Google PageRank ou une façon de classer les pages web

Olivier GUIBÉ
Laboratoire de Mathématiques Raphaël Salem
CNRS-Université de Rouen

Grève de février 2009

Pour celles et ceux qui se souviennent

Les moteurs de recherche tels que Yahoo, Altavista, Lycos ont été dépassés par Google :

- création de Google en 1998 par Brin et Page
- origine du PageRank : article de Brin et Page « The Anatomy of a Large-Scale Hypertextual Web Search Engine », Stanford University, 1998
- algèbre linéaire, algorithmique : classement des pages –souvent– pertinent

But

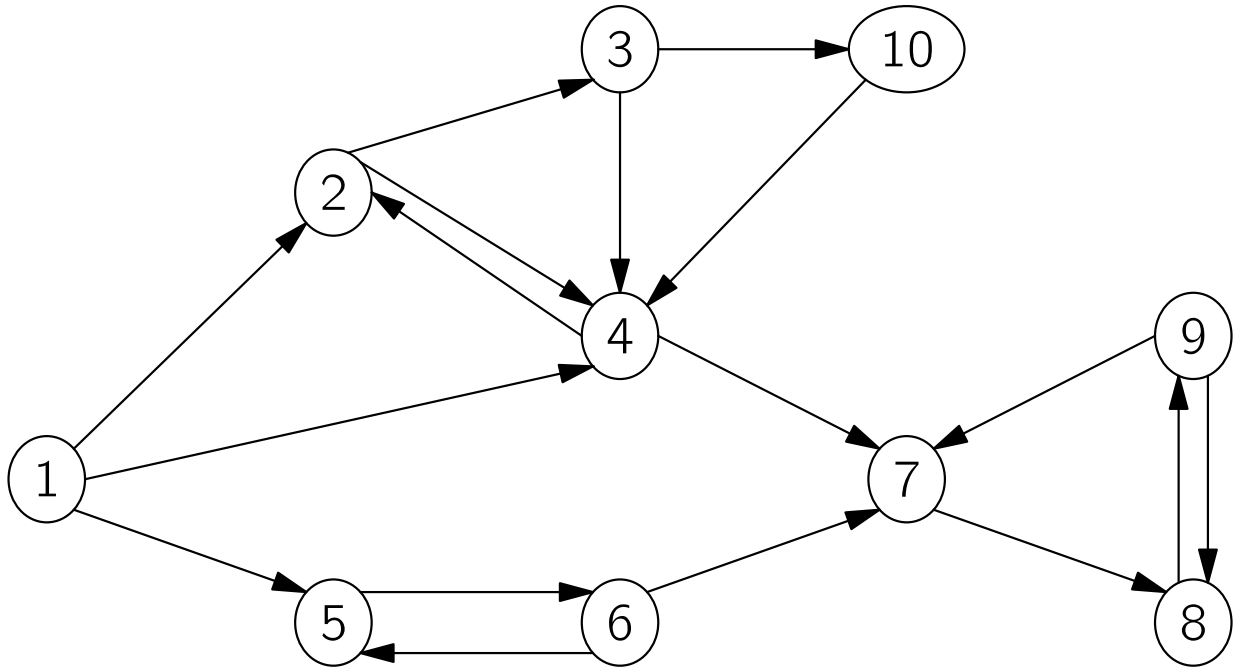
Un algorithme qui permet de classer les pages web

- qui peut être effectivement implémenté (nous ne le verrons pas)
- donner un (unique) classement pertinent pour l'utilisateur

Nous ne nous intéresserons pas au contenu (c'est un tort).

Idée

Une « flèche » de 1 vers 2 veut dire que la page 1 contient un lien vers la page 2, etc. Voici un modèle très réduit !



Plus une page est la cible de liens venant d'autres pages plus elle a de chance d'être fiable et intéressante

Mais

Un classement qui ne tient compte que du nombre de liens :

- trop naïf
- facile d'augmenter le rang d'une page A en créant des pages qui pointent vers A
- même défaut avec une pondération fonction du nbre de liens sur les pages

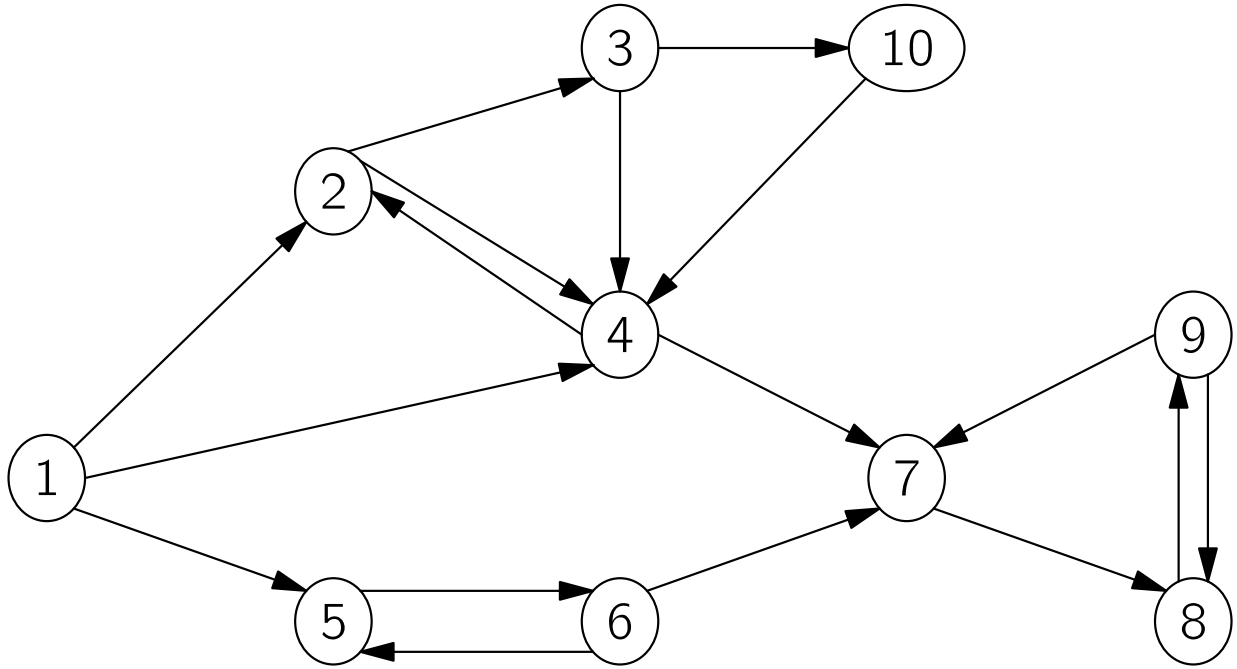
La page 4 serait classée en 1er alors que 7 et 8 semblent plus pertinentes.
Il serait moins naïf de demander :

une page i est importante si beaucoup de pages importantes pointent vers i

Transformation en un tableau ou une matrice

Tableau C où c_{ij} (ligne i, colonne j) contient 1 si j pointe sur i et 0 sinon.

On décide que $c_{ii} = 0$; ne pas tenir compte des liens internes.



Transformation en un tableau ou une matrice

Tableau C où c_{ij} (ligne i, colonne j) contient 1 si j pointe sur i et 0 sinon.

On décide que $c_{ii} = 0$; ne pas tenir compte des liens internes.

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Beaucoup de zéros !

Transformation en un tableau ou une matrice

Tableau C où c_{ij} (ligne i, colonne j) contient 1 si j pointe sur i et 0 sinon.

On décide que $c_{ii} = 0$; ne pas tenir compte des liens internes.

on compte le nombre de liens sur chaque page (en colonne)

$$\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

$$\begin{pmatrix}
3 & 2 & 2 & 2 & 1 & 2 & 1 & 1 & 2 & 1 \\
N_1 & N_2 & & & & & & & &
\end{pmatrix}$$

Score de pertinence

On cherche le rang de chaque page i, noté r_i, qui vérifie pour tout $i \in \{1, \ldots, N\}$

$$r_i = \sum_{j=1}^{N} \frac{c_{ij}}{N_j} r_j$$

- la somme tient compte à la fois du score de chaque page pondéré avec le nbre de liens
- si $N_j = 0$ le terme n'apparaît pas dans la somme
- l'ajout de pages « vides de sens » modifiera très peu le rang (auront un r_j proche de zéro)
- c'est une équation à résoudre !

Des maths !

Posons la matrice Q

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

Notre problème revient à trouver un vecteur r à composantes positives vérifiant $Qr = r$. r est vecteur propre de Q associé à la valeur propre 1.

Premier défaut

Il se peut que le vecteur r n'existe pas

Pour cela prendre l'exemple –extrème– avec deux pages isolées ! Ceci est directement lié à $N_k = 0$: une page qui ne pointe sur aucune autre.

Solution

On transforme la matrice Q : si une colonne ne contient que des zéros on remplace tous les termes cette colonne par $1/N$.

Conséquence

La matrice Q ainsi transformée est une matrice stochastique : la somme des termes d'une colonne vaut 1.

Exercice

Montrer que 1 est valeur propre de Q.

Second défaut

Le score r n'est pas unique

Pour cela prendre un ensemble de pages web A et un ensemble B tel qu'il n'y ait aucun lien entre A et B

Solution/transformation

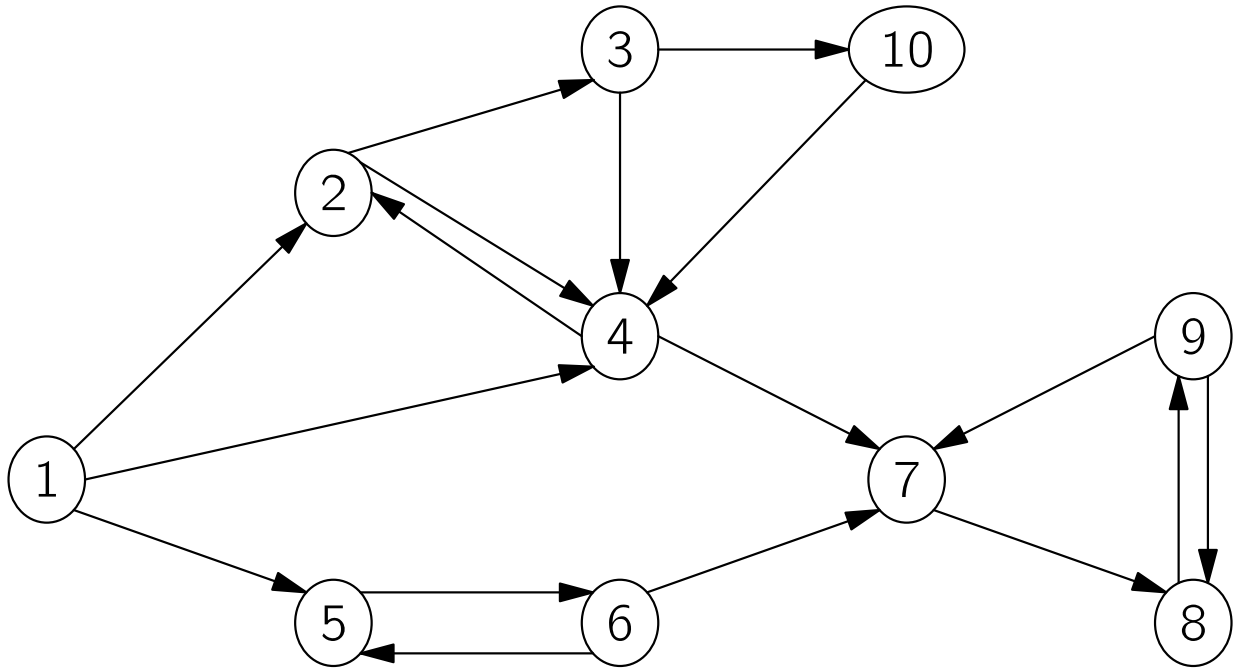
Le surfeur peut passer de la page i à

- une des pages référencées par i avec la probabilité α
- une page quelconque de façon équiprobable $(1-\alpha)/N$
- on travaille sur $A = \alpha Q + (1-\alpha)\frac{1}{N}J$ où J est la matrice ne contenant que des 1. On prendra $\alpha = .85$.

Le matheux est content !

Il existe un unique vecteur r vérifiant $Ar = r$. (théorème de Perron-Frobenius)

Et nos pages ?



Classement ordre décroissant

8, 9, 7, 4, 2, 6, 5, 3, 10, 1

Comment faire ? méthode irréaliste

Ce n'est pas un problème facile. Une méthode connue et efficace est la méthode de la puissance. On construit une suite de vecteur

- q_0 vecteur initial
- $z_k = Aq_{k-1}$, $q_k = z_k / \|z_k\|_1$ (on divise par la somme des composantes pour avoir une somme des composantes égale à 1)

Pour k grand q_k sera « proche » de r.

Mais les pages web étaient de l'ordre de 10^{10} en 2005 !

Conséquence

Impossibilité de stocker A et de calculer rapidement Aq_{k-1} !

Comment faire ?
idée dans la vraie vie de Google

Il faut se rappeler que la matrice Q contient beaucoup de zéros : c'est une matrice dite creuse. On utilise alors avec profit cette structure

- on se stocke pas les zéros mais juste les listes (i, j) et q_{ij} avec $q_{ij} \neq 0$: on passe de N^2 à $N \times \bar{\ell}$ avec $\bar{\ell}$ le nombre moyen de liens (lourd mais possible)
- on adapte les algorithmes (une boucle en N contenant une boucle en $\bar{\ell}$)

Il faut tout de même une puissance de calcul, une capacité de stockage très importante, ce dont dispose Google.
Tous les jours ça marche !

- si je créé une page web j'ajoute effectivement des liens que je trouve intéressants
- le succès est tel qu'aujourd'hui une page est importante parce qu'elle est bien classée
- le classement est crucial pour les sites commerciaux, on peut l'améliorer en choisissant bien mais peu les liens présents
- l'algorithme a évolué, la base est toujours le PageRank mais les évolutions sont secrètes
- le PageRank est calculé plusieurs fois par an. Il y a donc d'autres mécanismes qui permettent d'ajouter des nouvelles pages entre deux PageRank et aussi de modifier le classement en fonction des « cliques » sur les recherches
- TrustRank, nofollow, etc.
- un peu d'algèbre linéaire et d'algorithmique peuvent rapporter !

Trois liens ou presque

- sujet de modélisation à l'agrégation de mathématiques : texte sur Google
- excellent sujet de « popularisation » (niveau L3) par Michael Eisermann : Comment fonctionne Google ?
- histoire de se distraire : sujet de concours de l'ESSEC 2008, épreuve 2, voie économique (à chercher sur Google !)