Algoritmo *Page Rank* de *Google*
Proyecto Final
Cálculo Numérico I

Mario Becerra Contreras
Clave única: 124362

Fecha de entrega: 29 de mayo de 2014

1. Introducción

En teoría clásica de redes se quiere encontrar la forma de caracterizar la topología de redes particulares. El modelo de red es un objeto matemático que tiene su origen en Teoría de Gráficas, sin embargo con la explosión y atención que han recibido las redes sociales hoy en día, la teoría para estudiar dichos cuerpos matemáticos ha aumentado considerablemente. Es posible modelar situaciones y fenómenos de distinta naturaleza como una red. Es decir, como un conjunto de nodos ν y un conjunto de aristas ε, *vertices* y *edges* respectivamente, denotado por $\mathcal{G} = (\nu, \varepsilon)$. Donde una arista o arco, existe si y solo si dos nodos potencialmente distintos están conectados. Es decir, $e_{ij} \in \varepsilon$ si v_i se comunica con v_j.
En este contexto se quiere estudiar, entre otras cosas, qué tan importante es un nodo en la red. De acuerdo a la interacción que existe entre éste y los otros nodos. En el contexto de Internet, en particular a Google le interesa saber qué tan importante es una página de internet. Esto puede ser desde cuantas páginas apuntan a una en particular y cuantas ligas existen en dicha página hacia el exterior. Es por esto que estudiaremos y daremos una breve introducción a lo que se refiere al algoritmo ideado por los técnicos en Google para calificar la importancia de un nodo en una red.
El algoritmo de *Page Rank* está basado en caminatas aleatorias alrededor de la red. Pensando específicamente en un proceso de Markov, definido más adelante. La premisa en este algoritmo es que un vínculo entre dos sitios es un voto de confianza de la página fuente a la página destino. Es por esto que solo puede ser aplicado a redes dirigidas. Siguiendo la idea de los votos la importancia del voto de un individuo depende de los votos que ha recibido dicho sujeto.
Al momento de navegar en una red consideremos que uno puede pasar a uno de los vecinos con probabilidad α o sin ningún compromiso podemos navegar hacia otro nodo en la red con probabilidad $1 - \alpha$. Notemos que este último movimiento puede ser realizado y se dirigirá a un nodo en una sección completamente diferente en la red. A esto también se le conoce como *teletransportación*.

2. Desarollo

Comenzaremos dando algunas definiciones y teoremas. Conocemos la *matriz de adyacencias* en una red dirigida como la matriz $A \epsilon \mathbb{R}^{n \times n}$ tal que $A_{ij} = 1$ si existe un arco con origen el nodo i y destino el nodo j denotado por k_i^{out} como el número de arcos que salen del nodo i. De tal forma que podemos expresar dicha relación en forma vectorial como

$$k_i^{out} = A\underline{1} \tag{2.1}$$

El resultado de esta ecuación proviene de que los renglones de A tienen 1 o 0 dependiendo si existe un arco entre el nodo i (fila i) y el nodo j (columna$j$). Entonces, si se suman las columnas de A se obtiene un vector

k^{out}, y eso es lo que $A\underline{1}$ hace.

Se prosigue a dar la definición de proceso estocástico. Un proceso estocástico es un conjunto de variables aleatorias $X_\bullet = \{X_t, t \in T\}$ con conjunto de índices T. Si el conjunto T es numerable, entonces se dice que el proceso estocástico es a tiempo discreto.
En particular, un proceso de Markov es un proceso estocástico que satisface la propiedad de Markov. Un proceso de Markov a tiempo discreto es llamado una *cadena de Markov*. La propiedad de Markov, en el caso discreto, se puede definir como sigue:

$$\mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_0 = x_0) = \mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1}).$$

Se define solamente la cadena de Markov pues es el tipo de proceso estocástico que se usa en este trabajo.
Una matriz estocástica por filas es tal que todas sus entradas están en el intervalo $[0, 1]$ y la suma de los elementos de cada fila es 1.
Definamos ahora la siguiente matriz estocástica B como

$$B = \alpha D^{-1} A + \frac{1}{n}[(1-\alpha)I + \alpha diag(a)]\underline{1}\underline{1}^T$$

Donde $D = diag(k^{out})$. Si se llega a dar el caso de que para algún nodo, digamos el nodo i, $k_i^{out} = 0$ entonces definimos la entrada de la matriz D_{ii} como 1. Por otro lado también definimos $a \, \epsilon \, \mathbb{R}^{n \times n}$ como la matriz cuya entrada $a_{ij} = 1$ si $k_i^{out} = 0 \; \forall \; j$.

Se sabe que α es la probabilidad de pasar de un nodo a alguno de sus vecinos; αD^{-1} pondera la probabilidad de pasar de un nodo a sus adyacentes (α) entre las salidas de un nodo, es decir, si $k_i^{out} = m$ entonces con probabilidad α/m se puede pasar a cualquiera de los nodos adyacentes al nodo i.

La matriz B refleja la probabilidad de pasar del nodo i al nodo j, sin importar si estos dos nodos están o no conectados. Esto implica que B^T define la caminata aleatoria con teletransportación en la red. Es por esto que se piensa en el siguiente proceso markoviano

$$x(n+1) = B^T x(n)$$

Cuyo estado estacionario está definido por el eigenvector izquierdo de la matriz B, i.e., el eigenvector derecho de B^T:

$$x = B^T x$$

Esto está garantizado por el teorema de Perron-Frobenius, el cual se analiza más adelante.
Definición: Una matriz A es irreducible si no puede ser acomodada de forma triangular superior por bloques por una matriz de permutación P, es decir, $PAP^{-1} \neq \begin{pmatrix} C & D \\ 0 & E \end{pmatrix}$
con C y E de dimensión mayor a cero. Asimismo, se dice que una matriz es irreducible si su digrafo asociado es fuertemente conectado, esto es, si existe un camino que vaya en ambas direcciones para cada par de vértices del digrafo. Esta definición puede ser aplicada a una matriz estocástica, en particular a una cadena de Markov, en el sentido de que una cadena de Markov es irreducible si su matriz de transición es irreducible, significando que de cualquier estado se puede pasar a cualquier otro estado; donde la matriz de transición P es tal que $P_{ij} = \mathbb{P}(X_{n+1} = i | X_n = j)$.
Teorema de Perron-Frobenius para matrices irreducibles: Sea A una matriz irreducible cuadrada de $n \times n$ no negativa, i.e., tal que $(a_{ij}) > 0$ para todo $i, j \leq n$, entonces:

1. alguno de sus eigenvalores es positivo y mayor o igual que los demás eigenvalores en valor absoluto;
2. existe un eigenvector positivo asociado a ese eigenvalor;

3. no existe otro eigenvector de A cuyas entradas sean todas positivas.

Teorema de Perron-Frobenius para matrices positivas: Sea A una matriz cuadrada de $n \times n$ tal que todas sus entradas son positivas, entonces:

1. existe λ_{pf} eigenvalor de A tal que λ_M es no negativo y sus eigenvectores izquierdo y derecho asociados son no negativos.

2. para todo λ eigenvalor de A tal que $\lambda \neq \lambda_M$ se tiene que $|\lambda| \leq \lambda_M$.

3. el eigenvalor λ_M es de multiplicidad 1.

Esta última versión, en particular, se puede aplicar al problema con el que se está trabajando pues la matriz B es positiva. Volviendo a las matrices de transición, se dice que π es un vector (fila) de probabilidad invariante si $\pi P = \pi$, con P una matriz de transición. Esto es lo mismo que decir que $P^T \pi^T = \pi^T$, o sea, que π^T sea eigenvector de P^T. El teorema de Perron-Frobenius se puede aplicar a esto y concluir que toda matriz estocástica tiene dicho vector. Esto además se puede mezclar con el método de la potencia estudiado en el trabajo anterior, este vector está asociado al eigenvalor más grande, por lo que puede ser calculado con el método de la potencia. El vector invariante puede ser interpretado como aquel vector al cual no le afecta la aplicación de la matriz de transición, o sea que no varía con el tiempo, de ahí su nombre de *invariante*. En particular, si la matriz B es positiva y estocástica, como en este caso, el máximo eigenvalor de la matriz B es igual a 1. Por lo que el eigenvector asociado a λ_M, el máximo eigenvalor, representa el estado estacionario del proceso.

3. Experimentos numéricos

Se consideran tres redes para probar el algoritmo:

1. Celegans: la red de neuronas del gusano más estudiado en la literatura de redes.

2. Rutas de Europa: una red que representa las rutas entre ciudades de Europa.

3. Red de energía de Europa: la red de energía de toda Europa.

Para probar el algoritmo se definen los elementos de la matriz B, para así calcular B y encontrar su eigenvalor más grande con su correspondiente eigenvector, para de esta forma, tomar las 10 entradas más grandes y considerar los nodos correspondientes como los más importantes. Se realizó el experimento considerando dos valores de α, $\alpha_1 = 0.85$ y $\alpha_2 = 0.5$, para de esta forma ver si cambian mucho los nodos más importantes.

4. Resultados numéricos

Para cada uno de los experimentos numéricos se verificó que el eigenvalor máximo de la matriz B^T fuera igual o cercano a 1. Después se encontraron los eigenvectores correspondientes y, ordenándolos, se llegó a los siguientes resultados:

Tabla 1: Top 10 de nodos importantes. Celegans.

$\alpha_1 = 0.85$	$\alpha_1 = 0.5$
48	48
56	56
106	106
97	97
182	95
95	107
107	39
164	182
169	67
175	59

Tabla 2: Top 10 de nodos importantes. Rutas de Europa.

$\alpha_1 = 0.85$	$\alpha_1 = 0.5$
257	257
115	212
212	115
27	27
89	142
2	89
182	700
700	182
142	92
159	11

Tabla 3: Top 10 de nodos importantes. Red de energía de Europa.

$\alpha_1 = 0.85$	$\alpha_1 = 0.5$
1833	1833
750	1815
1815	1806
1628	750
1806	1421
1421	1628
762	1634
1537	1537
1700	1834
1834	1700

En las Tablas 1, 2 y 3 se pueden ver los nodos más importantes de cada red, siendo el primero el más importante y el último el menos.

5. Conclusiones

Al cambiar los valores de la probabilidad α, los nodos más importantes, es decir, los *top 1* de cada uno no cambian, por lo que si un nodo es muy importante, siempre va a ser de los más importantes sin importar la probabilidad.

6. Apéndice

Script 1 utilizado para las carreteras:

```
alpha1=.85;
alpha2=.5;

load('Eroads.mat')

[~,n]=size(A);

B1=MatrizB(A,alpha1);
B2=MatrizB(A,alpha2);

max1=max(eig(B1'));
max2=max(eig(B2'));

%x1=linsolve(eye(n)-B1',zeros(n,1));
[V1,D1]=eig(B1');
v1=V1(:,1); %Es el eigenvector asociado al eigenvalor más grande. Es la primera columna de la matriz V1

[V2,D2]=eig(B2');
v2=V2(:,1); %Es el eigenvector asociado al eigenvalor más grande. Es la primera columna de la matriz V2

[t1,I1]=sort(-abs(v1));
[t2,I2]=sort(-abs(v2));

Top10_1=I1([1:10]')
Top10_2=I2([1:10]')

csvwrite('CarreterasTop10_alpha1.csv', Top10_1)
csvwrite('CarreterasTop10_alpha2.csv', Top10_2)
```

Script 2 utilizado para Celegans:

```
alpha1=.85;
alpha2=.5;

load('celegans_connectivity.mat')

[~,n]=size(A);

B1=MatrizB(A,alpha1);
B2=MatrizB(A,alpha2);

max1=max(eig(B1'));
max2=max(eig(B2'));

%x1=linsolve(eye(n)-B1',zeros(n,1));
[V1,D1]=eig(B1');
v1=V1(:,1); %Es el eigenvector asociado al eigenvalor más grande. Es la primera columna de la matriz V1
```

```
[V2,D2]=eig(B2');
v2=V2(:,1); %Es el eigenvector asociado al eigenvalor más grande. Es la primera columna de la matriz V2

[t1,I1]=sort(-abs(v1));
[t2,I2]=sort(-abs(v2));

Top10_1=I1([1:10]')
Top10_2=I2([1:10]')

csvwrite('CelegansTop10_alpha1.csv', Top10_1)
csvwrite('CelegansTop10_alpha2.csv', Top10_2)
```

Script 3 utilizado para la red de Energía

```
alpha1=.85;
alpha2=.5;


load('Power_Europe.mat');

[~,n]=size(A);

B1=MatrizB(A,alpha1);
B2=MatrizB(A,alpha2);

max1=max(eig(B1'));
max2=max(eig(B2'));

%x1=linsolve(eye(n)-B1',zeros(n,1));
[V1,D1]=eig(B1');
v1=V1(:,1); %Es el eigenvector asociado al eigenvalor más grande. Es la primera columna de la matriz V1

[V2,D2]=eig(B2');
v2=V2(:,1); %Es el eigenvector asociado al eigenvalor más grande. Es la primera columna de la matriz V2

[t1,I1]=sort(-abs(v1));
[t2,I2]=sort(-abs(v2));

Top10_1=I1([1:10]')
Top10_2=I2([1:10]')

csvwrite('PowerTop10_alpha1.csv', Top10_1)
csvwrite('PowerTop10_alpha2.csv', Top10_2)
```

Función que devuelve la matriz B:

```
function [B] = MatrizB(A, alpha)
%Calcula la matriz estocástica B a partir de la matriz de adyacencia A y la
%probablidad alpha

[~,n]=size(A);
```

```
k=A*(ones(n,1));

 D=diag(k);
 [~,n]=size(D);
 for i=1:n
     if k(i)==0
         D(i,i)=1
     end
 end

 a=zeros(n);
 for i=1:n
     for j=1:n
         if k(i)==0
             a(i,j)=1
         end
     end
 end



 B1=alpha*inv(D)*A;
 B2=(1/n)*(1-alpha)*eye(n)*ones(n,1)*ones(n,1)';
 B3=(1/n)*alpha*diag(diag(a))*ones(n,1)*ones(n,1)';
 B=B1+B2+B3;


end
```