

## Projet PageRank

---

L'objectif de ce projet est d'étudier sur l'exemple du classement des pages web, la méthode de la puissance. Ce sujet est fortement inspiré d'un sujet d'agrégation option B, 2008.

*Le projet consiste à écrire un compte rendu donnant les réponses aux différentes questions posées ci-dessous et à joindre au compte-rendu une copie des programmes informatiques effectués.*

La recherche d'informations pertinentes sur le Web est un des problèmes les plus cruciaux pour l'utilisation de ce dernier. Des enjeux économiques colossaux sont en jeu, et diverses multinationales se livrent à de grandes manoeuvres. Le leader actuel de ce marché, Google, utilise pour déterminer la pertinence des références fournies, un certain nombre d'algorithmes dont certains sont des secrets industriels jalousement gardés, mais d'autres sont publics. On va s'intéresser ici à l'algorithme PageRank<sup>1</sup>, lequel fait intervenir des valeurs propres et vecteurs propres d'une énorme matrice.

## 1 Le principe de l'algorithme PageRank

On peut considérer pour simplifier que le Web est une collection de  $N \in \mathbb{N}$  pages, avec  $N$  très grand (de l'ordre de  $10^{10}$  en octobre 2005). La plupart de ces pages incluent des liens hypertextes vers d'autres pages. On dit qu'elles pointent vers ces autres pages. L'idée de base utilisée par les moteurs de recherche pour classer les pages par ordre de pertinence décroissante consiste à considérer que plus une page est la cible de liens venant d'autres pages, c'est-à-dire plus il y a de pages qui pointent vers elle, plus elle a de chances d'être fiable et intéressante pour l'utilisateur final, et réciproquement. Il s'agit donc de quantifier cette idée, c'est-à-dire d'attribuer un score de pertinence à chaque page.

Pour ce faire, on représente le Web comme un graphe orienté : un *graphe* est un ensemble de points, dont certaines paires sont directement reliées par un "lien". Ces liens peuvent être orientés, c'est-à-dire qu'un lien entre deux points  $u$  et  $v$  relie soit  $u$  vers  $v$ , soit  $v$  vers  $u$  : dans ce cas, le graphe est dit orienté. Sinon, les liens sont symétriques, et le graphe est non-orienté. Les points sont généralement appelés sommets, et les liens "arêtes". On peut représenter le graphe par une matrice, qu'on appelle matrice d'adjacence : le coefficient de la  $i$ -ème ligne et  $j$ -ème colonne est 1 s'il existe une arête allant du sommet  $i$  au sommet  $j$ , et 0 sinon. Remarquer que si le graphe est non orienté, alors une arête est définie comme liant les sommets  $i$  et  $j$  sans ordre, et la matrice d'adjacence est symétrique.

Pour représenter le Web comme un graphe, on se donne donc un ordre arbitraire sur l'ensemble des pages que l'on numérote ainsi de  $i = 1$  à  $i = N$  : ce sont les sommets du graphe. La structure de connectivité du Web peut alors être représentée par la matrice d'adjacence  $C$  de taille  $N \times N$  telle que  $C_{ij} = 1$  si la page  $j$  pointe sur la page  $i$ ,  $C_{ij} = 0$  sinon. Remarquons que c'est un graphe non symétrique. Les liens d'une page sur elle-même ne sont pas significatifs, on pose donc  $C_{ii} = 0$ . Remarquons que la ligne  $i$  de la matrice  $C$  contient tous les liens significatifs qui pointent sur la page  $i$ , alors que la colonne  $j$  contient tous les liens significatifs qui partent de la page  $j$ .

---

1. Le nom "PageRank" vient du nom de Larry Page, l'un des inventeurs de Google.

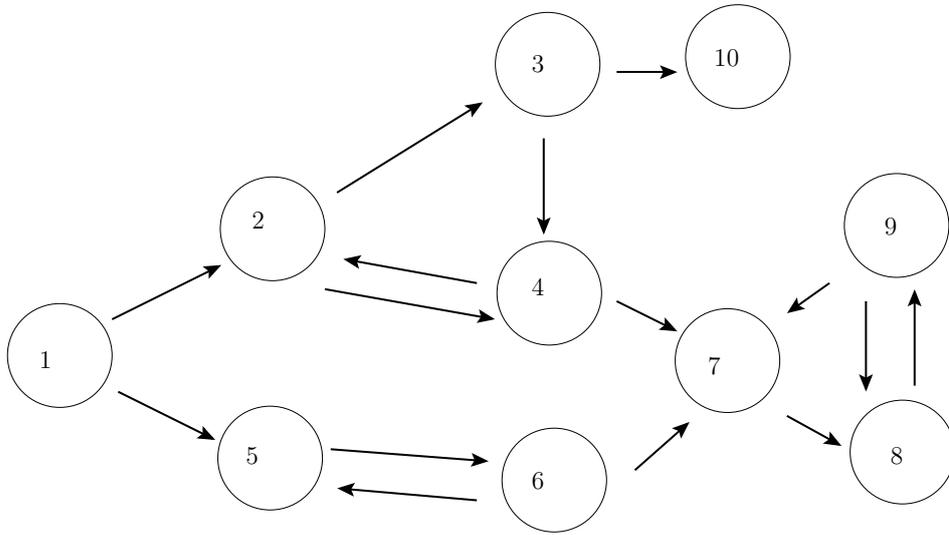


FIGURE 1 – Exemple 1

Pour illustrer notre étude, on propose les trois exemples de graphe décrits sur les figures 1, 2 et 3. .

1. *Ecrire les matrices  $C_1$ ,  $C_2$  et  $C_3$  associées respectivement aux exemples 1, 2 et 3.*

On souhaite attribuer à chaque page  $i$  un score  $r_i \in \mathbb{R}_+^*$  de façon à pouvoir classer l'ensemble des pages par score décroissant et présenter à l'utilisateur une liste ainsi classée des pages correspondant à sa requête. L'algorithme PageRank part du principe qu'un lien de la page  $j$  pointant sur la page  $i$  contribue positivement au score de cette dernière, avec une pondération par le score  $r_j$  de la page dont est issu le lien (une page ayant un score élevé a ainsi plus de poids qu'une n'ayant qu'un score médiocre) et par le nombre total de liens présents sur ladite page  $N_j = \sum_{k=1}^N C_{kj}$ . On introduit donc la matrice  $Q$  définie par  $Q_{ij} = \frac{C_{ij}}{N_j}$  si  $N_j \neq 0$ ,  $Q_{ij} = 0$  sinon.

2. *Déterminer les matrices  $Q_1$ ,  $Q_2$  et  $Q_3$  associées aux trois exemples. Vérifier que la somme des coefficients des colonnes non nulles de  $Q$  vaut toujours 1.*
3. *Démontrer que toutes les colonnes non nulles de la matrice  $Q$  générale sont telles que la somme de leurs coefficients est égale à 1.*

L'application des principes ci-dessus conduit donc à une équation pour le vecteur  $r \in \mathbb{R}^N$  des scores des pages de la forme

$$(1) \quad r_i = \sum_{j=1}^N Q_{ij} r_j \text{ c'est à dire } r = Qr,$$

où  $Q$  est une matrice dont la somme des coefficients de chaque colonne non nulle est égale à 1. Le problème du classement des pages du Web se trouve ainsi ramené à la recherche d'un vecteur propre d'une énorme matrice, associé à la valeur propre 1 ! Mais il peut arriver que la matrice  $Q$  n'admette pas la valeur propre 1 ce qui invalide quelque peu la philosophie originale de l'algorithme.

4. *Vérifier avec un programme informatique que les matrices  $Q_1$  et  $Q_3$  admettent 1 comme valeur propre mais que ce n'est pas le cas de  $Q_2$ .*

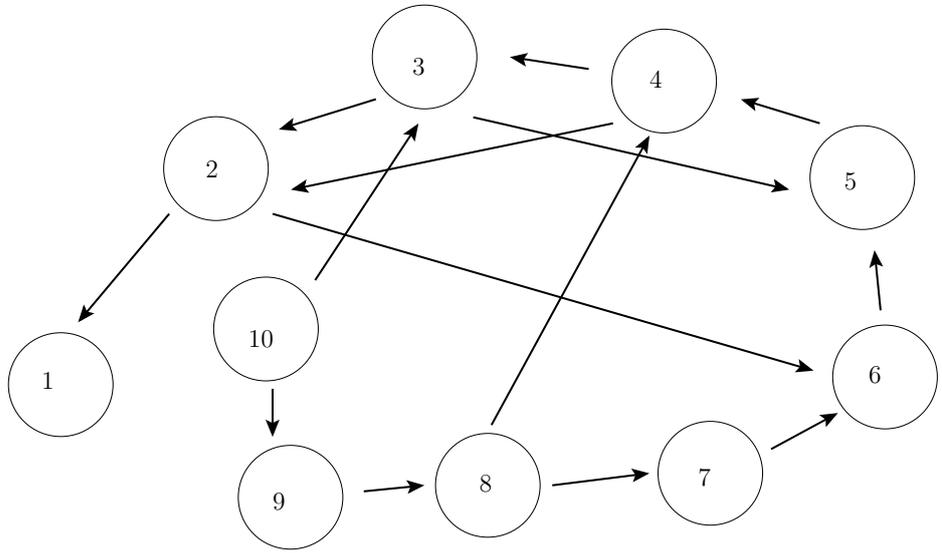


FIGURE 2 – Exemple 2

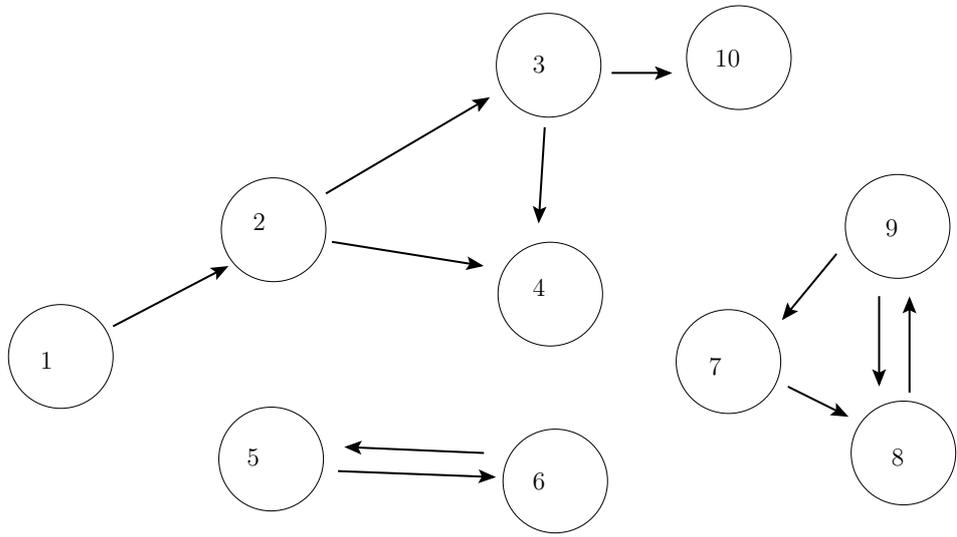


FIGURE 3 – Exemple 3

5. Soit  $Q$  la transposée d'une matrice stochastique<sup>2</sup>

(a) Montrer que 1 est valeur propre de  $Q^t$  et donc de  $Q$ .

(b) Montrer que  $\rho(Q^t) = \rho(Q) = 1$ .

Pour faire en sorte que 1 soit valeur propre, on va donc modifier la matrice  $Q$  de manière à ce qu'elle soit la transposée d'une matrice stochastique, et donc en particulier qu'elle n'ait plus de colonne nulle. On considère pour cela les vecteurs  $e = (1\ 1 \cdots 1)^t \in \mathbb{R}^N$  et  $d \in \mathbb{R}^N$  de composantes  $d_j, j = 1, \dots, N$ , avec  $d_j = 1$  si  $N_j = 0$ ,  $d_j = 0$  sinon. On définit alors la matrice

$$(2) \quad P = Q + \frac{1}{N}ed^t.$$

6. Visualiser sur ordinateur les matrices  $P_1, P_2$  et  $P_3$  associées aux trois exemples et calculer leurs valeurs propres.

7. Montrer que le passage de  $Q$  à  $P$  revient à remplacer les colonnes de zéros de  $Q$  par des colonnes dont toutes les composantes sont égales à  $\frac{1}{N}$ .

8. Montrer que  $P$  est bien la transposée d'une matrice stochastique, et en déduire que  $P$  admet bien la valeur propre 1 et que  $\rho(P^t)$  et  $\rho(P)$  sont égaux à 1.

On remarque sur l'exemple 3 que la valeur propre 1 peut être multiple. Or, notre problème consiste à trouver un vecteur propre associé à cette valeur propre et il est alors préférable que 1 soit valeur propre simple. On va donc encore modifier la matrice de manière à ce que tous ses coefficients soient strictement positifs. En effet, on a le théorème suivant :

**Théorème 1 (Perron-Frobenius, cas des matrices stochastiques)** Soit  $A$  une matrice transposée d'une matrice stochastique et qui est de plus strictement positive (c.à.d. dont tous les coefficients sont strictement positifs). Alors  $\rho(A) = 1$ , et 1 est une valeur propre simple ; de plus, il existe un vecteur propre  $r$  strictement positif associé à la valeur propre 1 .

Ce théorème est important pour nous car il va nous permettre non seulement de construire une matrice avec 1 comme valeur propre simple, mais de plus d'obtenir le vecteur  $r$  donnant un classement des pages du web. La question 9 a pour objet de démontrer le théorème de Perron-Frobenius. En fait, sous les hypothèses de ce théorème, on sait déjà par les questions précédentes que  $\rho(A) = 1$  et que 1 est valeur propre. Il reste à montrer que 1 est valeur propre simple et qu'il existe un vecteur propre  $r$  strictement positif associé à la valeur propre 1.

9. Soit  $A = B^t$  où  $B$  est une matrice stochastique strictement positive.

(a) Montrer que le sous-espace propre associé à la valeur propre 1 de  $B$  est  $\mathbb{R}e$ .

(b) En déduire que la valeur propre 1 de  $A$  est simple.

(c) Soit  $f$  un vecteur propre de  $A$  pour la valeur propre 1 .

i. Montrer que  $|f_i| < \sum_j a_{ij}|f_j|$  sauf si les  $f_j$  sont tous de même signe.

ii. En raisonnant sur  $\sum_i |f_i|$ , en déduire que les  $f_j$  sont tous de même signe.

Dans le but d'obtenir une matrice strictement positive, on effectue alors une dernière modification sur la matrice en choisissant un nombre  $0 < \alpha < 1$  et en posant

$$(3) \quad A_\alpha = \alpha P + (1 - \alpha) \frac{1}{N}ee^t.$$

---

2. On appelle *matrice stochastique* une matrice dont tous les coefficients appartiennent à  $[0, 1]$  et dont la somme des coefficients de chaque ligne vaut 1.

10. *Ecrire un programme calculant les matrices  $A_{\alpha,1}$ ,  $A_{\alpha,2}$  et  $A_{\alpha,3}$  associées aux trois exemples pour  $\alpha = 0.1$  et  $\alpha = 0.5$ . Calculer les modules valeurs propres de ces matrices et classez les par ordre décroissant.*
11. *Montrer que  $A_\alpha$  est toujours la transposée d'une matrice stochastique, et qu'elle est de plus strictement positive. En déduire que  $\rho(A_\alpha) = 1$ , que 1 est une valeur propre simple de  $A_\alpha$  et qu'il existe un vecteur propre  $r_\alpha$  strictement positif associé à la valeur propre 1.*

Finalement, PageRank calcule un tel vecteur propre  $r_\alpha \in \mathbb{R}^N$ , normalisé d'une façon ou d'une autre, qui est tel que

$$(4) \quad r_\alpha = A_\alpha r_\alpha,$$

dont les  $N$  composantes fournissent le classement recherché des pages du Web. On remarquera que pour  $N$  grand, la matrice  $A_\alpha$  n'est qu'une petite perturbation de la matrice  $Q$ . On sait combien cette stratégie s'est révélée efficace, puisque Google a totalement laminé les moteurs de recherche de première génération, comme Altavista, lesquels ont essentiellement disparu du paysage.

## 2 Calcul effectif du score des pages web

On décrit dans cette section des méthodes pour approcher ce vecteur  $r$ .

1. *Programmer la méthode de la puissance : pour  $r_0 \neq 0$*

$$(5) \quad q_k = A_\alpha r_{k-1}, \quad r_k = \frac{q_k}{\|q_k\|_1}.$$

*On normalisera les vecteurs en utilisant la norme 1.*

*Approcher le vecteur  $r$  solution de  $r = A_\alpha r$  pour nos trois exemples pour  $\alpha = 0.1$ . On pourra discuter le choix du test d'arrêt utilisé pour stopper l'algorithme.*

*Vérifier numériquement que  $\|r_k - r\|_2$  se comporte (en fonction de  $k$ ) sensiblement comme  $|\alpha\mu_2|^k$  où  $\mu_2$  est la seconde plus grande valeur propre (en module) de  $P$ .*

*Tester la sensibilité de la méthode au choix de  $\alpha$ . En particulier on pourra répondre aux deux questions suivantes :*

- (a) *Y a-t-il une influence du choix de  $\alpha$  (dans l'intervalle  $]0, 1[$ ) sur le classement final (on espère que la réponse est "non") ?*
- (b) *Comment le nombre d'itérations dans la méthode de la puissance varie avec  $\alpha$  ?*

On remarque que les matrices  $A_\alpha$  sont des matrices pleines alors que la matrice initiale  $Q$  était creuse. Il est en pratique hors de question d'assembler cette matrice  $A_\alpha$ .

2. *Montrer que si  $z \in \mathbb{R}^N$ ,  $z \geq 0$  avec  $\|z\|_1 = 1$ , alors  $y = A_\alpha z = \alpha Qz + \frac{1 - \| \alpha Qz \|_1}{N} e$  et  $y \geq 0$ .*

On a ainsi ramené le calcul du produit "matrice pleine-vecteur" original à un produit "matrice creuse-vecteur" et à une évaluation de norme, effectivement calculables à l'échelle du Web (à condition de disposer de ressources informatiques conséquentes, quand même).

3. *Programmer alors l'algorithme*

---

```
1 Choisir r0
2 Tant que s > tol, faire
3   r1=alpha Q r0
4   beta=1-norm(r1,1)
5   r2=r1+beta/N*e
6   s=norm(r2-r0,1)
7   r0=r2
8 retourner r0
```

---

Construire un exemple de réseau de pages webs de “grande taille”  $N = 1000$  (la taille réelle du WWW est de plusieurs milliards !...) et comparer la vitesse de cet algorithme avec l’algorithme de la puissance.

Ici aussi on pourra répondre aux trois questions suivantes :

- (a) Pour des petites valeurs de  $N$ , le classement est-il bien le même que celui donné par la méthode de la puissance ?
- (b) Y a-t-il une influence du choix de  $\alpha$  sur le classement final ?
- (c) Comment le nombre d’itérations varie avec  $\alpha$  ?