Obligatorio 1

Camila Rojí 4.726.856-4 camiroji@gmail.com
Guillermo Leopold 5.485678-6 guilleleopold@gmail.com
Juan Pascual 4.569.366-6 juanpablopascual@hotmail.com
Ernesto Fernandez 4.567.713-3 cora.ernesto@gmail.com

Septiembre, 2015

Índice

1. Introducción

1.1. Historia de Google

Los orígenes de Google se centran en dos personas en particular: Sergey Brin y Larry Page, cofundadores de Google y actualmente presidente y CEO de la empresa, quienes se conocieron en la Universidad de Stanford.
En 1995 Larry y Sergey comienzan a trabajar en el "Digital Library Project"de la Universidad de Stanford. Comenzando a crear un algoritmo para la búsqueda de datos. Esa tecnología que Larry le da nombre de "PageRank"se convertiría mas tarde en el çorazón"que hará funcionar a Google.
En el año 1996 Larry y Sergey empiezan a colaborar en el desarrollo de un motor de búsqueda llamado BackRub. BackRub se utiliza en los servidores de Stanford durante más de un año, pero finalmente la Universidad deja de emplearlo porque requiere demasiado ancho de banda.
El dominio "Google.com" fue registrado el 15 de septiembre de 1997, nunca pensaron que se convertiría en algo "tan grande" como lo es hoy en día. El nombre "Google" proviene de un juego de palabras con el término"googol", término para referirse a un número representado por un 1 y seguido por 100 ceros.
En 1998 Larry y Sergey continuarón trabajando para perfeccionar la tecnología de búsqueda. A pesar de la fiebre "puntocom", no lograban encontrar inversionistas que financiaran Google, teniendo que conseguir dinero de sus familiares y amigos. Hasta que en el verano de ese mismo año Andy Bechtolsheim (cofundador de Sun Microsystems y vicepresidente de Cisco Systems) les firma un cheque por 100,000 dólares a nombre de "Google Inc". sin embargo "Google Inc."no existía, y para cobrar el cheque necesitaron buscar un local y fundar una compañía con ese nombre.
El 7 de diciembre de 1998, Google Inc., ya disponía de oficinas propias en Menlo Park, California.
Google.com tenía 10,000 visitas por día. En 1999 consiguieron 25 millones de dólares de dos importantes inversionistas Sequoia Capital y Kleiner Perking Caufield & Buyers.
Meses después las oficinas en Menlon Park, ya eran pequeñas para ellos, así que se transladaron a Googleplex, la actual sede central de Google en Mountain View, California, con más empleados y respondiendo a alrededor de 500,000 visitas al día. Google es hoy el mejor buscador de la red y el más utilizado.

2. Matriz de Google

2.1. Introducción:

Para Google es fundamental disponer de un sistema de clasificación de páginas que sea rápido y fiable para poder ordenar la enorme cantidad de páginas indexadas por el buscador, la cual continúa creciendo.
PageRank, el método inicial de cálculo que usaron los fundadores de Google para clasificar las páginas web según su importancia, tiene como finalidad la obtención de un vector, también llamado PageRank, que le otorga un valor relativo a cada página. Este método confía en el uso colectivo de la web empleando su gran estructura de enlaces como un indicador de trascendencia de una página en particular.
Para su formulación se emplea la matriz de Google, la cual es una matriz estocástica (matriz de probabilidad) particular.
Google interpreta un enlace de una página A a una página B como un voto de la primera hacia la segunda. Además, no solo mira la cantidad de votos, o enlaces que una web recibe, también analiza la página que emite el voto. Por eso, los votos emitidos por las páginas consideradas ïmportantes", es decir con un PageRank elevado, tienen un valor mayor, aumentando así el valor de otras páginas. Por lo tanto, el PageRank de una página refleja la importancia de la misma en Internet.

Para explicar el algoritmo se modela la red mediante un grafo orientado, donde los vértices son las distintas páginas y las aristas orientadas son los enlaces entre páginas. Cada página corresponde a una URL diferente. Por lo tanto, un sitio web puede contener muchas páginas pero este modelo no diferencia entre las páginas individuales (ejemplo: https://www.fing.edu.uy/bedelia) de un sitio web y su página principal (ejemplo: https://www.fing.edu.uy). Sin embargo, es más probable que el algoritmo otorgue más valor a la principal.

2.2. Construcción de la matriz:

Si tenemos un grafo como el siguiente:

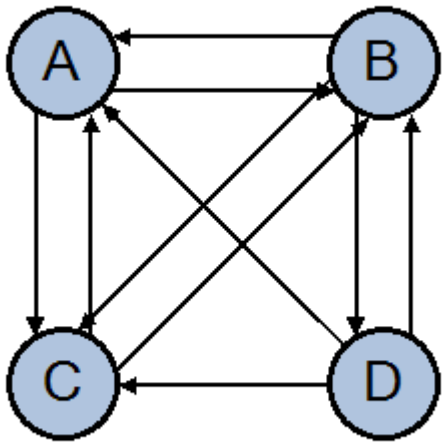


Figura 1: A, B, C y D páginas webs

Podemos representar el grafo mediante una matriz cuadrada $n * n$ tal que tenga un 1 si existe un enlace de una página a otra y un 0 en el caso contrario. El grafo anterior se vería modelado mediante la siguiente matriz:

$$G= \begin{array}{c} \begin{matrix} A & B & C & D \end{matrix} \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Queremos construir una matriz de transición P, cuyos elementos den las probabilidades condicionadas de cambio de página, o sea la probabilidad de ir a B estando en A. Para ello, vamos a asumir que, cuando se encuentra en una página, tiene la misma probabilidad de elegir cualquier enlace saliente. Esta elección es la base del modelo de Brin y Page.
Por ejemplo, en nuestra red, si comenzamos en A, entonces podemos elegir entre ir a B o a C con probabilidad $\frac{1}{2}$ para cada caso, mientras que si empezamos en B, entonces la probabilidad de ir a cualquier otra página es $\frac{1}{3}$.
Desde el punto de vista del cálculo es muy fácil obtener la matriz P a partir de la matriz G, basta dividir cada columna por la suma de los elementos de la

misma, siempre que esta cantidad no sea cero, es decir, siempre que esta columna no corresponda a una página sin salida.
Ahora ya podemos construir la matriz P asociada a la estructura del conjunto de páginas web de acuerdo con nuestra hipótesis de probabilidad de saltos.
Para automatizar el proceso, resumimos la información de la red en la siguiente matriz P, donde cada columna representa las posibles páginas de salida y cada fila es una página de destino.

$$P = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{3} & 0 & 0 \end{pmatrix}$$

Cada elemento toma un valor entre 0 y 1 correspondiente a una probabilidad. En el ejemplo, la matriz P dada por la ecuación es estocástica. Además, esta matriz tiene otras propiedades matemáticas interesantes: su espectro (conjunto de valores propios) es: $\lambda(P) = \{1, \frac{1}{2}, \frac{1}{3}, \frac{1}{6}\}$. Su radio espectral es $e(P) = 1$, y la matriz P es irreducible (grafo es fuertemente conexo) y primitiva (es irreducible, no negativa, y con $e(P)$ estrictamente mayor que cualquier otro valor propio). Estas propiedades son importantes ya que el vector PageRank es el vector limite de distribucion de probabilidad de una cadena de Markov; este existe y coincide con el de estado estacionario (un vector de probabilidad asociado al valor propio 1) y es independiente del vector de estado inicial. Esta propiedad se verifica, en particular, para matrices estocásticas y primitivas: $P \geq 0$, P irreducible y P solo tiene un valor propio ($(\lambda) = 1$) de módulo el radio espectral $e(P) = 1$. En este ejemplo, el vector de estado estacionario (que coincide con el de estado límite) resulta:
$vTest = [\frac{8}{28} \ \frac{9}{28} \ \frac{8}{28} \ \frac{3}{28}] \approx$ [0,29 0,32 0,29 0,10] , y es el vector PageRank de las cuatro páginas.
Esto está estrechamente relacionado con la navegación por internet, si un usuario cualquiera se mueve por el conjunto de las cuatro páginas enlazadas como en el grafo, entonces, para un tiempo suficientemente prolongado, lo más probable es encontrarlo en la página B (con una probabilidad de 0,32), mientras que la probabilidad de encontrarlo en las páginas A o C es de 0,29. Dicho de otra forma, el navegante emplea un 32 % de su tiempo visitando la página B. Un dato importante es que las páginas A, B y C son citadas por el mismo número de páginas, lo único que las diferencia es que la página B tiene un enlace a la página D que revierte posteriormente en un aumento de la probabilidad de la página B.

2.2.1. Modelo de Brin y Page

2.2.1.1 Modelo sin contemplar nodos sin salida

El modelo inicial para el cálculo del vector PageRank se basaba en calcular el vector estacionario de la matriz P de orden n, siempre que esta matriz fuera estocástica y primitiva. En este modelo no se contemplan los nodos sin salida con lo cual todas las columnas son distintas de cero y, en consecuencia, P es estocástica. Sin embargo, Brin y Page se dieron cuenta que la estructura de la web daba lugar a que P no fuera primitiva e introdujeron un nuevo modelo basado en una matriz estocástica P' que podemos escribir en la forma:

$P' = \alpha P + (1 - \alpha)ve^T$

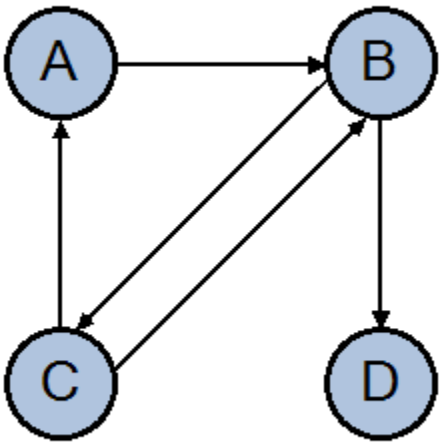
donde $(0 < \alpha < 1)$, $e^T \in \mathbb{R}^{1xn}$ es el vector de unos: $e^T = [11..,11]$, y $v \in \mathbb{R}^{1xn}$ es el llamado vector de personalización y es un vector de distribución de probabilidad que se suele tomar como $v = e(\frac{1}{n})$.

El producto ve^T es una matriz de orden n. El parámetro α se denomina de amortiguamiento (damping) y se suele tomar $\alpha = 0{,}85$, ya que fue el que usaron originalmente Brin y Page . El término $(1 - \alpha)ve^T$, con v un vector de distribución de probabilidad positivo, da lugar a que todos los elementos de P' sean no nulos, con lo cual, P' es irreducible. El efecto estadístico de este término es introducir saltos aleatorios que no dependen de las propiedades de enlace de la página. Valores de α próximos a uno ofrecen comportamientos más realistas pero pueden arruinar la irreducibilidad (en el límite $\alpha = 1$) y aumentar el número de iteraciones del método de la potencia. Por otra parte, es conocido, que una matriz irreducible y no negativa con algún elemento diagonal no nulo es primitiva.
En consecuencia, si no hay nodos sin salida, la matriz P' es estocástica y primitiva, que es lo que se desea. Sin embargo, en internet hay páginas sin enlaces salientes y se han de incorporar al modelo.

2.2.1.2 Modelo contemplando nodos sin salida

En el modelo contemplando nodos sin salida, cuando hay nodos que no tienen enlaces salientes en las columnas respectivas se tienen ceros. En consecuencia P no será estocástica ni tampoco lo será P'.

Un ejemplo es el siguiente grafo:



Con la matriz de adyacencias siguiente:

$$P = \begin{pmatrix} 0 & 0 & \frac{1}{2} & 0 \\ 1 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \end{pmatrix}$$

Se define entonces la nueva matriz:

$P' = \alpha[P + vd^T] + (1 - \alpha)ve^T$, con $\alpha = 0,85$ donde v y e son los mismos que en el modelo anterior y el vector $d \in \mathbb{R}^{nx1}$ se define como: $d_i = 1$, si $c_i = 0$, y $d_i = 0$ en otro caso. De esta forma, la matriz P, que generalmente se denomina matriz Google, es estocástica aunque haya nodos sin salida. La matriz vd^T actúa sobre las columnas pertenecientes a nodos sin salida, asignándoles una probabilidad de salto no nula. Por eso se dice que el modelo PageRank es un modelo de paseo y salto.
En síntesis la matriz de Google es una matriz estocástica particular que se utiliza por el algoritmo PageRank de Google. La matriz representa un gráfo dirigido con aristas que representan enlaces entre páginas. El valor de cada página se puede generar de forma iterativa de la matriz de Google utilizando el método de la potencia. Sin embargo, para que el método de la potencia converja, la matriz debe ser estocástica, irreducible y aperiódica.
Una de las características del PageRank es que si uno navega aleatoriamente por internet y está un tiempo suficientemente grande paseando, entonces tendrá una gran probabilidad de encontrar las páginas con mayor PageRank.
Por este motivo muchos administradores de páginas webs se centran en conseguir

enlaces entrantes de páginas bien rankeadas y eliminar los enlaces de páginas mal rankeadas a si mismo como conseguir que estos enlaces estén relacionados con el tema o clasificación de dicha página.

2.3. Cadena de markov

2.3.1. Proceso estocástico:

Una sucesión de observaciones $X_1, X_2, ..., X_n, ...$ se denomina proceso estocástico:

- Si los valores de estas observaciones no se pueden predecir exactamente
- Pero se pueden especificar las probabilidades para los distintos valores posibles en cualquier instante de tiempo.

Sea X_1: variable aleatoria que define el estado inicial del proceso
X_n: v.a. que define el estado del proceso en el instante de tiempo n

2.3.2. Cadena de markov

Una cadena de Markov es un proceso estocástico en el que si el estado actual X_n y los estados previos $X_1, ..., X_{n-1}$ son conocidos entonces la probabilidad del estado futuro X_{n+1} no depende de los estados anteriores $X_1, ..., X_{n-1}$, y solamente depende del estado actual X_n. Es decir:
$P(X_{n+1} = s_{n+1}|X_1 = s_1, X_2 = s_2, ..., X_n = s_n) = P(X_{n+1} = s_{n+1}|X_n = s_n)$

2.3.3. Clasificación de los estados

En una cadena homogénea con m estados $E_1, E_2, ..., E_m$ y matriz de transición $T = [p_{ij}]$, $(1 \leq i, j \leq m)$ el valor de p_{ij} es la probabilidad de que haya una transición entre E_i y E_j en un momento dado. Según lo anterior se pueden clasificar los estados de una cadena.

2.3.3.1 Estado Absorbente

Un estado es absorbente cuando una vez que se entra en él no se puede salir del mismo.

2.3.3.2 Estado Periódico

La probabilidad de que se regrese al estado E_i en el paso n es $p_{ii}^{(n)}$. Sea t un número entero mayor que 1. Supongamos que:
$p_{ii}^{(n)} = 0$ para $n <> t, 2t, 3t, ...$
$p_{ii}^{(n)} = 0$ para $n = t, 2t, 3t, ...$
En este caso se dice que el estado E_i es periódico de periodo t. Si para un estado no existe dicho valor de t entonces se dice que el estado es aperiódico

2.3.3.3 Estado Recurrente

Denominamos como $f_j^{(n)}$ la probabilidad de que la primera visita al estado E_j ocurra en la etapa n. Esta probabilidad no es la misma que $p_{jj}^{(n)}$, que es la probabilidad de que se produzca un retorno en el n-ésimo paso y esto incluye a los posibles retornos en los pasos 1, 2, 3,...,n-1 también.
De esta manera, la probabilidad de regresar en algún paso al estado E_j es:
$f_j = \sum_{n=1}^{\infty} f_j^{(n)}$
Por lo tanto, si $f_j = 1$, entonces seguro que se regresa a E_j y se denomina a E_j estado recurrente.
Los estados recurrentes se pueden diferenciar en:

- Recurrentes Positivos
- Recurrentes Nulos

Es posible demostrar que todos los estados recurrentes de una cadena de Markov finita son recurrentes positivos, por lo tanto, si tenemos una cadena de Markov que tiene una cantidad finita de estados e identificamos un estado recurrente, este será recurrente positivo. Si la cantidad de estados es infinito entonces un estado recurrente será recurrente nulo.

2.3.4. Cadena Irreductible

Cuando una C.M. finita y homogénea posee una sola clase, es decir si todos sus estados se comunican entre si, $i \leftrightarrow j \forall_{i,j \in E}$, se dice que la cadena es irreductible.

2.3.5. Vector distribución de estados

El vector de distribución de estados de la cadena en la etapa n, $Q(n)$, es un vector fila estocástico (estocástico = suma de sus elementos igual a 1) que contiene las probabilidades para la n-ésima etapa del estado de la cadena.

$Q(n) = [\pi_1(n), \pi_2(n), ..., \pi_r(n)]$, $|E| = r$, $\sum_{j=1}^{r} \pi_j(n) = 1$
donde $\pi_j(n) = P(X_n = j), j \in E$ es la probabilidad de que el estado de la cadena en la n-ésima etapa sea j, independientemente del estado inicial.

2.3.6. Distribución Estacionaria

Un vector estocástico π es una distribución estacionaria cuando $Q(n) = \pi$, entonces $Q(m) = \pi$, para todo $m > n$.

2.3.7. Teorema Ergódico

Sea X_n,$n > 0$ una cadena de Markov irreducible y con matriz de transicion P, supongamos ademas que P tiene una distribucion estacionaria π

$$\text{Sea } f : \varepsilon \to \mathbb{R} \text{ tal que: } \pi|f| = \sum_{i\in\varepsilon} \pi(i)|f(i)| < \infty$$

Entonces:

$$\lim_{x\to\infty} 1/n \sum_{0}^{n-1} f(X_j) = \pi f = \sum_{i\in\varepsilon} \pi(i)f(i), \text{ con probablidad 1.}$$

2.4. Teorema de Perron Forebenius

Sea A una matriz (cuadrada) con entradas no negativas, $A \geq 0$. Si A es irreductible, entonces
(a) existe un valor propio (simple) $\lambda > 0$ tal que $Av = \lambda v$, donde el vector propio es $v > 0$. Además, $\lambda \geq |\mu|$, para cualquier otro vector propio μ de A.
(b) Cualquier *vectorpropio* ≥ 0 es un múltiplo de v.
(c) Si hay k valores propios de módulo máximo, entonces son las soluciones de la ecuación $x^k - \lambda^k = 0$

2.4.1. Importancia en el problema de estudio

Debido a que la matriz de Google cumple con las hipótesis del teorema de Perron Frobenius, entonces se sabe que existe un único vector propio con entradas no negativas, y que además está asociado al valor propio positivo de módulo máximo.
Por lo tanto, este teorema no solo permite asegurar que existe el vector propio a buscar sino que también indica que está asociado al valor propio de módulo máximo, permitiendo así utilizar un algoritmo eficiente para calcularlo como es el Método de las Potencias, ya que este se utiliza particularmente para calcular el vector propio del mayor valor propio en matrices grandes.
Ademas se puede demostrar que:

$$min_j \sum_{i=1} a_{ij} \leq \lambda_1 \leq max_j \sum_{i=1} a_{ij}$$

Como todas las columnas de A suman 1, se desprende inmediatamente el hecho que $\lambda = 1$. Como dicho λ es un valor propio, se cumple que hay una única solución para el sistema $v_1 = Av_1$ (salvo otros vectores linealmente dependientes). Por otra parte, si se considera una solución para x_1 tal que todos los componentes del vector sumen 1, se obtendra el vector estacionario asociado a la cadena de Markov, el cual es tambien el vector PageRank de Google.

2.5. Vector propio dominante $v1$ y el PageRank

Como ya se comento en partes anteriores, el PageRank es el resultado obtenido como vector propio asociado al valor propio 1. Debido a que el radio espectral de la matriz de Google es 1 y considerando el teorema de Perron Frobenius, se puede concluir entonces que dicho valor propio será único y por lo tanto su vector propio asociado será dominante.
Por otra parte, el vector asociado al valor propio 1 es el vector de distribución estacionario ya que cumple la propiedad $\pi P = \pi$ que es equivalente a $Pv_1 = \lambda v_1$ para $\lambda = 1$.
Recordando la definición de la parte 3.5, el vector de distribución estacionario determina la probabilidad de encontrarse en cada sitio luego de transcurrido cierto tiempo (a largo plazo), debido a esta razón es que dicho vector es el utilizado para determinar el ranking de los distintos sitios web.

3. Métodos de Cálculo para $v1$

3.1. Método de las Potencias

El método de las potencias tiene como objetivo hallar el valor propio dominante y su vector propio asociado a una matriz, este es actualmente usado para obtener el vector dominante de la matriz de Google.
El mismo se plantea del siguiente modo:

- Partimos de un vector x^0 no nulo cualquiera.

- Se define $y^1 = Gx^0$ donde G es la matriz de Google. Para este paso, la implementación del mismo sufre algunas modificaciones con la finalidad de considerar los saltos a páginas sin hacer uso de links de salida, lo que se conoce como teletransportación, utilizandose la siguiente fórmula: $y^1 = p * P * y^0 + e * z^T * y^0)$. Donde p es el valor comentado en partes anteriores que suele valer 0.85, P es la matriz de probabilidades explicada en la primer parte, e es un vector de unos y el vector z se calcula tal que z_i es $\frac{1}{n}$ si la columna i de la matriz de Google suma y $\frac{1-p}{n}$ en otro caso.

- Luego, $c^1 = Maximo\{abs(y_i)\ de\ las\ componentes\ de\ y^1\}$.

- Y finalmente, $x^1 = \frac{y^1}{c^1}$.

- Se procede iterativamente repitiendo los pasos anteriores, hasta lograr predecir el valor hacia el que se aproxima la sucesion de valores c^i. El vector propio asociado al valor propio dominante sera el vector obtenido en y^n en n pasos

Un planteo mas genérico del problema es el siguiente:
Considerar un vector inicial $x^0 = x_1x_2...x_n$ con al menos un x_i no nulo y el sistema:

$$\begin{cases} y^{j+1} = Ax^j \\ c^{j+1} = componente\ dominante\ de\ y^{j+1} \\ x^{j+1} = \frac{1}{c^{j+1}}y^{j+1} \end{cases}$$

3.2. Método Lineal

Tambien es posible obtener el vector $v1$ de forma lineal, planteandose el siguiente sistema:
$(I - p * P) * v_1 = \beta * e$
donde I es una matriz identidad de igual dimension a P, β se toma con valor 1 y e es un vector de unos.
Despejando v_1 se obtiene el vector estacionario, lo que es lo mismo que el vector asociado al valor propio dominante $\lambda_1 = 1$

3.3. Implementación de los diferentes Métodos

Para la implementación de ambos métodos, se espera recibir como parametro la matriz P (matriz de probabilidades. Ademas se le pasa como parametro un valor p (valor de amortiguación) el cual típicamente tiene un valor de 0.85. El método de potencias recibe un tercer parametro correspondiente a un margen utilizado como condición de parada en las iteraciones del método.
Los archivos adjuntos son potencia.m y sistema.m respectivamente para las siguientes dos implementaciones.

3.3.1. Método de las Potencias

```
function [y_fin] = potencia(P,p,Margen)
        [m,n] = size(P);
    Diferencia = 1;
        It = 0;
        X = ones(m,1);
        e = ones(m,1);
        y = X;
        PAlt = p*P;
        sumas = sum(PAlt);
        for i=1:m
        if sumas(i)!=0
                z(i)=(1-p)/n;
        else
                z(i)=1/n;
                endif
        endfor
        while (Diferencia>Margen)
                y = PAlt*y+(z*y*e);
                C = max(y);
                y = 1/C * y;
                Diferencia = max(abs(y-PAlt*y-(z*y*e)));
                It = It + 1; %para posibles estadisticas quedo
        end;
        y_fin = y/sum(y) % se hace la escala para que sume 1
end
```

3.4. Método de Sistema Lineal

```
function [x] = sistema (P, p)
        [m,n]=size(P);
        e=ones(m,1);
        M=eye(m)-p*P;
        x=M\e;
        x=x/sum(x);
endfunction
```

3.5. Comparativa entre métodos

La comparación que realizamos entre los métodos fue sobre el tiempo de ejecución de los mismos utilizando matrices con distintas dimensiones. Con una misma matriz de 500×500 tomada de harvard500.mat, los resultados obtenidos fueron 8.21475 segundos para el método del sistema lineal y 11.6294 segundos para el método de las potencias.
Utilizando la matriz bcsstk10.mat de dimensiones 5300×5300, los resultados obtenidos fueron 8.20084 segundos para el método del sistema lineal y 9.46169 segundos para el método de las potencias.
Por útimo, utilizamos bcsstk32.mat con dimensiones $44,609 \times 44,609$ y los resultados obtenidos fueron, 12.1534 segundos para el método del sistema lineal y 7.68168 para el método de las potencias.
Por lo tanto se puede notar que el método de las potencias es más eficiente cuanto más grande es la matriz y el caso contrario para el método del sistema lineal, éste se vuelve menos eficiente cuanto más grande es la matriz.
En conclusión, cuando se utilizan matrices de dimensiones no tan elevadas (hasta un tamaño aproxiamdo de 6000x6000), es más eficiente utilizar el sistema lineal, y cuando las matrices tienen dimensiones mayores en más conveninte utilizar el método de las potencias.
Con esto, podemos entender que Google utilice el método de las potencias para resolver el algoritmo del PageRank ya que la matriz que utiliza es enorme.

4. El link Spamming y el segundo valor propio

Se conoce como Link Spamming a la acción de insertar vinculos entre distintos sitios webs con el único proposito de obtener una mejor valoración en el ranking PageRank. Este tipo de generación de Spam funciona aprovechando algoritmos de posicionamiento basados en enlaces (como PageRank), ya que estos como ya se comentó, funcionan asignando una posición más alta a páginas con mas enlaces a este. Dentro del Link Spamming, lo más destacado en su funcionamiento son los siguientes mecanismos y técnicas:

- Granja de Enlaces: Es la creación de comunidades de sitios webs mutuamente enlazados, los cuales se los conoce como *sociedades de admiracion mutua*. Estas tienen como objetivo aumentarse su valoración en el PageRank mutuamente y asi posicionarse bien. Luego otras páginas externas pagan a los dueños de estas granjas (hay empresas enteras que se dedican a realizar estas granjas) para asi mejorar su posicionamiento en la web.
- Enlaces ocultos: Refiere a la inserción de enlaces donde los usuarios que visitan la web no los puedan ver, con el fin de incrementar la popularidad de dichos enlaces.
- Sybil Attack Es la creación de multiples personalidades falsas con objetivos maliciososo. Así, un spammer puede crear muchos sitios webs y linkearlos entre si, un ejemplo de esto es la creación de falsos blogs, hecho conocido como *spam blogs*.

Debido a esto y como se busca explotar por parte de quien utiliza link spamming el funcionamiento en esencia de algoritmos como PageRank es que se hace necesario la existencia de tener métodos para poder detectar y filtrar estos casos, lograndose así tener una busqueda y una ordenación mas fiable, la cual convierte a Google en el motor de busqueda dominante en el mercado que es hoy.

Es así que mientras en el PageRank el vector propio dominante de la matriz de Google se usa para calcular el puntaje de las webs (su PageRank), el segundo valor propio se utiliza para detectar este problema de Spam mediante distintos métodos. Para Google es esencial la presencia de un método efectivo para la detección del Link Spamming ya que sin este, los resultados no serian lo fiable que son hoy por hoy.

Antes de entrar en detalle acerca de los métodos de detección utilizados, vale la pena detallar algunas definiciones y enunciados usados para el desarrollo de los mismos.

Definición Un conjunto de estados es cerrado para una cadena Markov correspondiente a P^T si y solo si $i \in S$ y $j \notin S$ implica $p_{ji} = 0$, esto quiere decir que una cadena

de Markov es cerrada si no es posible salir del subconjunto cerrado S una vez que se esta en este.
Tambien es importante recordar la definición de cadena irreductible que se puede encontrar en la seccion 2.3.4, agregando ademas que si el subconjunto es cerrado, la presencia de un conjunto de nodos con esta caracteristica es un sintoma de la presencia de Spam.

Corolario En presencia de dos subconjuntos irreductibles y cerrados, se cumple que el segundo valor propio de la matriz correspondiente al conjunto padre (matriz de Google para el caso) es p (el mismo que ya se ha mencionado a lo largo de todo el informe como factor de amortiguación utilizado en la matriz de Google).

Teorema Considerando $v_2 = (x1....xn)^T$ vector propio asociado a $\lambda_2 = p$. Se cumple que para toodos los x_i tal que $x_i = 0$, estos no pertenencen a ningún conjunto irreductible cerrado. Reduciendose asi los posibles candidatos a spam.

Método de deteccion de Link Spamming Asumiendo la existencia de por lo menos dos subconjuntos irreducibles cerrados de forma que $\lambda_2 = p$.
Se comentara en detalle unicamente uno de los dos métodos utilizados para detectar link spamming usando el segundo valor propio, el cual se considera sustancialmente mas rápido para problemas grandes en comparación al otro método.
Los pasos para su aplicación son los siguientes:

- 1- Computar un vector propio dominante de P^T resolviendo $(I - P^T)y = 0$
- 2- Determinar todos los coeficientes distintos de 0. (Son los unicos que pueden pertenecer a un conjunto irreductible y cerrado).
- 3- Aplicar el algoritmo de Tarjan (el algoritmo se puede estudiar en el anexo) al subgrafo conformado por los nodos obtenidos en el paso anterior. Los CFC (componentes fuertemente conexas) en este gafo son subconjuntos irreducibles y cerrados.
- 4- Formar las matrices de transición $P_{r+j,r+j}$ (r es el número de CFC no cerrados, j es un indice de 1 a k siendo k con $k = \#$Componentes Cerrados) correspondiente a los distintos subconjuntos irreducibles y cerrados o CFC.
- 5- Computar los vectores dominantes $y_{r+j,r+j}$ de las matrices $P_{r+j,r+j}$ y luego se los normaliza para que sea estocástico y se agregan la cantidad de 0s necesarios para convertirlo en vectores propios de la matriz P^T. Se le llama $\vec{y}_{r+j,r+j}$ a los vectores resultantes.
- 6- Finalmente, combinar los vectores $\vec{y}_{r+j,r+j}$ de a pares con la finalidad de obtener los vectores propios v_2 independientes de la matriz de Google, mediante la fórmula $v_2 = \vec{y}_{r+j,r+j} - \vec{y}_{r+j+1,r+j+1}$

Para detectar link spamming, unicamente se deben computar los subconjuntos irreducibles y cerrados en los pasos 1-3.

Este unicamente es un algoritmo de detección de link spamming mediante el uso del segundo vector propio, el cual es eficiente para dicha técnica de crear subconjuntos cerrados irreducibles, pero no es la única utilizada para link spamming y diferentes técnicas requiriran diferentes formas de evitarlo.

5. Conclusiones

- Como reflexión general, podemos ver como el PageRank afecta de una manera destacada en nuestro día a día, ya que en la actualidad, se realizan grandes cantidades de busquedas y los sitios a los cuales llegamos se ven totalmente influenciados por su ranking PageRank.
- Considerando la dimensión que tiene la web en la actualidad, un sitio mal posicionado en el PageRank (junto con el hecho de que Google es ampliamente el motor de busqueda mas usado) se vuelve practicamente inalcanzable.
- Del ultimo punto se desprende que para cualquier Webmaster resulta vital estar al tanto del funcionamiento de estos algoritmos (ya sea PageRank u otro algoritmo) para saber que "trucosütilizar para posicionarse mejor en la web.
- En base a las comparaciones y estudios realizados en la parte 2, se logra comprender porque Google utiliza el método de las potencias para calcular el vector dominante, ya que el mismo resulta mas eficiente para matrices de enormes dimensiones como suelen ser las matrices de Google.
- Google dedica un gran esfuerzo a la mejora de los resultados obtenidos, tal es asi que resulta mas complejo la implementación de los algoritmos para combatir el link spamming que el propio algoritmo de puntuación PageRank y en gran medida es gracias a este esfuerzo que su motor de busqueda es catalogado como el mejor y es el mas usado a nivel mundial.

6. Anexo

6.1. Otros algoritmos de puntuación web

PageRank es el algoritmo de puntuación de Google, el principal motor de busqueda en la web. Sobre este se conoce en detalle su funcionamiento y reglas ya que hasta hace un tiempo Google tenía una politica de no ocultamiento acerca del funcionamiento del mismo (ultimamente con el fin de evitar engaños y manipulación ha ido reduciendo el conocimiento público del algoritmo y sus actualizaciones). Debido a esto es que los algoritmos de busqueda de algunos motores no son tan conocidos ni se cuenta con tanta información al respecto, pese a esto a continuación se detallan brevemente algunos de los mas conocidos con información que se ha ido desprendiendo sobre los mismos a lo largo de los años.

6.1.1. WebRank

WebRank es como se le conoce al algoritmo de posicionamiento de Yahoo!, el mismo se mide con una valoracion entre 1 y 10.
La medida de este, a diferencia del PageRank, no se ve influenciada por la cantidad de enlaces hacia dicha página sino que es afectada por la información que se obtiene a traves de la barra de busqueda instalada en los navegadores Web.
Desde la misma, se envia información de las URLs visitadas a los servidores de Yahoo!, y en estos se recopila y realizan los cálculos de WebRank.
Esta información tambien servirá al robot Slurp, a rastrear los sitios web visitados por los usuarios para asi indexarlos a los resultados de busqueda en caso de que ya no hayan sido agregados anteriormente.

6.1.2. Algoritmo de Bing

El algoritmo de Bing, del cual recien a fines del ultimo año se revelo parte de su funcionamiento, se basa fuertemente en la calidad del contenido; la misma esta compuesta segun Microsoft por tres pilares: Autoridad, Utilidad y Presentación.
Autoridad: ¿Cómo saben si pueden confiar en el contenido? No solo importa los enlaces que apuntan a la página, sino tambien de como son el sitio web y la página que apuntan.
Utilidad: Se determina si el contenido es util y lo suficientemente detallado. Se analiza inclusive si la página contiene videos, imagenes, graficos, etc.
Presentación: Se busca contenido fácil de encontrar y fácil de leer. Páginas con muchos anuncios se pueden ver perjudicadas en su puntuación por ejemplo.

Finalmente, la fórmula matemática del ranking se podria resumir en la siguiente expresión:

$Ranking = f(RelevanciadelTopico, Contexto, CalidaddelContenido)$
Donde calidad del contenido refiere a los tres puntos mencionados anteriormente, el contexto analiza tanto la ubicación física del usuario como la posible relacion de la busqueda con algún tópico reciente y la relevancia del topico analiza si los resultados estan relacionados con la busqueda.

6.1.3. Algoritmo de HITS: El predecesor del PageRank

Jon Kleinberg es el matemático estadounidense nacido a principios de los setenta creador en 1999 del algoritmo HITS (Hypertext Induced Topic Selection). Este guarismo de Kleinberg tiene en cuenta la importancia de una página web a través de analizar sus enlaces entrantes. En realidad, estudiando el comportamiento de Google a la hora de indexar sus resultados de búsqueda, no es descabellado pensar que este algoritmo se presenta como uno de las grandes factores influyentes para el buscador.

El algoritmo HITS se basa en dos tipos de sitios, los cuales ayudan a identificar cuál es la importancia del resto de las páginas existentes en la Red. Estos son los que dan título a este post: Hubs y Authorities. Veamos en qué consiste cada uno de ellos:

Hubs: Estos son todos aquellos sitios que, recibiendo una buena cantidad de links, enlazan, a su vez, a numerosísimas páginas web que consideran importantes. De esta manera los sites 'hubs' determinan la importancia de otros sites. El exponente más evidente de sitio hub es el directorio DMOZ.

Authorities: Son los web referente en temas concretos. Es decir, aquellos que tienen muchos enlaces entrantes pero que, por su parte, apuntan a muy pocos sites (muy pocos enlaces salientes) La diferencia que existe entre este algoritmo HITS y el PageRank está en la manera de considerar cada portal o site. Si para Google, su PageRank otorga a cada página web una puntuación atendiendo al valor y cantidad de los enlaces entrantes que posee. Para el HITS existen dos valores para cada página web, el hub para determinar su calidad como recurso de enlaces y el authority los valora como recurso de información.

No es de extrañar, que el propio Google, a pesar de basarse en un PageRank que probablemente haya modificado sus fórmulas actuales en relación a las primeras que tuvo, se esté acercando o esté combinando sus criterios teniendo en cuenta cada vez más a la clasificación determinada por Kleinberg, el autor del algoritmo HITS

6.2. Algoritmo de Tarjan

Es un algoritmo de la teoría de grafos para encontrar las componentes fuertemente conexas de un gráfo . A pesar de que lo precede cronológicamente, puede ser visto como una versión mejorada del algoritmo de Kosaraju , y es comparable en eficiencia con el algoritmo de Gabow.

El algoritmo toma un grafo dirigido como entrada, y produce una partición de los vértices del gráfico en las componentes fuertemente conexas del gráfico. La idea básica del algoritmo es la siguiente: una búsqueda en profundidad comienza a partir de un nodo inicial cualquiera. Se lleva a cabo una busqueda en profundidad en los nodos que aún no han sido encontrados. La búsqueda no explora cualquier nodo que ya ha sido explorado. Las componentes fuertemente conexas forma los subárboles del árbol de búsqueda. Los nodos se colocan en una pila en el orden en que se visitan.posteriormente los nodos se toman de la pila y se determina si cada nodo es la raíz de una componente fuertemente conexa.

7. Bibliografía

- Alex Sangers and Martin B. van Gijzen. "The eigenvectors corresponding to the second eigenvalue of the Google matrix and their relation to link spamming". Journal of Computational and Applied Mathematics, vol. 277, pp. 192-201, 2015. Consultado en Agosto-Septiembre 2015.
- http://blog.kleinproject.org/?p=1605&lang=es Como funciona Google: Cadenas de Markov y Valores Propios. Consultado el 23 de Agosto de 2015..
- https://es.wikipedia.org/wiki/PageRank
- http://personales.upv.es/ pedroche/inv/_docs/fpedrochev4(sema).pdf
- https://en.wikipedia.org/wiki/Google_matrix
- http://www.math.cornell.edu/ mec/Winter2009/RalucaRemus/Lecture3/lecture3.html
- http://www.ugr.es/ mibanez/ejemplos/potencias.pdf Metodo de Potencias. Consultado el 26 de Agosto de 2015.
- https://www.fing.edu.uy/inco/cursos/io/
- http://www.ugr.es/ bioestad/_private/cpfund10.pdf
- http://www.cimat.mx/ jortega/MaterialDidactico/modestoI11/CMarkov2.pdf
- https://www.google.com/about/company/history/?hl=es
- http://www.cad.com.mx/historia_de_google.htm
- http://www.sema.org.es/ojs/index.php?journal=sema&page=article&op=viewFile&path%5B%5D= Consultado el 8 de Setiembre de 2015
- https://es.wikipedia.org/wiki/Spamdexing Spamdexing. Consultado el 13 de Septiembre de 2015.
- http://www.spamflag.com/link-identification-guide/ The Definitive Guide to Manipulate Links. Consultado el 16 de Septiembre de 2015.
- http://blogs.bing.com/search-quality-insights/2014/12/08/the-role-of-content-quality-in-bing-ranking/ The Role of Content Quality in Bing Ranking. Consultado el 20 de Septiembre.
- http://www.sitiosargentina.com.ar/notas/Marzo_2004/95.htm Como funciona el WebRank de Yahoo. Consultado el 20 de Septiembre.
- http://eapd2g6.wikispaces.com/Algoritmo+de+Tarjan Algoritmo de Tarjan. Consultado el 23 de Septiembre de 2015.

- http://www.beevoz.com/2014/11/05/algoritmo-hits-hubs-y-authorities-alternativa-yo-dependecia-del-pagerank/ Algoritmo de Hits. Consultado el 24 de Septiembre de 2015.
- http://www.cise.ufl.eduresearchsparsematrices/list_by_id.html. Consultado el 24 de Setiembre de 2015.