Laboratorio computazionale numerico
Lezione 9

f.poloni&sns.it

2008-12-17

1 Come diventare ricchi con l'algebra lineare: Google

1.1 Modello

Il nostro progetto è quello di costruire un algoritmo che, dato un insieme di n pagine web, riesca a "dare un punteggio" di importanza a ognuna di esse (algoritmo PageRank). Il modo di farlo si basa su questa idea: prendiamo un "navigatore casuale" che clicchi ogni volta su un link a caso. Una pagina è tanto più importante quanto più tempo il navigatore casuale passa su di essa. Si può vedere che:

- L'importanza di una pagina dipende solo da come sono disposti i collegamenti tra una pagina e l'altra;
- Più pagine linkano a una pagina, e più essa è importante;
- In particolare, più pagine *importanti* linkano a una pagina, più essa è importante.

1.2 Matrici di adiacenza e di transizione

Andiamo a tradurre l'algoritmo nel linguaggio dell'algebra lineare. Costruiamo innanzitutto la *matrice di adiacenza* del nostro insieme di pagine web, cioè la matrice

$$A_{ij} = \begin{cases} 1 & \text{se c'è un link dalla pagina } i \text{ alla pagina } j. \\ 0 & \text{altrimenti.} \end{cases}$$

Per l'esempio della figura 1 si ha

```
octave:6> A
A =

   0   1   0   1   0
   1   0   0   0   1
   0   0   0   1   1
   1   0   1   0   0
   1   0   1   0   0
```

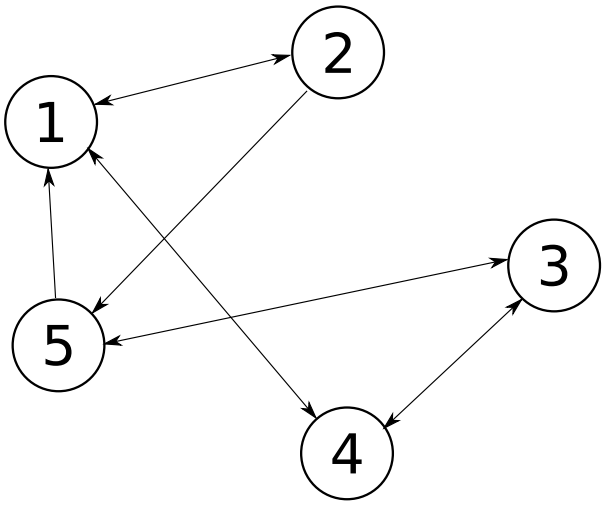


Figura 1: Un esempio di insieme di pagine web

Questa matrice descrive completamente la figura. A partire da essa, possiamo costruire la *matrice di transizione* B:

$$B_{ij} = \{\text{probabilità di passare dalla pagina } i \text{ alla pagina } j\} :$$

```
B =

   0.00000   0.50000   0.00000   0.50000   0.00000
   0.50000   0.00000   0.00000   0.00000   0.50000
   0.00000   0.00000   0.00000   0.50000   0.50000
   0.50000   0.00000   0.50000   0.00000   0.00000
   0.50000   0.00000   0.50000   0.00000   0.00000
```

Esercizio 1. Scrivere una **function** B=adj2trans(A) che, data una matrice di adiacenza, costruisca la corrispondente matrice di transizione. (Hint: questo più che un problema di programmazione è un problema di capire cosa bisogna fare: qual è il legame tra le due matrici?)

1.3 Dalla probabilità all'algebra lineare

Chiamiamo $p_{i,t}$ la probabilità che il "navigatore casuale" si trovi nella pagina i all'istante t. La probabilità che all'istante $t+1$ esso sia nella pagina j è uguale a

$$p_{j,t+1} = \sum_i p_{i,t} B_{i,j} \tag{1}$$

(perché?). Possiamo tradurre questa relazione in una relazione tra vettori in questo modo: prendiamo il vettore riga

$$p^{(t)} = \begin{bmatrix} p_{1,t} & p_{2,t} & \dots & p_{n,t}; \end{bmatrix} \tag{2}$$

la (1) diventa allora

$$p^{(t+1)} = p^{(t)} B.$$

Inoltre, abbiamo questo vincolo: a un qualunque tempo t, la somma delle probabilità che il navigatore sia in uno qualunque degli stati deve fare 1[1]:

$$p_1^{(t)} + p_2^{(t)} + \cdots + p_n^{(t)} = 1. \tag{3}$$

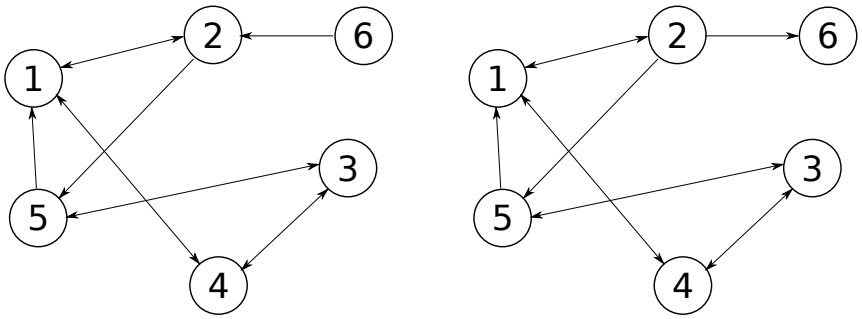
Teorema 1. *Se B è irriducibile, più un'altra "ipotesi tecnica" (il grafo di partenza ha almeno due cicli di lunghezza prima tra loro), allora, per qualunque vettore di partenza $p^{(0)}$ che soddisfi la proprietà richiesta, si ha che $p^{(t)}$ converge per $t \to \infty$ al vettore che dà i "tempi medi" che il navigatore casuale passa su ogni pagina.*

Quindi per ottenere il tempo medio passato su ogni pagina (che nel nostro modello misura l'importanza della pagina) dobbiamo semplicemente "far passare molto tempo":

```
octave:22> p=[1 0 0 0 0];
octave:23> for k=1:500; p=p*B; endfor ;
octave:24> p
p =
  0.26667   0.13333   0.20000   0.23333   0.16667
```

Quindi, nel nostro modello, la pagina più "importante" è la 1, seguita dalla 4, la 3, la 5 e infine la meno importante è la 2.

Esercizio 2. Provare ad applicare tutto l'algoritmo a partire da qualche altra rete. Provare in particolare anche quelle della figura 2 (rispetto alle precedenti, cambia solo l'aggiunta di un nodo): cosa succede al vettore p? Come mai?



Il problema che si verifica con le due reti precedenti si può risolvere con un piccolo cambiamento al nostro modello. Supponiamo che ad ogni passo il navigatore abbia una certa probabilità α di seguire un link della pagina in cui si trova e una certa probabilità $1 - \alpha$ di "teletrasportarsi" su una pagina a caso (ognuna con probabilità $1/n$). Tradizionalmente si sceglie $\alpha = 0.85$. Come cambia la matrice di transizione?

[1] In realtà, se $p^{(0)}$ soddisfa questa proprietà e se B soddisfa $Be = e$, dove u è il vettore di tutti uni, allora questa proprietà è valida a ogni tempo t (provare a dimostrarlo se vi avanza tempo).

1.4 Se vi state annoiando...

In realtà, nella pratica i calcoli con le matrici non possono essere fatti in questa forma (su $n \approx 10^9$ pagine web il costo $2n^3$ del prodotto matrice vettore è troppo elevato...). Quello che si fa è sfruttare il fatto che la matrice è *sparsa* (contenente moltissimi zeri), perché ogni pagina non può avere troppi link uscenti. Invece di memorizzare l'intera matrice, possiamo memorizzare la matrice L di dimensione $l \times 2$ dei "links", che ha una riga contenente (i, j) per ogni link dalla pagina i alla pagina j: per esempio nell'esempio originario

$$L = \begin{bmatrix} 1 & 2 \\ 1 & 4 \\ 2 & 1 \\ 2 & 5 \\ 3 & 4 \\ 3 & 5 \\ 4 & 1 \\ 4 & 3 \\ 5 & 1 \\ 5 & 3 \end{bmatrix}.$$

Esercizio 3. Riscrivere l'algoritmo utilizzando la matrice L invece della matrice B, senza costruire esplicitamente quest'ultima.