PageRank

Modélisation

IUT Informatique
Fontainebleau

Introduction

La plupart des gens utilisent des moteurs de recherche pour trouver ce qu'ils cherchent sur internet. Avec l'explosion du nombre de pages Web, le problème de classement des résultats de recherche est devenu primordial.

Les premiers moteurs de recherche (AltaVista ou Yahoo) ne faisaient essentiellement qu'indexer, c'est-à-dire trouver toutes les pages contenant le ou les mots-clés recherchés. Le nombre de fois où apparaissait le mot-clé faisait apparaître la page en haut de la liste de résultats, ce qui n'est pas pertinent car aisément falsifiable.

Sergey Brin et Lawrence Page, étudiants à l'Université Standford, ont trouvé une solution originale : utiliser l'information des liens entre les pages pour mesurer leur importance et classer les résultats d'une recherche de mots-clés.

PageRank : idée

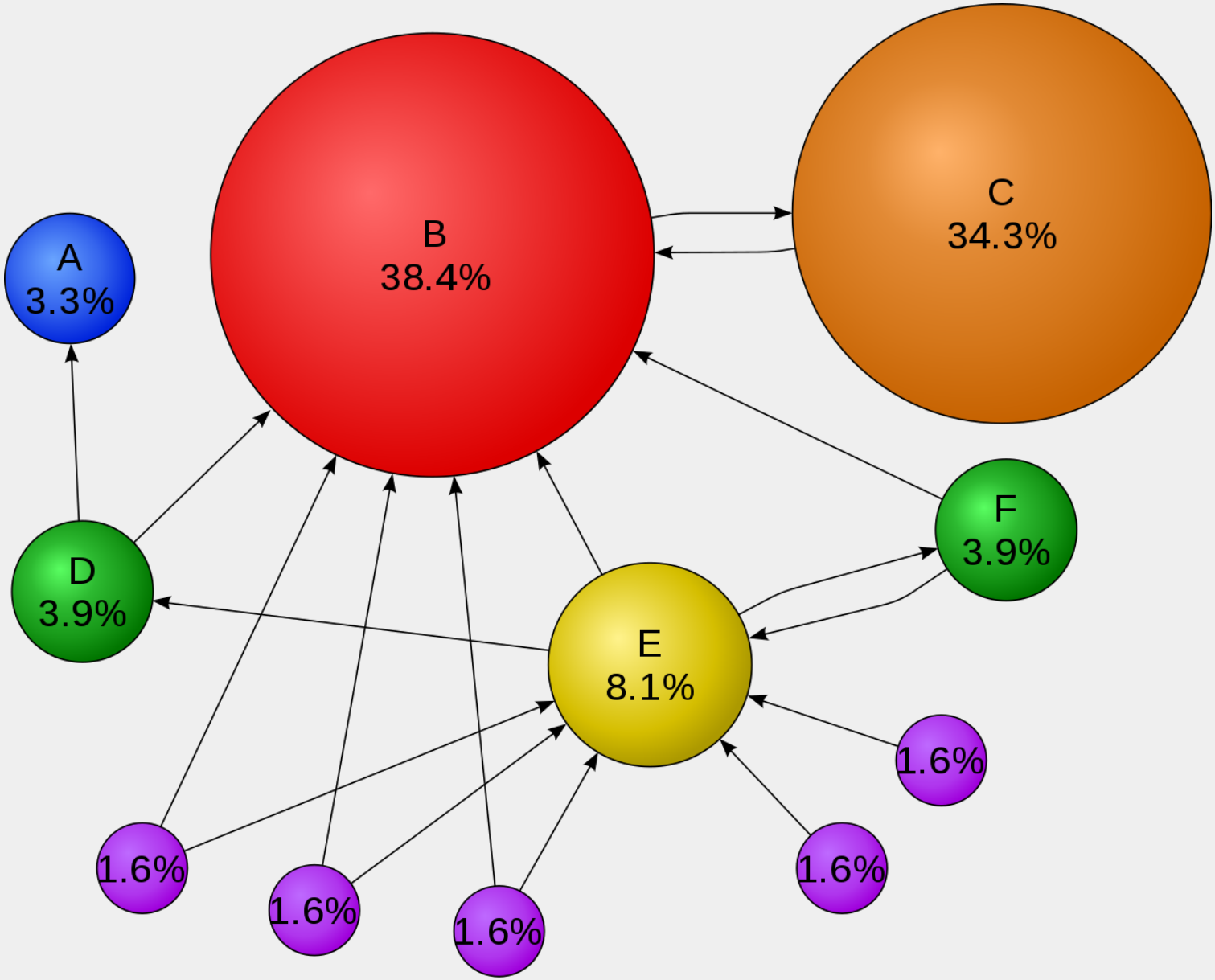
Leur algorithme PageRank calcule un indice de popularité associé à chaque page Web. C'est cet indice qui est utilisé pour trier le résultat d'une recherche.

Il s'inspire du Science Citation Index fondé par Eugène Garfield en 1964, un indice de classement des articles scientifiques en fonction du nombre de citations.

PageRank (1998) reprend le principe de la citation et y substitue la notion de lien entrant : « L'indice de popularité d'une page est d'autant plus grand qu'elle a un grand nombre de pages populaires la référençant (ayant un lien vers elle) ».

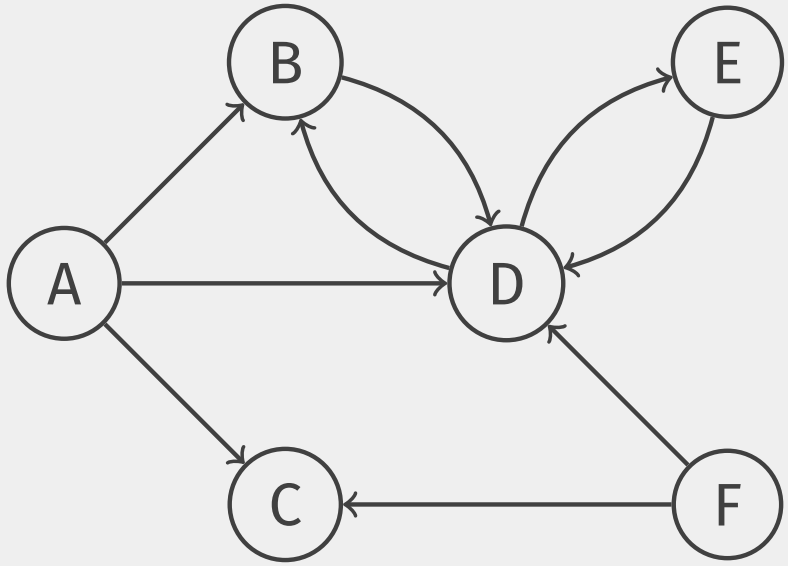
Cette définition est auto-référente car pour connaître l'indice d'une page, il faut connaître l'indice des pages ayant un lien vers elle.

PageRank : illustration



PageRank : formalisation

On voit le Web comme un graphe orienté. Chaque page est un nœud du graphe, chaque lien entre les pages est un arc entre deux nœuds.

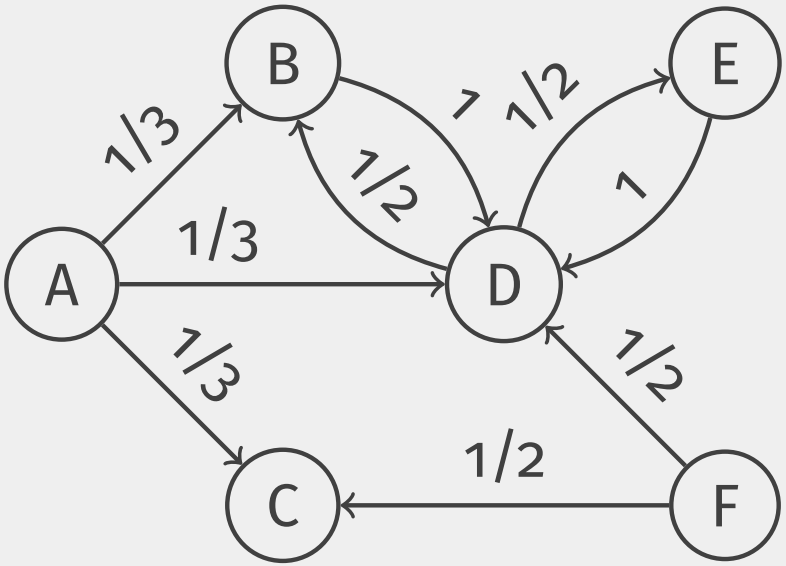


En comptant les liens (“votes”), on obtient le classement suivant :

1: D (4), 2: B et C (2), 3: E (1), 4: A et F (0)

Question: est-il normal que B et C ont le même score ? Quel classement semble plus logique ?

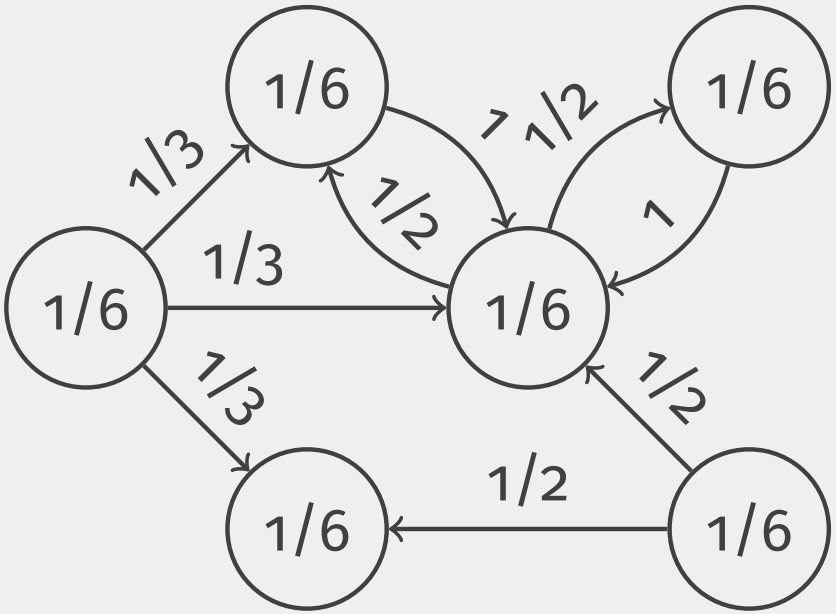
On doit éviter d'abord que les "gros votants" aient plus d'influence que ceux qui votent plus "ciblé". On divise les votes par le nombre des liens, en supposant que ne pas voter revient à voter pour tous. On obtient une matrice markovienne $P = (p_{ij})$:



$$\begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

Ses lignes représentent les votes émis et les colonnes les votes reçus.

On attribue à chaque page un score (poids de vote) initial égal à $1/6$ et on le partage entre les pages référencées :



$$p_A^{(1)} = \underbrace{\frac{1}{6}}_{p_C^{(0)}} \times \underbrace{\frac{1}{6}}_{p_{CA}} = \frac{1}{36}$$

$$p_B^{(1)} = \frac{1}{6} \times \frac{1}{3} + \frac{1}{6} \times \frac{1}{6} + \frac{1}{6} \times \frac{1}{2} = \frac{1}{6}$$

etc

Le nouveau vecteur de scores est donc

$$p^{(1)} = \left(\frac{1}{36}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2}, \frac{5}{36}, \frac{1}{36} \right)$$

On remarque que ce vecteur est obtenu par multiplication de $p^{(0)} = (1/6, \ldots, 1/6)$ par P :

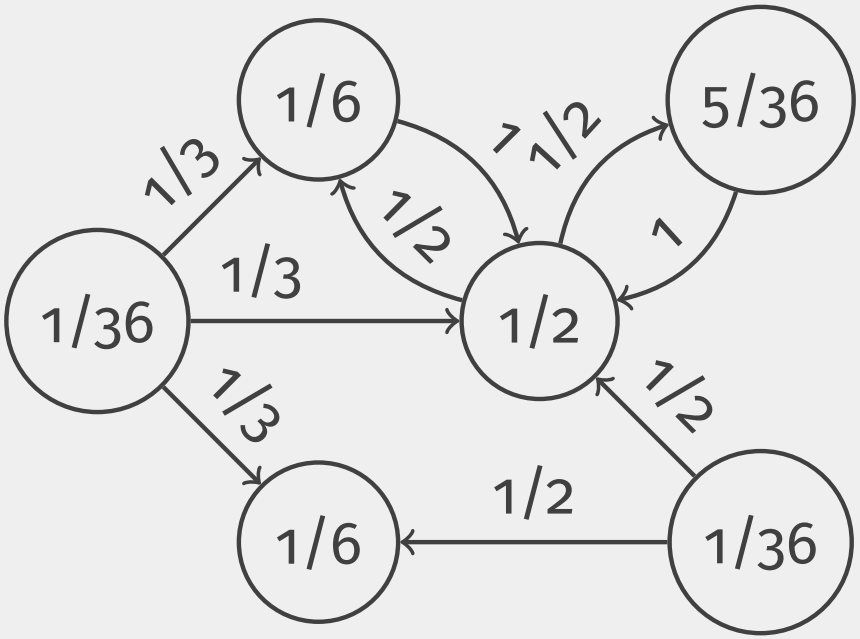
$$p_j^{(1)} = \sum_i p_i^{(0)} p_{ij},$$

càd

$$p^{(1)} = (p_A^{(0)}, \ldots, p_F^{(0)}) \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}$$

On peut recalculer les scores plusieurs fois en utilisant les scores précédents :

$$p^{(n+1)} = p^{(n)} P$$



$$p_A^{(2)} = \underbrace{\frac{1}{6}}_{p_C^{(1)}} \times \underbrace{\frac{1}{6}}_{p_{CA}} = \frac{1}{36}$$

$$p_B^{(2)} = \frac{1}{36} \times \frac{1}{3} + \frac{1}{6} \times \frac{1}{6} + \frac{1}{2} \times \frac{1}{2} = \frac{31}{108}$$

etc

Dans notre exemple on obtient

$$p^{(1)} = (0.0277, 0.1666, 0.1666, 0.5, 0.1111, 0.0277)$$

$$p^{(2)} = (0.0277, 0.2870, 0.0509, 0.3287, 0.2777, 0.0277)$$

$$p^{(3)} = (0.0084, 0.1820, 0.0316, 0.5964, 0.1728, 0.0084)$$

etc. On espère que les scores se “stabilisent” après plusieurs itérations autour d’un score final.

Si un vecteur limite $p = \lim_{n\to\infty} p^{(n)}$ existe, il vérifie l’équation

$$p = \lim_{n\to\infty} p^{(n+1)} = \lim_{n\to\infty} p^{(n)}P = pP$$

qui signifie que le score de chaque page est égal à la somme des votes pondérée par les scores de pages la référençant.

PageRank : difficultés

- Il se trouve que le vecteur de scores ne se stabilise pas toujours. Dans notre cas il oscille entre deux états :

$$(0, 0.3108, 0, 0.3783, 0.3108, 0)$$

et

$$(0, 0.1891, 0, 0.6216, 0.1891, 0)$$

- L'équation $pP = p$ admet bien une solution

$$p = (0, 0.25, 0, 0.5, 0.25, 0),$$

mais les scores de A, C et F sont nuls et leurs votes sont donc ignorés.

On peut expliquer ces phénomènes en considérant une chaîne de Markov associée ($p^{(n)}$ = distribution à l'instant n).

PageRank : solution

Pour pallier à ces difficultés, Brin et Pages ont proposé de "diluer" la matrice P, en prélevant à chaque page une fraction α de son score et en la distribuant entre toutes les pages :

$$P_\alpha = (1-\alpha)P + \alpha J,$$

où

$$J = \begin{pmatrix} 1/N & \dots & 1/N \\ \vdots & \dots & \vdots \\ 1/N & \dots & 1/N \end{pmatrix}$$

et N est le nombre des pages.

La matrice P_α étant non-creuse, le théorème de Perron-Frobenius affirme que $p^{(n)}$ converge vers son unique état stable p_α.

Dans notre exemple, avec $\alpha = 0.15$, on obtient

$$p_\alpha = (0.0329, 0.2266, 0.0563, 0.4337, 0.2173, 0.0329)$$

Classement

1. D
2. B
3. E
4. C
5. A et F