Práctica 4
Algoritmo de PageRank

27 de enero de 2014

Índice

1. Introducción **1**

2. Modelizando la WWW **1**

3. Matrices estocásticas (de nuevo) **4**

4. Ejemplo trabajado con Scilab **6**

1. Introducción

Los algoritmos de búsqueda en la web, como el algoritmo PageRank de Google, constituyen excelentes aplicaciones del álgebra matricial.

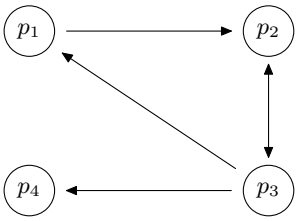
PageRank es el método de cálculo utilizado por los fundadores de Google (Sergey Brin y Lawrence Page) para clasificar las páginas web de acuerdo con su grado de importancia. El objetivo del método es obtener un vector, el vector PageRank, que proporciona la importancia de las páginas (es decir, asigna, a cada página web, un grado "de importancia", ordenando todas las páginas de acuerdo con dicho grado). Este vector se calcula a partir de la estructura de las conexiones (enlaces) entre las páginas web. Vamos a describir aquí sólo los aspectos básicos del método de PageRank, sin entrar en las modificaciones y mejoras que se han realizado (y se siguen realizando actualmente).

2. Modelizando la WWW

Supongamos que queremos buscar el mejor libro de Álgebra Lineal. Probablemente tratarías de realizar una búsqueda en Google. La lista de páginas web que se obtienen como resultado de

la búsqueda aparecen ordenadas según su "grado de importancia". Este "grado de importancia" se define de manera que una página es importante si otras páginas importantes enlazan a ella. De manera algo más precisa: "una página puede tener mayor PageRank si hay muchas páginas que enlazan a ella, o bien si existe un número suficiente de páginas importantes que enlazan a ella". Pero, ¿cómo podemos decir si una página es importante a partir de la propia importancia de las páginas? Esta es la cuestión que trataremos de resolver.

Modelizaremos la World Wide Web (que es una colección de páginas web conectadas mediante enlaces) por medio de un grafo dirigido cuyos vértices se corresponden con las páginas y cuyos arcos representan los enlaces entre ellas. Para una mejor comprensión del proceso que vamos a explicar a continuación, en lugar de trabajar con la totalidad de la WWW consideraremos, como modelo simplificado, una pequeña red de páginas cuyos enlaces están representados por medio del siguiente grafo dirigido:



(La flecha doble debe ser interpretada como dos arcos, uno en cada sentido).

La idea clave es que las páginas deberían ser más importantes si, o bien son enlazadas a muy a menudo desde otras páginas, o bien son enlazadas desde una cantidad suficiente de páginas importantes. Es decir, la importancia de una cierta página p_i (denotada por $I(p_i)$) debe aumentar cada vez que es enlazada desde otra páginas p_j; además el incremento debe ser directamente proporcional al "grado de importancia" de la página p_j que la enlaza e inversamente proporcional al número de enlaces a_j presentes en la página p_j (la importancia total de la página p_j se reparte entre todos sus enlaces). Teniendo esto en cuenta, tiene sentido la siguiente fórmula:

$$I(p_i) = \sum_{\text{páginas } p_j \text{ que enlazan a } P_i} \frac{I(p_j)}{a_j}$$

En nuestro ejemplo:

- La página p_1 tiene sólo un enlace (a p_2) y, por tanto, $a_1 = 1$.
- La página p_2 tiene también sólo un enlace y, por tanto, $a_2 = 1$.
- La página p_3 tiene 3 enlaces. Luego $a_3 = 3$.
- La página p_4 no tiene ningún enlace y, por tanto, $a_4 = 0$.

Así pues, tenemos que:

$I(p_1) = \frac{1}{3}I(p_3)$
$I(p_2) = I(p_1) + \frac{1}{3}I(p_3)$
$I(p_3) = I(p_2)$
$I(p_4) = \frac{1}{3}I(p_3)$

Observa que esto es equivalente a la siguiente igualdad matricial:

$$\underbrace{\begin{bmatrix} I(p_1) \\ I(p_2) \\ I(p_3) \\ I(p_4) \end{bmatrix}}_{\vec{I}} = \underbrace{\begin{bmatrix} 0 & 0 & 1/3 & 0 \\ 1 & 0 & 1/3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \end{bmatrix}}_{G} \underbrace{\begin{bmatrix} I(p_1) \\ I(p_2) \\ I(p_3) \\ I(p_4) \end{bmatrix}}_{\vec{I}} \tag{1}$$

Así pues, el "vector de importancias" $\vec{I}$ (que es lo que queremos calcular) debe de satisfacer la igualdad $G\vec{I} = \vec{I}$, donde $G = [g_{ij}]$ es la matriz 4×4 tal que

$$g_{ij} = \frac{1}{\text{número de enlaces de } p_j}$$

si p_j enlaza a p_i, y $g_{ij} = 0$ si p_j no enlaza a p_i. El video que viene en el siguiente enlace (en inglés) te ayudará a entender todo esto:

http://www.youtube.com/watch?v=ZstQKxUW7oM

La igualdad (1) significa que queremos un *vector estacionario* $\vec{I}$ de la matriz G (es decir, un vector que permanece invariante si lo multiplicamos por G). Queremos, además, que $\vec{I}$ sea un *vector de probabilidad*, que significa que todas las componentes de $\vec{I}$ sean no negativas y sumen 1. Un vector $\vec{I}$ satisfaciendo todas estas propiedades *ordenaría* las páginas asignando, a cada una de ellas, un número entre 0 y 1 que "significa" su "grado de importancia". Pero, además, debería haber un **único** vector con estas características (dado que la existencia de dos vectores diferentes significaría la existencia de dos ordenaciones distintas de las páginas, y no es esto lo que queremos). Resumiendo, lo que queremos es que exista un vector $\vec{I}$ satisfaciendo las siguientes propiedades:

(1) $\vec{I}$ es un *vector estacionario* para G, es decir, $G\vec{I} = \vec{I}$,

(2) $\vec{I}$ es un *vector de probabilidad*, es decir, sus componentes son no negativas y suman 1.

(3) $\vec{I}$ es el **único** satisfaciendo (1) y (2).

¿Satisface la matriz anterior G estos requisitos? En este caso la respuesta es NO (puede comprobarse fácilmente que el único vector estacionario es $\vec{0}$). Sin embargo, como veremos, podemos modificar ligeramente (y de manera inteligente) la matriz G para forzarla a satisfacer las 3 condiciones anteriores.

3. Matrices estocásticas (de nuevo)

Recordemos que una matriz $n \times n$ es *estocástica* si todas su entradas son no negativas y la suma de las entradas en cada columna es 1. Las matrices estocásticas son especialmente interesantes debido a la siguiente propiedad (que ya vimos en la Práctica 3):

Teorema 1. Cualquier matriz estocástica tiene, al menos, un vector de probabilidad estacionario.

Se sigue de la definición de G que la suma de las entradas en cada una de sus columnas no nulas es 1. Sin embargo, la última columna es nula y, por tanto, G no es estocástica; el problema es la *existencia de columnas nulas*, y esto ocurre porque hay una página (p_4) que no tiene ningún enlace (se corresponde con un vértice "sumidero" del grafo dirigido). La manera más simple de evitar este problema es imaginar que, cuando alguien está "navegando a través de la red" y llega a una página como esta (sumidero), lo que hace para continuar es elegir una nueva página "totalmente al azar". Así pues, podemos pensar que una página "sumidero" tiene, realmente, un enlace a cada una de las demás páginas (y a sí misma). Por tanto, podemos redefinir la matriz G sustituyendo las columnas nulas por vectores $(1/n, 1/n, \ldots, 1/n)$, donde n es el número total de páginas (4, en nuestro caso):

$$G = \begin{bmatrix} 0 & 0 & 1/3 & 1/4 \\ 1 & 0 & 1/3 & 1/4 \\ 0 & 1 & 0 & 1/4 \\ 0 & 0 & 1/3 & 1/4 \end{bmatrix} \tag{2}$$

Debido al teorema anterior, esta matriz (y cualquier matriz definida de esta manera a partir de cualquier red de páginas) satisface las condiciones (1) y (2). En este ejemplo somos afortunados y, como tú mismo/a puedes comprobar, la condición (3) se satisface también; sin embargo, en general, esto no es verdad. Así pues, necesitamos modificar otra vez la definición de G para asegurarnos de que el vector de probabilidad estacionario es siempre único. El siguiente resultado (que ya vimos en la Práctica 3 con más generalidad y que es una consecuencia del Teorema de Perron-Fröbenius) nos proporciona la propiedad que necesitamos[1]:

Teorema 2. Si G es una matriz estocástica que tiene todas sus entradas estrictamente positivas entonces G tiene un **único** vector de probabilidad estacionario.

Nuestra matriz G (y en la práctica, cualquier matriz G obtenida de esta manera) tiene entradas nulas y, por tanto, no se le puede aplicar este teorema. Sin embargo hay una manera coherente de modificar G para conseguir que todas sus entradas sean estrictamente positivas:

Imaginemos a una persona que está "navegando" por la red siguiendo los enlaces de las páginas que se encuentra: cada vez que él/ella visita una página p_i, va a otra página usando

[1] El enunciado y la prueba del Teorema de Perron-Fröbenius usan, como ingredientes clave, los conceptos de valor propio y vector propio, que veremos en el Tema 6.

uno de los enlaces de p_i; si p_i es una "página sumidero" entonces él/ella va a una página aleatoria (escribiendo una dirección aleatoria en la barra de direcciones). Quizás, de vez en cuando (e independientemente de si la página visitada en ese momento es un "sumidero" o no lo es) él/ella podría querer escribir, en la barra de direcciones, una dirección aleatoria en lugar de "seguir" los enlaces de la página web actual. Es decir, podría "querer seguir", en lugar de la matriz G, la siguiente matriz (que vamos a denominar **matriz de aleatoriedad**:

$$E := \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix};$$

esto significa que, cuando "se sigue" esta matriz, todas las páginas web tienen la misma probabilidad de ser elegidas.

Ahora asumamos lo siguiente: cada vez que un "navegante" visita una página tiene dos posibilidades para actuar: "seguir" la matriz G (es decir, continuar usando uno de los enlaces de la página web en la que se encuentra) o bien "seguir" la **matriz de aleatoriedad** E (es decir, elegir una página web al azar).

Fijemos un número real $\alpha \in]0,1[$ que denote la probabilidad de navegar "siguiendo" la matriz G. Entonces, la probabilidad de navegar "siguiendo" la matriz E es $1-\alpha$. α mide, en cierto modo, el "grado de libertad" que tenemos para navegar "siguiendo" cada una de las dos matrices. (Observa que, como estamos interesados en dar mucha más importancia a la matriz G que a la matriz E, α debería ser un valor cercano a 1). Esta nueva situación equivale a considerar, en lugar de la matriz G, esta otra:

$$\mathbf{G} = \alpha G + (1-\alpha)E.$$

Es fácil comprobar que esta matriz es estocástica y que, además, todas sus entradas son estrictamente positivas. Entonces podemos aplicarle los teoremas que hemos visto antes, deduciendo que existe un vector $\vec{I}$ satisfaciendo las condiciones anteriores (1), (2) y (3). Éste es el vector PageRank que estábamos buscando.

Una manera de calcular este vector $\vec{I}$ es resolver el siguiente sistema de ecuaciones lineales[2]:

$$(\mathbf{G} - I_{4\times 4})\vec{x} = \vec{0}. \tag{3}$$

El papel del parámetro α es muy importante. Observa que si $\alpha = 1$ entonces $\mathbf{G} = G$. Esto significa que estamos trabajando con la estructura de enlaces original de la web. Sin embargo, si $\alpha = 0$ entonces $\mathbf{G} = E$; en otras palabras, estamos considerando que cualquier página tiene un enlace a cualquier otra perdiendo, así, la estructura original de la red. Claramente nos gustaría que α fuera un valor cercano a 1 para dar mucha importancia a la estructura original.

[2]Observa que $\mathbf{G}\vec{x} = \vec{x} \;\Leftrightarrow\; \mathbf{G}\vec{x} = I_{4\times 4}\vec{x} \;\Leftrightarrow\; \mathbf{G}\vec{x} - I_{4\times 4}\vec{x} = \vec{0} \;\Leftrightarrow\; (\mathbf{G} - I_{4\times 4})\vec{x} = \vec{0}$

Sin embargo, existe otra consideración. En la práctica, la matriz $\mathbf{G}$ correspondiente con la WWW es una matriz extremadamente grande. Como consecuencia de esto, resolver un sistema como (3) usando los métodos habituales no es una buena idea[3]. En lugar de esto, en la práctica, el vector $\vec{I}$ se calcula usando un método iterativo conocido como *método de la potencia*. *Esencialmente* consiste en calcular, por iteración, una buena aproximación al límite de la siguiente cadena de Markov [4]:

$$\vec{x}_0, \;\; \vec{x}_1 = \mathbf{G}\vec{x}_0, \;\; \vec{x}_2 = \mathbf{G}\vec{x}_1, \ldots$$

donde $\vec{x}_0$ es *cualquier* vector de probabilidad inicial.

Omitiremos aquí más explicación acerca de este método excepto el hecho de que, cuando el parámetro α está "demasiado cercano" a 1, la convergencia de este método es muy lenta. Sergey Brin y Larry Page, los creadores del PageRank, eligieron un valor de α próximo a $0{,}85$.

4. Ejemplo trabajado con Scilab

Para ilustrar el método, vamos a calcular el vector PageRank correspondiente al ejemplo anterior con la ayuda de Scilab. Primero introduciremos la matriz G dada en (2) a partir del grafo que describe los enlaces, pero sustituyendo los ceros de las columnas nulas por $1/n$, donde n es el número de páginas ($n = 4$ en nuestro caso):

```
-->G=[0 0 1/3 1/4; 1 0 1/3 1/4; 0 1 0 1/4; 0 0 1/3 1/4]
 G  =

    0.    0.    0.3333333    0.25
    1.    0.    0.3333333    0.25
    0.    1.    0.           0.25
    0.    0.    0.3333333    0.25
```

Ahora definimos la "matriz de aleatoriedad" E:

```
-->E=1/4*ones(4,4)
 E  =

    0.25    0.25    0.25    0.25
    0.25    0.25    0.25    0.25
    0.25    0.25    0.25    0.25
    0.25    0.25    0.25    0.25
```

Definimos ahora la *matriz Google* $\mathbf{G}$ tomando $\alpha = 0{,}85$:

[3] La acumulación de errores de redondeo es una razón. Otra es que, como el número de páginas web es tan elevado, la matriz E es "casi" la matriz nula

[4] En la práctica se usa una versión modificada del *método de la potencia* que permite usar sólo la matriz G en lugar de $\mathbf{G}$, evitando el uso de E en los cálculos.

```
-->G=0.85*G+(1-0.85)*E
 G  =

    0.0375    0.0375    0.3208333    0.25
    0.8875    0.0375    0.3208333    0.25
    0.0375    0.8875    0.0375       0.25
    0.0375    0.0375    0.3208333    0.25
```

Resolvemos ahora el sistema (3), es decir, calculamos el núcleo de la matriz $\mathbf{G} - I_{4\times 4}$:

```
-->x=kernel(G-eye(4,4))
 x  =

    0.3254602
    0.6021013
    0.6523997
    0.3254602
```

El único vector de probabilidad que es solución del sistema (3) puede calcularse fácilmente dividiendo el generador del núcleo que hemos obtenido por la suma de sus componentes:

```
-->x/sum(x)
 ans  =

    0.1708075
    0.3159938
    0.3423913
    0.1708075
```

Este es el vector PageRank. Esto significa que las páginas se ordenan, en orden decreciente de importancia, de la siguiente manera:

$$Page\ 3$$

$$Page\ 2$$

$$Page\ 1$$

$$Page\ 4$$

En este caso, como sólo tenemos 4 páginas, es muy fácil ordenar las páginas "a mano". Sin embargo, en casos con un mayor número de páginas, puede resultar útil escribir el siguiente comando de Scilab:

```
-->[w,k]=gsort(x/sum(x))
 k  =

    3.
    2.
    1.
    4.
 w  =

    0.3423913
    0.3159938
    0.1708075
    0.1708075
```

El vector w que devuelve escribe, en orden decreciente, el PageRank de las páginas. El vector k proporcionar directamente la lista ordenada de las páginas.