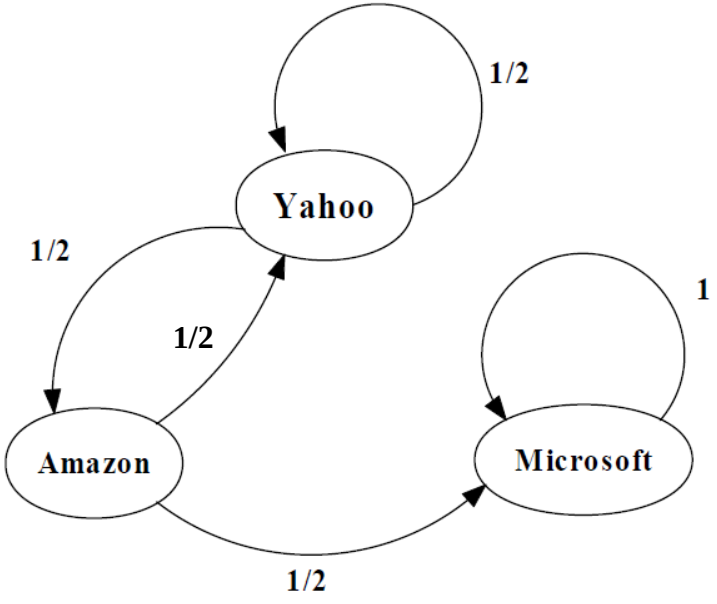
  Un grupo de páginas es una trampa de araña si no existen hiperenlaces que lleven alguna de sus páginas a otra de fuera del grupo



  Esto provoca que el surfero quede **atrapado** y termine moviéndose en círculos. Como consecuencia, los votos no se redistribuyen de forma adecuada

PAGERANK: Problemática de la formulación básica recursiva (4)

Ejemplo:



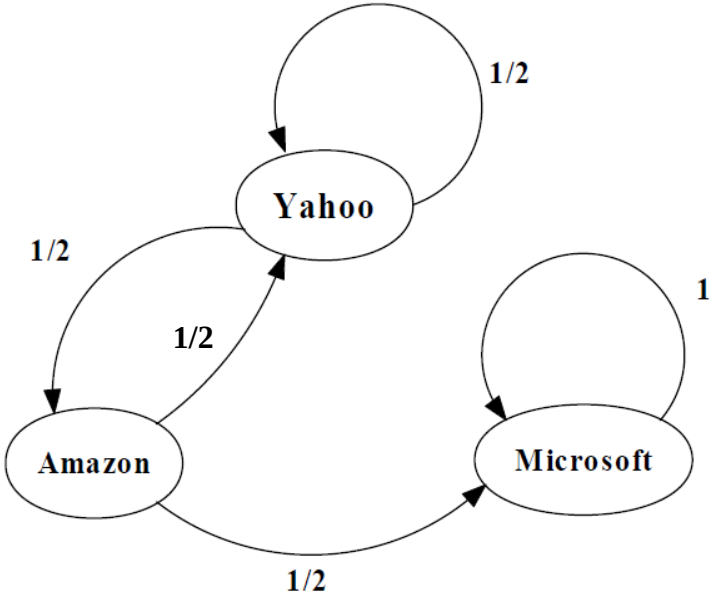
$$A^{T} = \begin{array}{c} \begin{matrix} \mathbf{Y} & \mathbf{A} & \mathbf{M} \end{matrix} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \end{array} \begin{matrix} \mathbf{Y} \\ \mathbf{A} \\ \mathbf{M} \end{matrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \boldsymbol{P}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \boldsymbol{P}$$

PAGERANK: Problemática de la formulación básica recursiva (5)

Ejemplo:



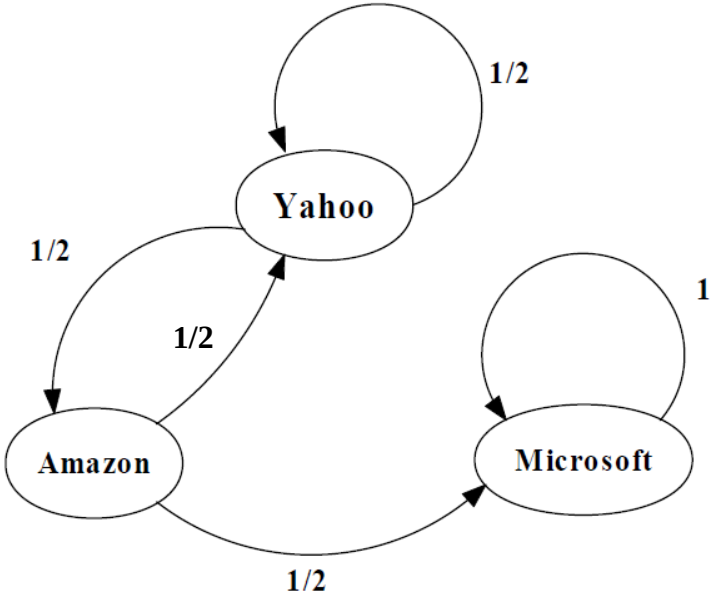
$$A^{\top} = \begin{array}{c} \begin{matrix} \mathbf{Y} & \mathbf{A} & \mathbf{M} \end{matrix} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \end{array} \begin{matrix} \mathbf{Y} \\ \mathbf{A} \\ \mathbf{M} \end{matrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = P$$

$$\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = P$$

PAGERANK: Problemática de la formulación básica recursiva (6)

Ejemplo:



$$A^{T} = \begin{array}{c} \begin{matrix} \mathbf{Y} & \mathbf{A} & \mathbf{M} \end{matrix} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \end{array} \begin{matrix} \mathbf{Y} \\ \mathbf{A} \\ \mathbf{M} \end{matrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \boldsymbol{P}$$

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \ldots \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{P}$$

PAGERANK: Problemática de la formulación básica recursiva (7)

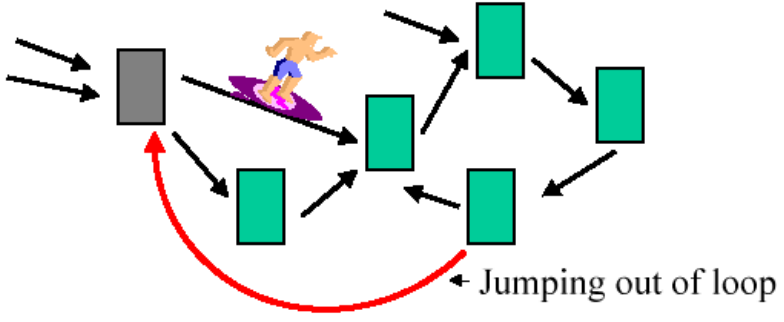
Los dos problemas se pueden solucionar de una forma simple con un **modelo modificado de PageRank**:

En cada instante de tiempo, el surfero tiene dos opciones:

- Con probabilidad *d*, escoger aleatoriamente una página a la que viajar de acuerdo a las probabilidades de transición de ***A***, **como siempre**
- Con probabilidad 1-*d*, saltar (**teletransportarse**) a cualquier otra página web escogida con probabilidad uniforme 1/*n*

El parámetro d se denomina **factor de desecho** ("*dumping factor*"). Sus valores típicos son entre 0.8 y 0.9 (p.ej. 0.85)

Gracias a esta modificación, el surfero puede escapar de la trampa de araña en unas pocas iteraciones (se aburre de dar vueltas ☺)



PAGERANK: Modelo Modificado de PageRank (1)

En representación matricial, el modelo modificado de PageRank equivale a trabajar con una nueva matriz de adyacencia ***A'*** en la que se cambian los valores iniciales de las probabilidades de transición de ***A***:

- Se añade un hiperenlace de teletransporte virtual de cada página *j* a cualquier otra página con probabilidad (1-*d*)/*n*
- Se reduce la probabilidad de los hiperenlaces reales de $1/O_i$ a d/O_i
- Esto equivale a quitarle a cada enlace una fracción 1-*d* de sus votos y repartirla equitativamente entre los enlaces no existentes

La nueva matriz obtenida, $\boldsymbol{A'} = (1-d)\cdot\boldsymbol{E}/n + d\cdot\boldsymbol{A}^T$, ($\boldsymbol{E}=\boldsymbol{e}\cdot\boldsymbol{e}^T$ es una matriz cuadrada con todas las posiciones a 1, ***A*** tiene que ser estocástica) cumple las tres propiedades necesarias y genera la ecuación asociada al modelo modificado de PageRank:

$$\boldsymbol{P} = \left((1-d)\frac{E}{n} + d\boldsymbol{A}^T \right)\cdot\boldsymbol{P}$$

PAGERANK: Modelo Modificado de PageRank (2)

Ejemplo (d=0.9) (**¡OJO! La matriz modificada está transpuesta**):

$$\overline{A} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \qquad (1-d)\frac{E}{n} + dA^T = \begin{pmatrix} 1/60 & 7/15 & 1/60 & 1/60 & 1/6 & 1/60 \\ 7/15 & 1/60 & 11/12 & 1/60 & 1/6 & 1/60 \\ 7/15 & 7/15 & 1/60 & 19/60 & 1/6 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 1/60 \end{pmatrix}$$

En la ecuación modificada, $\boldsymbol{e}^T \cdot \boldsymbol{P}$=1 ($\boldsymbol{P}$ es una distribución de probabilidad y $\sum_i p(i)$=1).
Si la escalamos multiplicando por n los dos lados de la desigualdad, nos queda:

$$\boldsymbol{P} = (1-d)\cdot\boldsymbol{e} + d\boldsymbol{A}^T \cdot \boldsymbol{P}$$

PAGERANK: Modelo Modificado de PageRank (3)

Esto deriva en la siguiente fórmula de PageRank para cada página web individual *i*:

$$P(i) = (1-d) + d \cdot \sum_{j=1}^{n} a_{ji} \cdot P(j)$$

la cual es equivalente a la propuesta en los artículos originales del PageRank:

$$P(i) = (1-d) + d \cdot \sum_{(j,i) \in E} \frac{P(j)}{O_j}$$

Con esta formulación, se puede obtener el vector propio principal ***P***, partiendo de cualquier inicialización, aplicando iterativamente el Método de las Potencias hasta que no se produzcan cambios significativos en los valores:

PageRank-Iterate(G)

$\boldsymbol{P}_0 \leftarrow \boldsymbol{e}/n$

$k \leftarrow 1$

repeat

$\boldsymbol{P}_k \leftarrow (1-d)\boldsymbol{e} + d\boldsymbol{A}^T \boldsymbol{P}_{k-1}$;

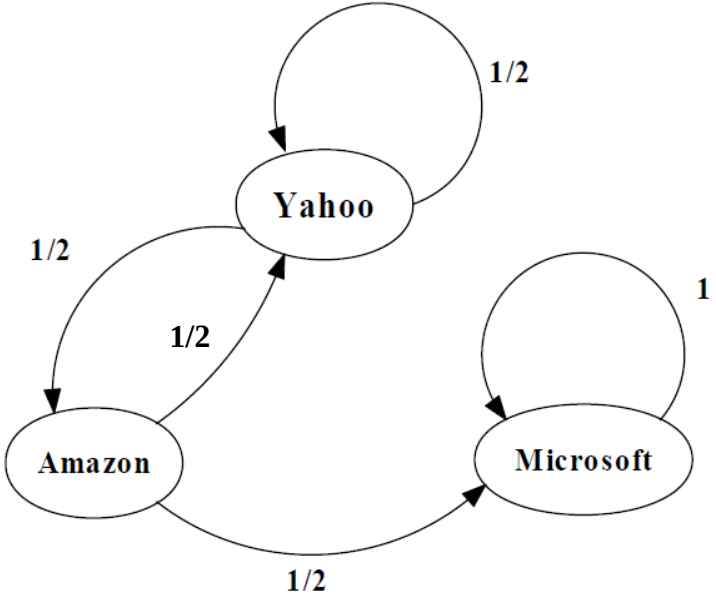
$k \leftarrow k+1$;

until $||\boldsymbol{P}_k - \boldsymbol{P}_{k-1}||_1 < \varepsilon$

return $\boldsymbol{P}_k$

PAGERANK: Modelo Modificado de PageRank (4)

Ejemplo:



$$A^{\top} = \begin{array}{c} \\ \mathbf{Y} \\ \mathbf{A} \\ \mathbf{M} \end{array} \begin{array}{c} \begin{array}{ccc} \mathbf{Y} & \mathbf{A} & \mathbf{M} \end{array} \\ \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \end{array}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} = \boldsymbol{P}$$

d=0.8

$$\begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.200 \\ 0.467 \end{bmatrix} \begin{bmatrix} 0.280 \\ 0.200 \\ 0.520 \end{bmatrix} \begin{bmatrix} 0.259 \\ 0.179 \\ 0.563 \end{bmatrix} \ldots \begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix} = \boldsymbol{P}$$

PAGERANK: Aspectos de Implementación y Convergencia (1)

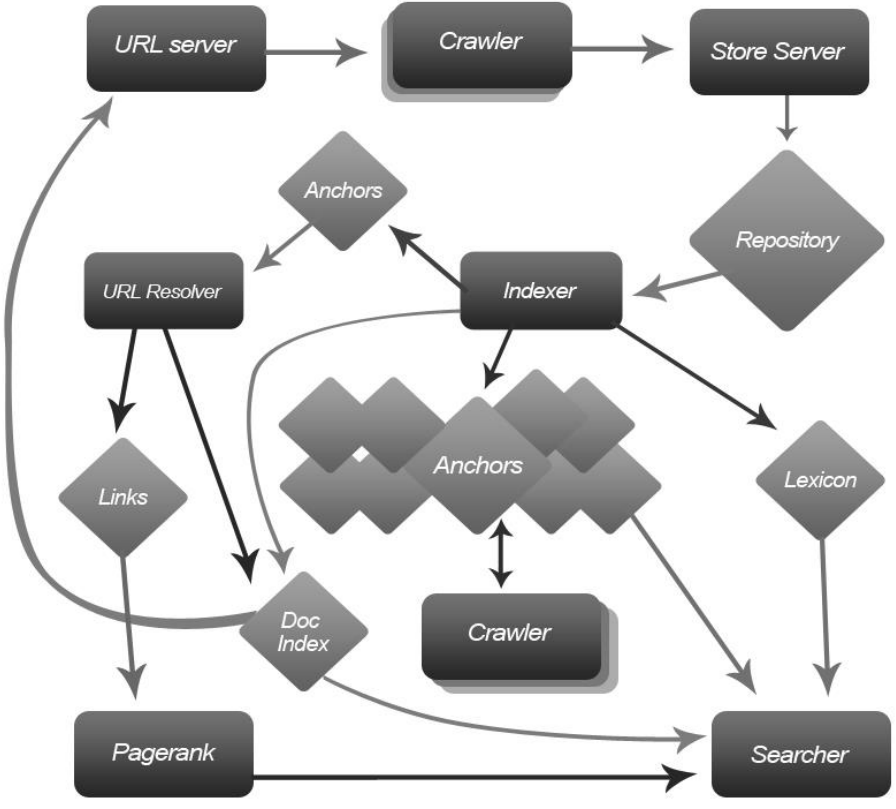
Procedimiento completo:

- La WWW se va indizando progresivamente mediante el uso de *crawlers*
  - Problemática: sitios muy grandes y caídos, errores en el HTML, cambios continuos en la web, …
- Se etiqueta cada URL con un valor entero único y se almacena cada hiperenlace en una base de datos que usa los IDs enteros para identificar a las páginas web
  - Esa base de datos se suele guardar en RAM
- Se ordena la estructura de enlaces por ID
- Se eliminan todos los enlaces descolgados de la base de datos
- Se inicializa el vector de rangos y se comienza la iteración del algoritmo PageRank
  - La elección de una buena asignación inicial puede acelerar la convergencia
- Se vuelven a incluir los enlaces descolgados y se itera el algoritmo de nuevo para calcular el ranking de sus páginas

PAGERANK: Aspectos de Implementación y Convergencia (2)

Aspectos prácticos:

- PageRank ordena la WWW completa (ranking global)
- Se lanza sólo cada varios meses (mínimo tres)
- No se ejecuta hasta que converja, se deja correr unas pocas iteraciones

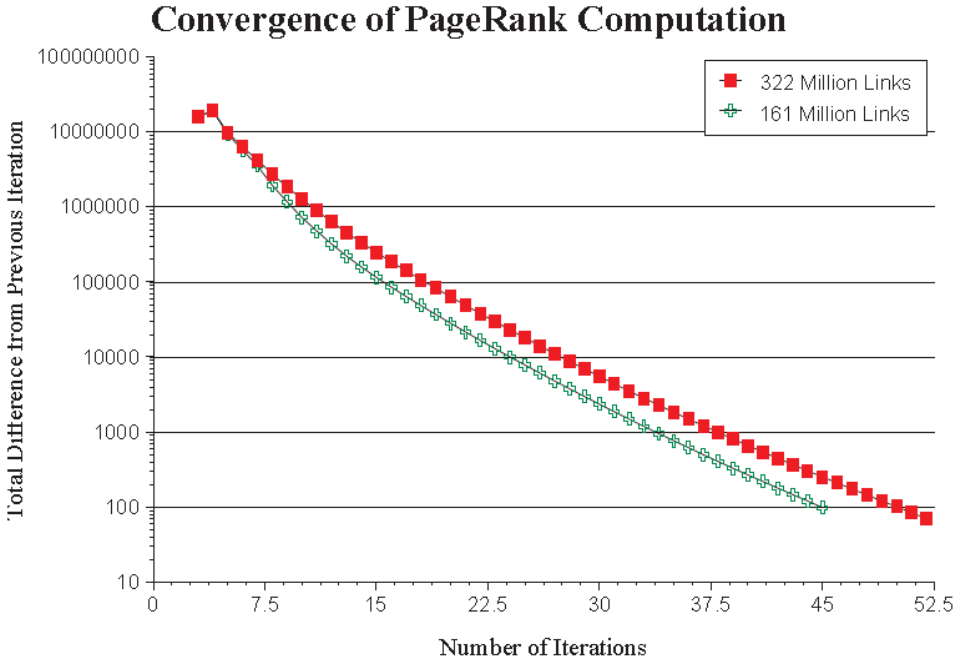


PAGERANK: Aspectos de Implementación y Convergencia (3)

Convergencia:

Como el fin último del algoritmo PageRank no es obtener el orden exacto de todas las páginas, no es necesario que el algoritmo converja totalmente sino que basta con que llegue a un nivel aceptable de tolerancia, necesitando menos iteraciones

- PR (322 millones de enlaces): 52 iteraciones
- PR (161 millones de enlaces): 45 iteraciones
- El factor de escala es aproximadamente lineal en log *n*, adecuado para grandes volúmenes



PAGERANK: Aspectos de Implementación y Convergencia (4)

Coste computacional:

- El paso clave es la multiplicación matriz-vector de la ecuación del PageRank:

$$\boldsymbol{P}^{k+1} = \boldsymbol{A}^{T} \cdot \boldsymbol{P}^{k}$$

- Es sencillo si se tiene suficiente memoria para almacenar la matriz y los dos vectores

- **Problema**: Pongamos n=1000 millones de páginas:
  - Asumamos que necesitamos 4 bytes para almacenar cada entrada
  - Necesitaríamos 2000 millones de entradas para los dos vectores $\boldsymbol{P}$, unos 8GBs (razonable)
  - Sin embargo, la matriz $\boldsymbol{A}$ tendría n^2 entradas $\rightarrow 10^{18}$ bytes (**¡mucho espacio!**)

PAGERANK: Aspectos de Implementación y Convergencia (5)

Reducción del espacio ocupado por la matriz *A*:

- Aunque ***A*** es una matriz densa, proviene de una matriz de adyacencia dispersa ***M***:
  - En media, unos 10 hiperenlaces por página, aproximadamente 10·*n* entradas

- Se puede reformular la ecuación del PageRank basándose en este hecho:

$$\boldsymbol{P} = d \cdot \boldsymbol{M} \cdot \boldsymbol{P} + [(1-d)/n]_n$$

  donde $[(1-d)/n]_n$ es un vector de dimensión *n* con todas las entradas a $(1-d)/n$

- Así, en cada iteración, se aplica el siguiente procedimiento:
  - Calcular $\boldsymbol{P}^{k+1} = d \cdot \boldsymbol{M} \cdot \boldsymbol{P}^{k}$
  - Añadir un valor constante $(1-d)/n$ a cada entrada de $\boldsymbol{P}^{k+1}$

PAGERANK: Aspectos de Implementación y Convergencia (6)

Reducción del espacio ocupado por la matriz *A*:

- La matriz dispersa ***M*** se almacena guardando solamente las entradas no nulas
  - Espacio aproximadamente proporcional al número de hiperenlaces
  - Pongamos $10n \rightarrow 4 \cdot 10 \cdot 1000$ millones = 40GB
  - Aunque no cabe en memoria, si es fácilmente almacenable en disco

ID de página	grado	páginas de destino
0	3	1, 5, 7
1	5	17, 64, 113, 117, 245
2	2	13, 23