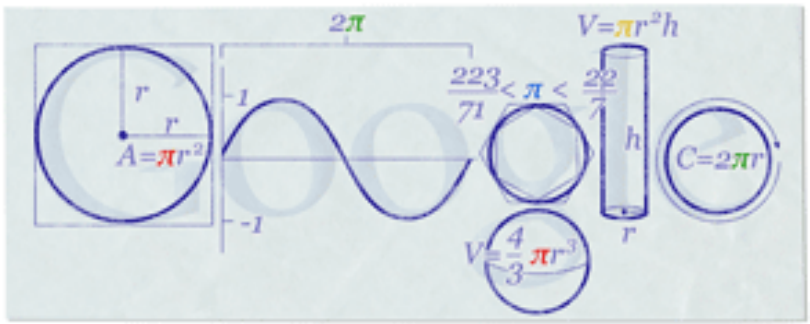
Les matrices et l'algorithme PageRank de Google

Olivier Guibé

LMRS -- Université de Rouen

Les mathématiques -- h2o -- 18 mars 2013



Google : une entreprise

à l'initiative de deux étudiants Sergey Brin et Larry Page

- fondée en 1998
- depuis 2000 vente de publicités
- cotation en bourse depuis 2004
- une recherche Google pour en savoir plus :)

Aujourd'hui

- nombre d'employés : plus 50 000 en 2011
- nombre de serveurs/PC : plus de 900 000 (2011)
- activité de recherche importante

Une des clefs de cette réussite

PageRank

ou un algorithme de tri des pages web qui est, la plupart du temps, pertinent.

Que fait un moteur de recherche ?

L'utilisateur lance une requête : mots clés cherchés.
Le moteur de recherche exécute

1. liste des pages contenant les mots clés
2. tri par ordre de pertinence
3. affichage des résultats

Il y a donc plusieurs aspects dont

- modélisation mathématique : comment définir/calculer la pertinence ? (personne n'est chargée de lire toutes les pages web !)
- ressource informatique : stockage, traitement d'une quantité énorme d'informations

Nous allons donner deux mauvais choix de calcul de pertinence et le PageRank.

Le WEB

Contenu hétérogène :

des pages sur tous les sujets existants, aucune centralisation, très peu de structure.

Des contributions multiples et variées qui changent tous les jours.

Point commun :

pages HTML, HyperText Markup Language, avec des **liens les reliant les unes aux autres**. Un fichier **HTML** est plus ou moins un fichier texte avec des balises, ce qui permet aux « robots » de lire, repérer les mots clefs.

Le point commun

L'idée est de travailler sur le fait qu'il y a des liens reliant les pages les unes aux autres. La première chose à faire :

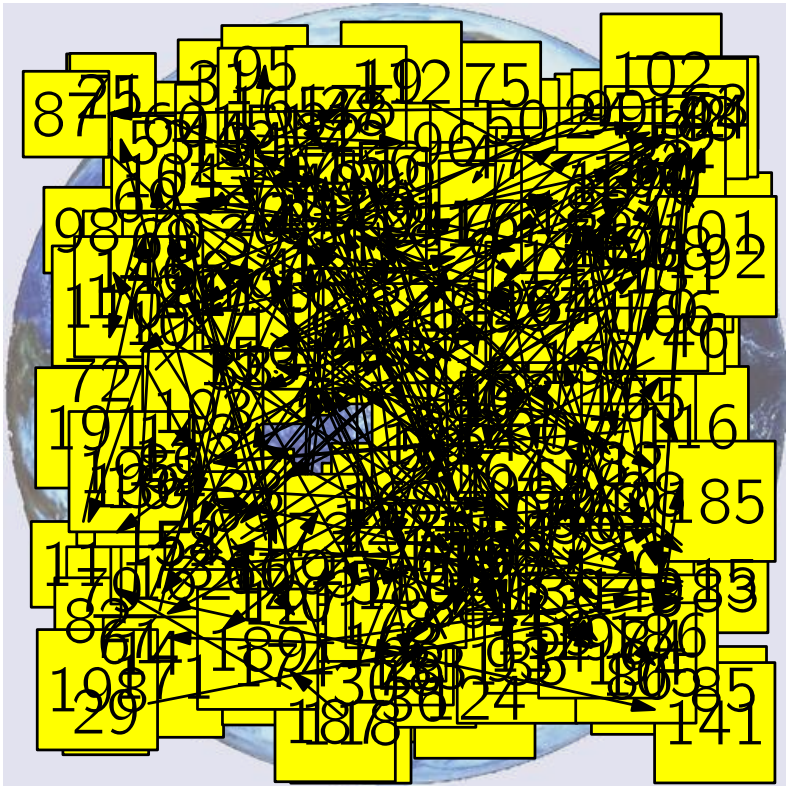
- numéroter toutes les pages : P_1, P_2, etc

Comme le but est de définir un critère « pertinence de la page » il faudra distinguer « P_1 contient un lien qui pointe sur P_2 » et « P_2 contient un lien qui pointe sur P_1 ».

- si la page P_j contient un lien vers la page P_i on matérialise cela par une flèche $P_j \longrightarrow P_i$.

Voici le résulat

200 pages web !



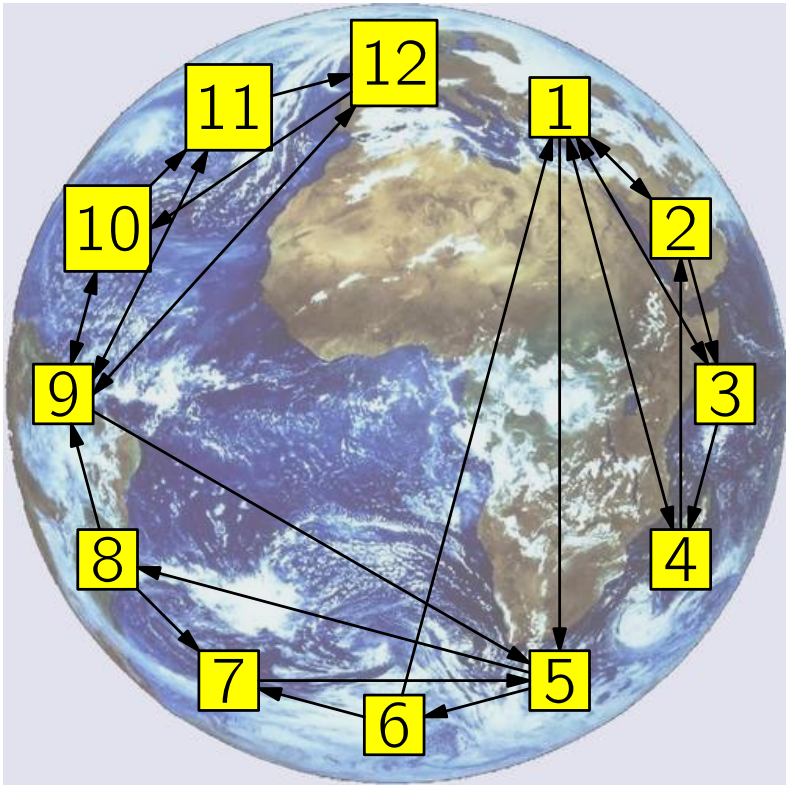
La réalité (ou presque)

Les chiffres sont difficiles à vérifier, il y a des pages « vides » ou générer automatiquement, sans contenu, etc.

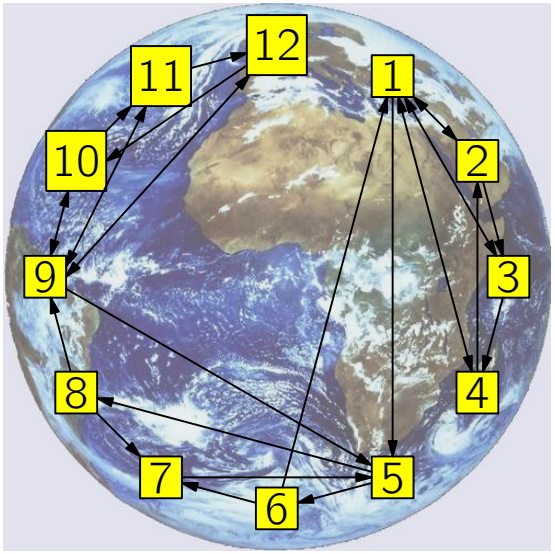
- plus de 600 millions de sites
- quelque mille milliards de pages
- il y a aussi des pages non indexées

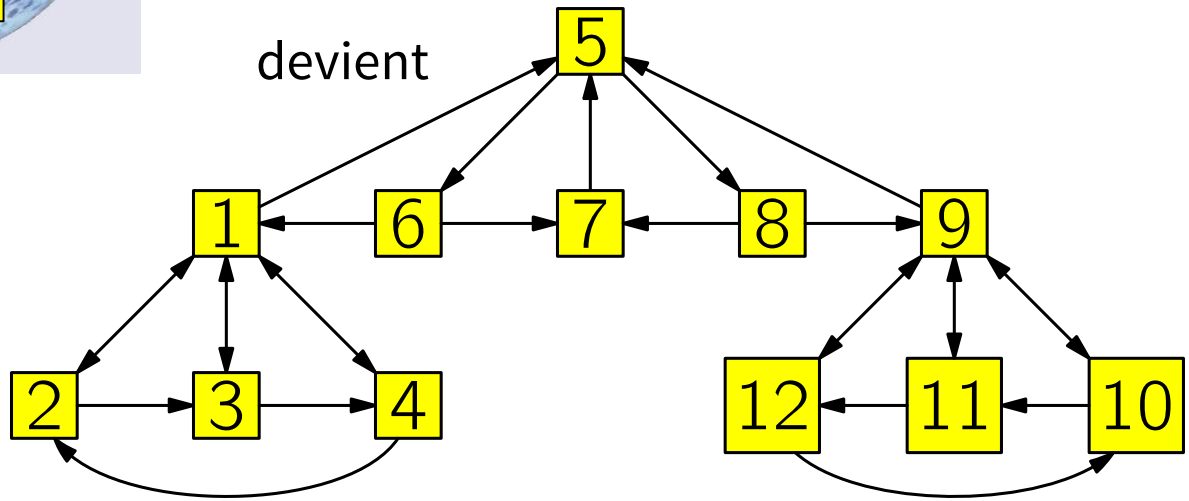
Le point commun

Traduisons en terme de graphe un Web à 12 pages !



Une autre disposition





Quelques explications

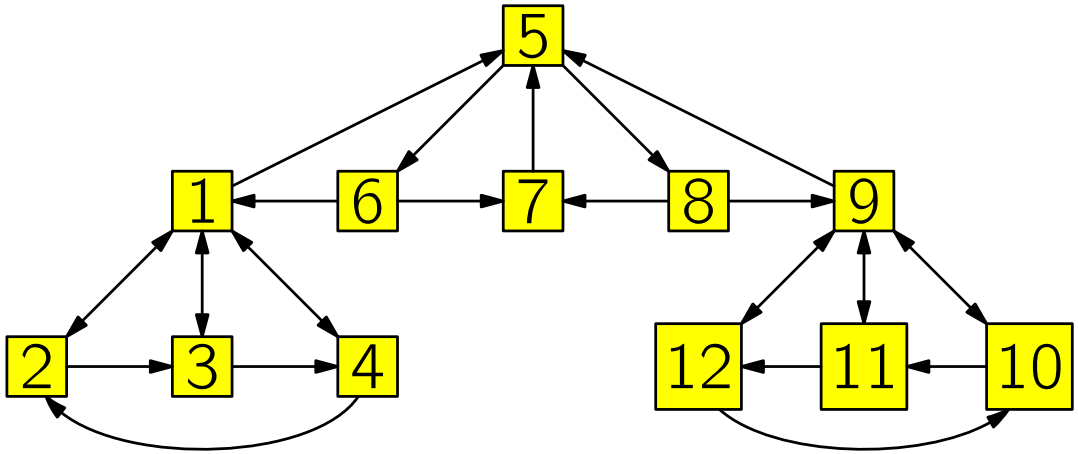
La structure de notre graphe est la suivante $1 \to 2,3,4,5$; $2 \to 1,3$; $3 \to 1,4$; $4 \to 1,2$; $5 \to 6,8$; $6 \to 1,7$; $7 \to 5$; $8 \to 7,9$; $9 \to 5,10,11,12$; $10 \to 9,11$; $11 \to 9,12$; $12 \to 9,10$.

Parmi P_1, P_2, P_3 et P_4, la page P_1 semble être une référence.

Parmi P_9, P_{10}, P_{11} et P_{12}, la page P_9 semble être une référence.

De la structure P_5, P_6, P_7 et P_8, P_7 est la plus citée mais avec un lien de P_7 sur P_5.

Finalement comme P_1 et P_9, reconnues comme importantes, pointent sur P_5, on propose P_5 comme page la plus pertinente.



Définir un score

Le but est de trouver une méthode, un algorithme qui calcule la pertinence de chaque page.

On peut représenter la pertinence par un nombre ou un score positif avec la convention que plus le score est grand plus la page est « importante ».

Évidemment on peut discuter très longtemps sur « qu'est qu'une page importante, pertinente ? ».

Le fait est que Google a semblé/semble encore répondre de façon satisfaisante aux requêtes des utilisateurs.

Idée naïve : compter les liens

Idée

Une page importante est une page qui reçoit beaucoup de liens

Le score de la page P_i est la somme du nombre de liens qui pointent sur P_i, ou encore le nombre de « votes ». Rappelons notre web :
$1 \to 2,3,4,5$; $2 \to 1,3$; $3 \to 1,4$; $4 \to 1,2$; $5 \to 6,8$;
$6 \to 1,7$; $7 \to 5$; $8 \to 7,9$; $9 \to 5,10,11,12$; $10 \to 9,11$;
$11 \to 9,12$; $12 \to 9,10$.
Calcul des scores :
$P_1 : 4$; $P_2 : 2$; $P_3 : 2$; $P_4 : 2$; $P_5 : 3$; $P_6 : 1$; $P_7 : 2$; $P_8 : 1$;
$P_9 : 4$; $P_{10} : 2$; $P_{11} : 2$; $P_{12} : 2$.

Les pages 1 et 9 arrivent en premier.

Idée naïve : bilan

Calcul de la pertinence par le nombre de « votes ».

Idée naïve : bilan

Calcul de la pertinence par le nombre de « votes ».

Avantage

Très simple à calculer

Idée naïve : bilan

Calcul de la pertinence par le nombre de « votes ».

Avantage

Très simple à calculer

Inconvénient

Si tout le monde est d'accord pour placer la page 5 en premier, cette méthode ne correspond pas à notre classement « ressenti ».

Idée naïve : bilan

Calcul de la pertinence par le nombre de « votes ».

Avantage

Très simple à calculer

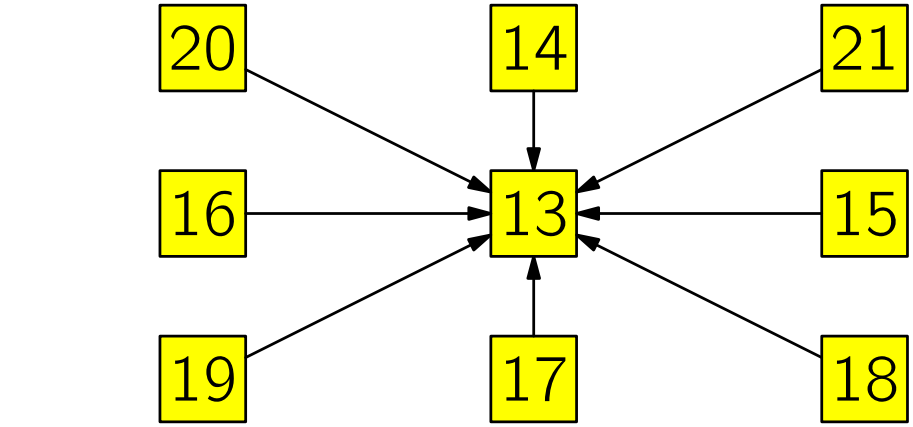
Inconvénient

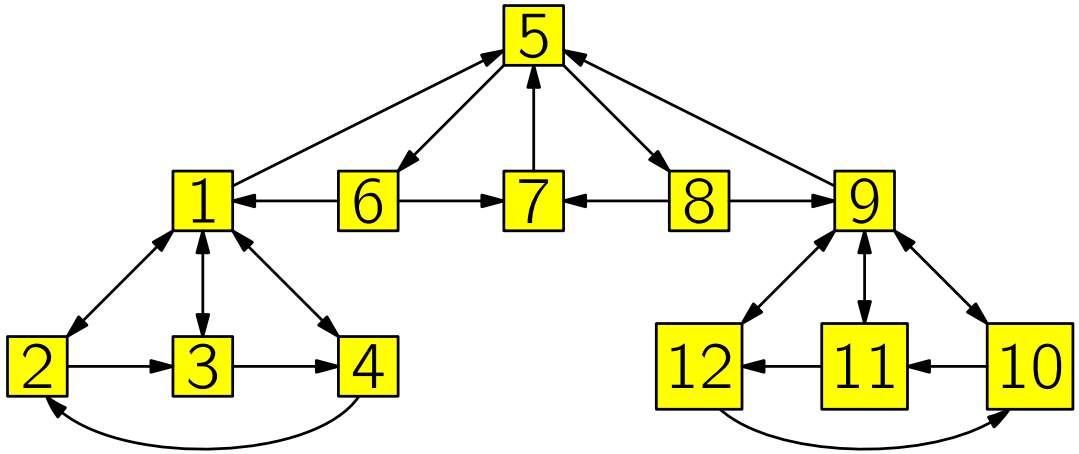
Si tout le monde est d'accord pour placer la page 5 en premier, cette méthode ne correspond pas à notre classement « ressenti ».

Inconvénient ++

Très facile de manipuler ; il suffit de créer des pages web qui pointent vers son site pour augmenter artificiellement son score !

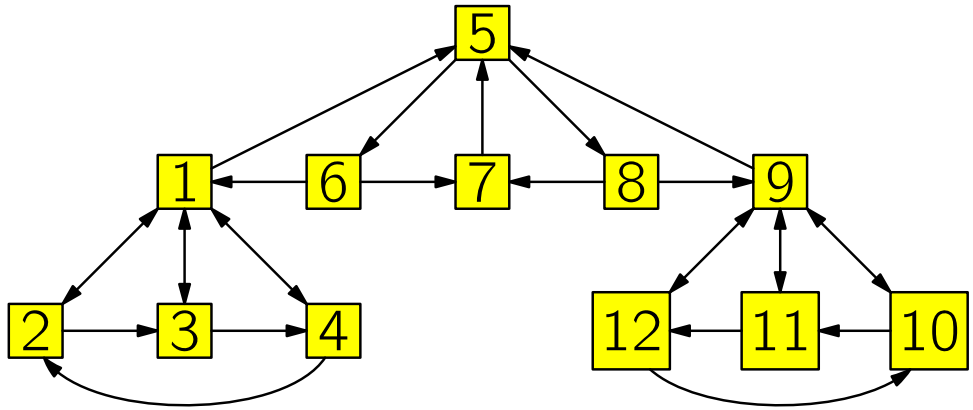
avec par exemple





Le gagnant est : page 13 !

nouvelle idée : pondération

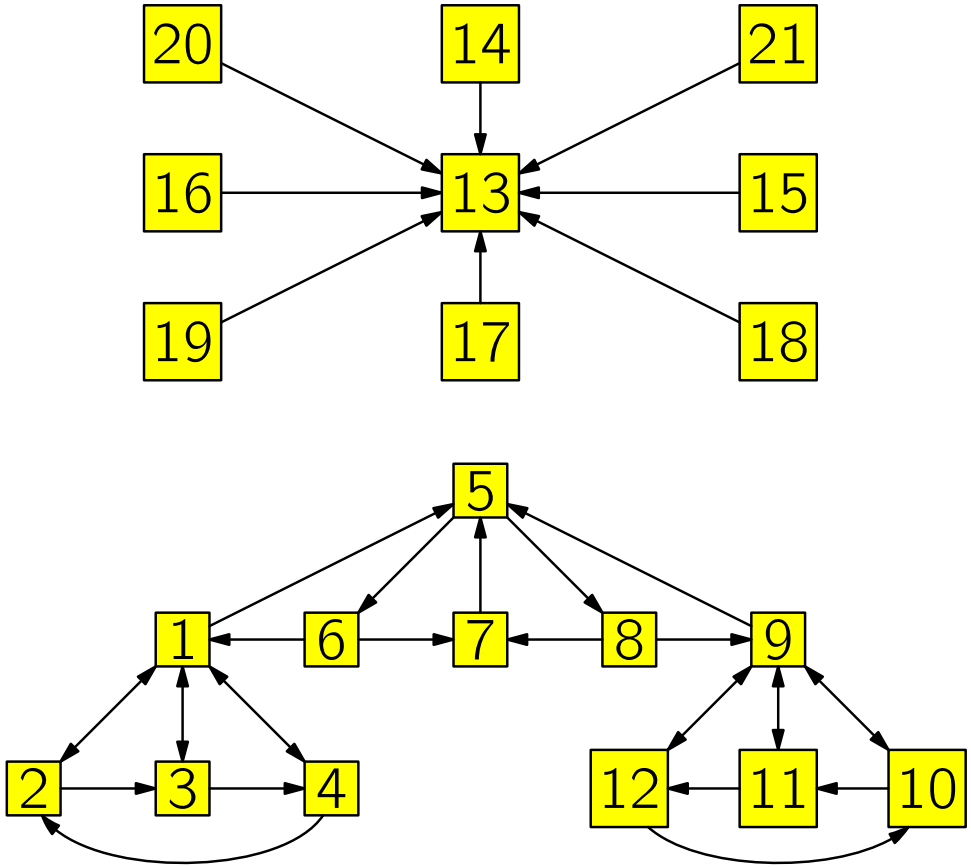


Pour minimiser l'impact des pages qui contiennent beaucoup de liens, on ajoute une pondération au « vote » : la page 2 contient 2 liens qui pointent vers les pages 1 et 3.
La page 2 « vote donc » pour la page page 1 pour 1/2.
Total

- Page 1 : $1/2 + 1/2 + 1/2 + 1/2 = 2$
- Page 9 : $1/2 + 1/2 + 1/2 + 1/2 = 2$
- Page 5 : $1/4 + 1 + 1/4 = 3/2$

Pondération : bilan

En fait c'est une méthode qui a les mêmes avantages et surtout les mêmes inconvénients (facile de manipuler un classement) :



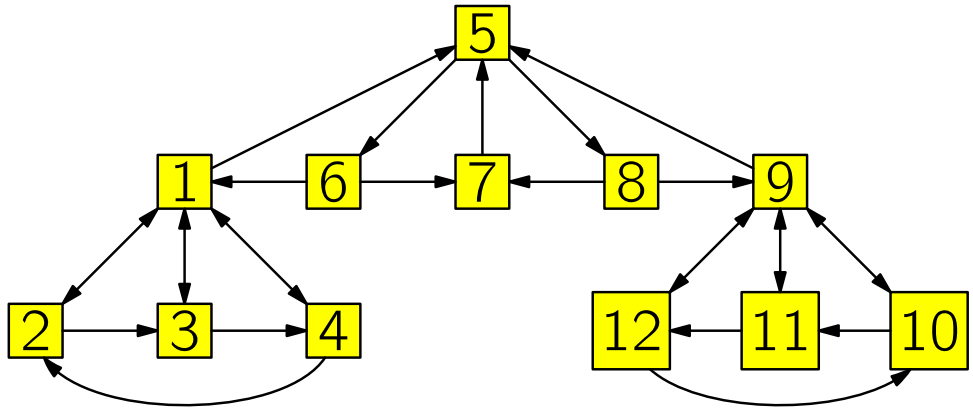
L'idée qui a fait ses preuves

« Une page est importante si beaucoup de pages importantes la citent »

Traduction

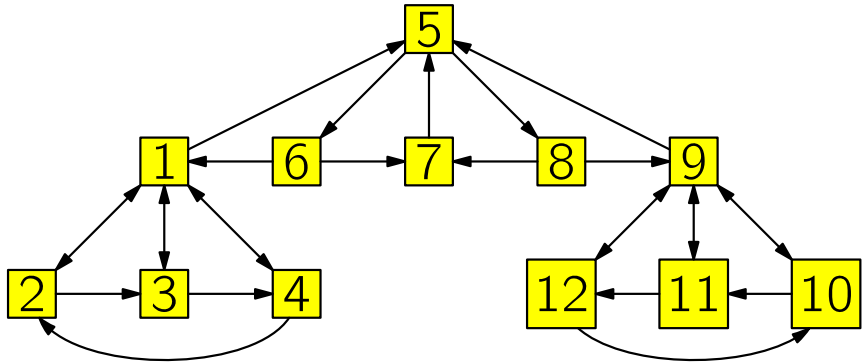
Le vote d'une page pour une autre dépend en plus de son score !

Désignons par s_1 le score de P_1, s_2 celui de P_3, etc.



$$s_1 = \frac{s_2}{2} + \frac{s_3}{2} + \frac{s_4}{2} + \frac{s_6}{2} = ??$$

Continuons



On obtient donc beaucoup d'équations !

$$s_1 = \frac{s_2}{2} + \frac{s_3}{2} + \frac{s_4}{2} + \frac{s_6}{2}$$

$$s_9 = \frac{s_{12}}{2} + \frac{s_{11}}{2} + \frac{s_{10}}{2} + \frac{s_8}{2}$$

$$s_5 = \frac{s_1}{4} + s_7 + \frac{s_9}{4}$$

$$\ldots$$

Math et matrices

On obtient (seulement) 12 équations

$$
\begin{aligned}
s_1 &= \frac{s_2}{2} + \frac{s_3}{2} + \frac{s_4}{2} + \frac{s_6}{2} \\
s_2 &= \frac{s_1}{4} + \frac{s_4}{2} \\
s_3 &= \frac{s_1}{4} + \frac{s_2}{2} \\
s_4 &= \frac{s_1}{4} + \frac{s_3}{2} \\
s_5 &= \frac{s_1}{4} + \frac{s_7}{1} + \frac{s_9}{4} \\
&\vdots \\
s_{12} &= \frac{s_9}{4} + \frac{s_{11}}{2}
\end{aligned}
$$

qui se mettent sous la forme $As = s$ avec A la matrice

$$\begin{pmatrix}
0 & 1/2 & 1/2 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1/4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 1/2 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0
\end{pmatrix}$$

et l'inconnue est s, le vecteur des scores, possédant douze coordonnées

$$s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{12} \end{pmatrix}$$

Matrice, vecteur

Définitions

Une matrice carrée est un « tableau » de n lignes et n colonnes contenant donc n^2 éléments.
Un vecteur est un « tableau » de n lignes et 1 colonne.

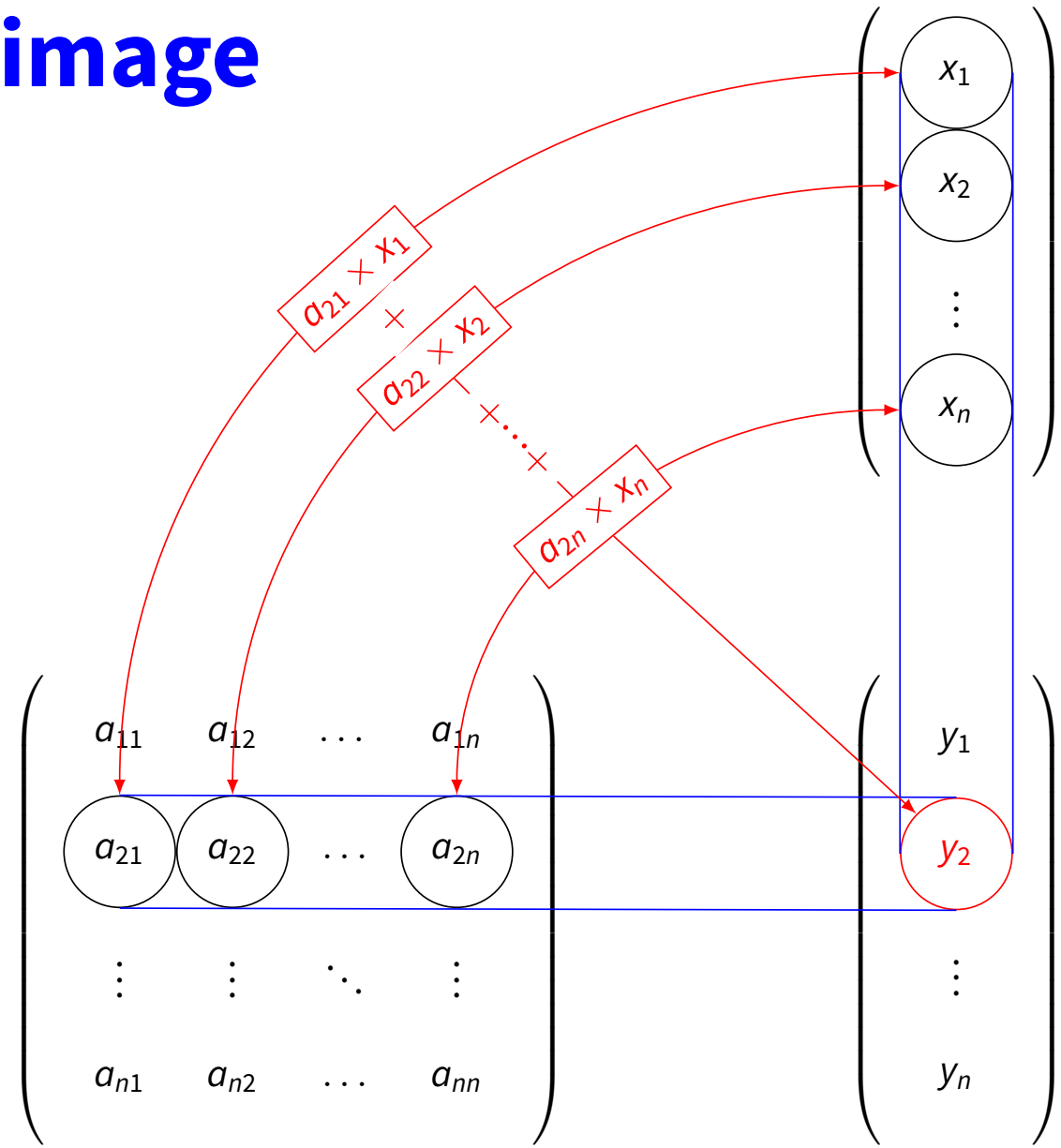
On peut (sous réserve de compatibilité des tailles)

- additionner les matrices (on additionne terme à terme)
- multiplier une matrice par un réel (on multiplie tous les termes de la matrice par ce réel)
- multiplier une matrice par un vecteur (et on obtient un vecteur)
- multiplier deux matrices

Les deux dernières opérations sont plus compliquées (ce n'est pas une multiplication terme à terme). Le produit matriciel est un bon exemple de « multiplication » non commutative : en général $A \times B \neq B \times A$.

En image

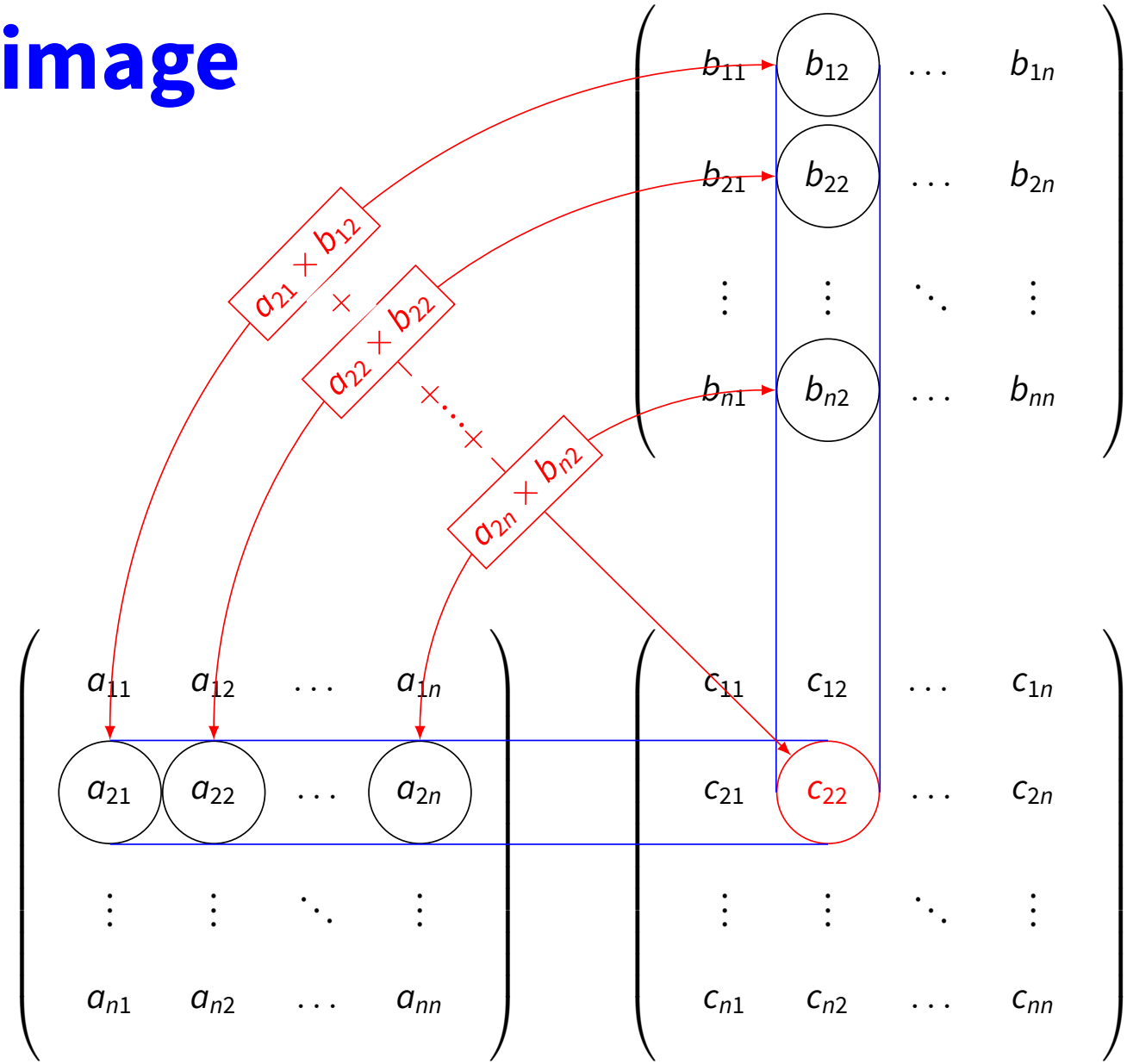
x : vecteur n lignes



A : n lignes n colonnes

$y = Ax$: vecteur n lignes

En image


B : n lignes n colonnes
b_{11} b_{12} ... b_{1n}
b_{21} b_{22} ... b_{2n}
b_{n1} b_{n2} ... b_{nn}
$a_{21} \times b_{12}$
+
$a_{22} \times b_{22}$
+ ... +
$a_{2n} \times b_{n2}$
a_{11} a_{12} ... a_{1n}
a_{21} a_{22} ... a_{2n}
a_{n1} a_{n2} ... a_{nn}
c_{11} c_{12} ... c_{1n}
c_{21} c_{22} ... c_{2n}
c_{n1} c_{n2} ... c_{nn}
A : n lignes n colonnes
C = A × B : n lignes n colonnes

Les matrices

Il y a aussi les matrices non carrées.
Les matrices sont utilisées dans de nombreuses applications des mathématiques :

- jeu 3D (moteur de rendu, matrices 4×4)
- météorologie, océanographie
- industrie automobile, spatiale, etc, où de nombreux codes de calcul (structure, fluide, combustion, mécanique) proviennent de la modélisation, des mathématiques
- cryptographie, code correcteur
- biologie (modèle de population), réaction chimique
- économie
- etc.

Résultats

La résolution (à l'aide d'un logiciel de calcul) donne :

- score de P_1, P_7 et P_9 : 2/12
- score de P_5 : 3/12
- les autres : 1/12.

Nouvelle idée : bilan

Avantage

Le modèle semble donner notre classement.

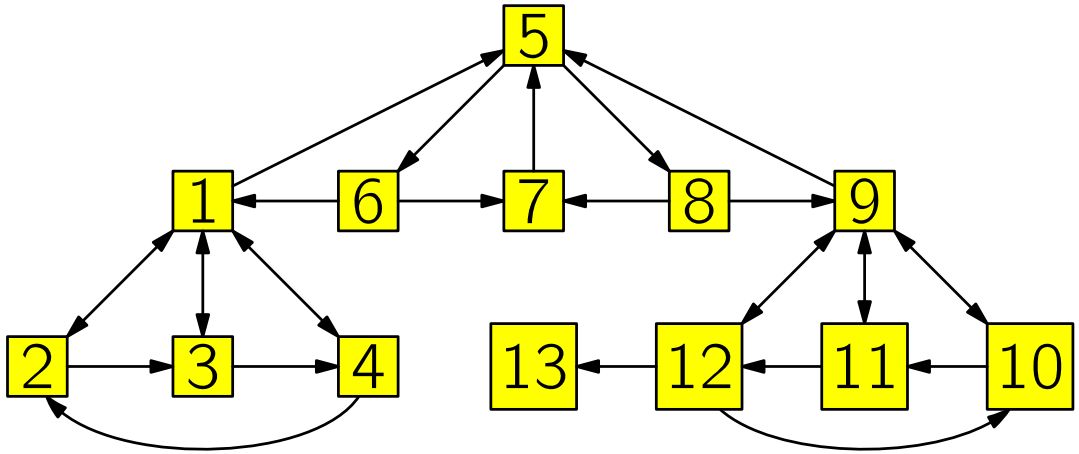
Inconvénient

À la main, c'est vite difficile (voire impossible) pour calculer ce classement.

Outils nécessaires

Nous aurons besoin de l'informatique et des mathématiques

Nouvelle idée : ne pas tomber dans un trou noir !

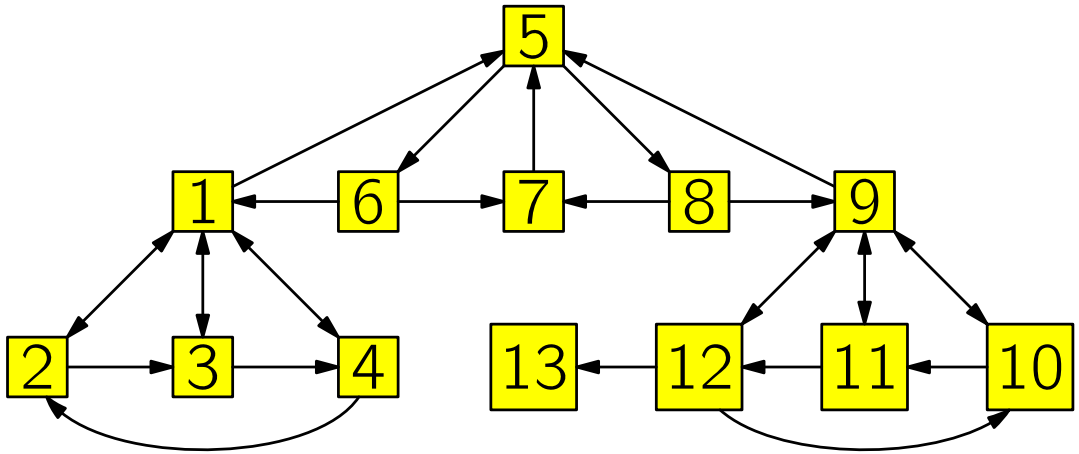


La page 13 est une page ne contenant aucun lien.
On peut vérifier

- score de la page 13 égal à 1
- les autres à zéro

est une solution !

Nouvelle idée : ne pas tomber dans un trou noir !



La page 13 est une page ne contenant aucun lien.

Interprétation

Le surfeur qui arrive à la page 13 ne peut plus en sortir puisque la page 13 n'a pas de lien externe.

Solution : ajouter du hasard

Le modèle traduit un comportement très prévisible du surfeur. Il clique sur les liens d'une page pour aller sur une autre.

On peut considérer que notre comportement serait plutôt : au bout de quelques pages visitées je vais aller sur une page qui n'a rien à voir ou très peu avec les précédentes.

Traduction

On autorise un saut vers une page aléatoire (n'importe quelle page du web) avec une faible probabilité.

Traduction mathématique

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

Le mathématicien est content !

Il existe un unique vecteur r (de norme 1) vérifiant $Br = r$. (théorème de Perron-Frobenius)

Score

- Pages 1 et 9 : 0.1289
- Pages 5 : 0.1255
- Page 7 : 0.0658
- Les autres : 0.0694

On peut multiplier par 12 de façon à y voir plus clair.

Calcul : ce qu'il ne faut pas faire

La matrice étant de très grande taille, il y a des obstacles :

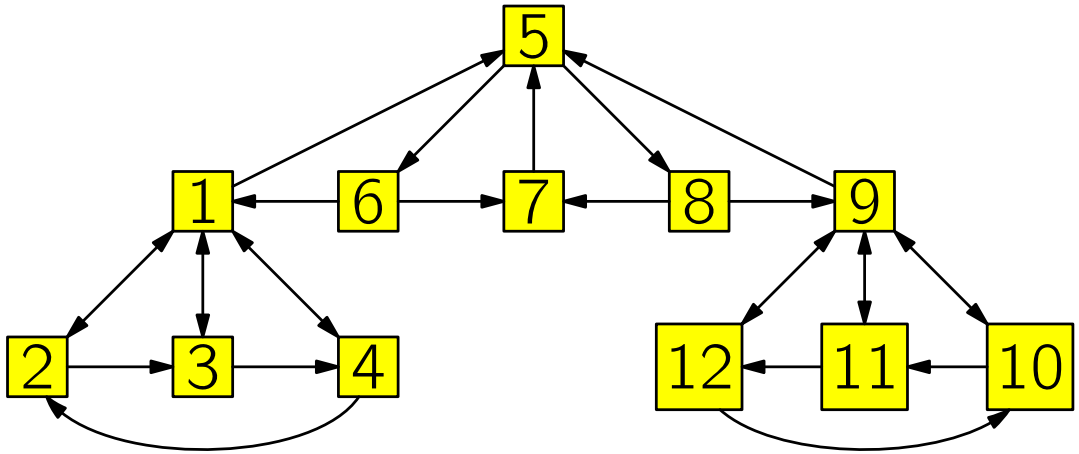
- stockage de la matrice
- temps de calcul : il ne faut pas attendre 10 jours (ou plus) pour une recherche

Des chiffres

Résoudre un système linéaire (ce qui est moins coûteux et moins compliqué que notre problème) pour une matrice de taille n nécessite de l'ordre de n^3 opérations. Pour une matrice de taille 10 millions (ce qui est moins que toutes les pages web) et avec une puissance de 1pétaFLOPS (10^{15} opérations par seconde) il faudrait de l'ordre 10^6 secondes soit 11 jours.

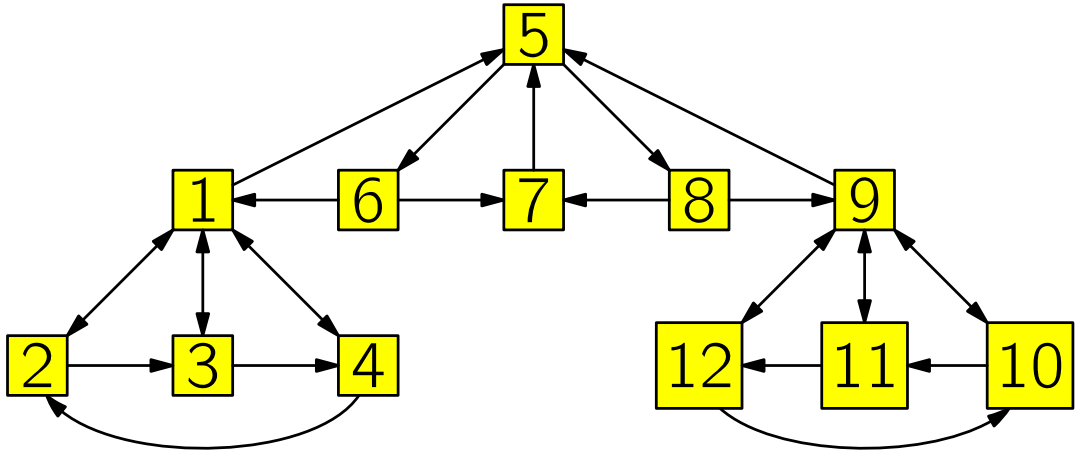
Comment faire ?

Plutôt que de chercher « la solution » on peut chercher à l'approcher, c'est moins coûteux et cela est en général suffisant.
Observons sous l'oeil des probabilités et acceptons de découper le surfeur en petits morceaux via une marché aléatoire sur le graphe des pages web



1. Temps 0 : surfeur sur la page 1 : probabilité 1
2. Temps 1 : surfeur sur la page 2, 3, 4, 5 avec la probabilité $0.85/4 + .15/12$, sur les autres pages avec la probabilité $.15/12$

Explications



1. Temps 0 : surfeur sur la page 1 ; probabilité 1

On mélange les deux effets

- la page 1 contient 4 liens (pages 2, 3, 4 et 5) : même probabilité d'aller sur une des 4 pages
- le hasard : de la page 1 on peut aller sur une autre page avec une faible probabilité $(1-\alpha)/N$ (avec $\alpha = .85$)

Itérons le processus

n	1	2	3	4	5	6	7	8	9	10	11	12
0	1	0	0	0	0	0	0	0	0	0	0	0
1	.0125	0.225	0.225	0.225	0.225	.0125	.0125	.0125	.0125	.0125	.0125	.0125
2	.3047	.1108	.1108	.1108	.0284	.1 081	.0231	.1081	.0338	.0205	.0205	.0205
3	.1997	.1243	.1243	.1243	.1041	.0 246	.1044	.0246	.0846	.0284	.0284	.0284
⋮												
10	.1392	.0774	.0774	.0774	.1309	.0 655	.0640	.0655	.1150	.0627	.0627	.0627
20	.1299	.0699	.0699	.0699	.1252	.0659	.0684	.0659	.1280	.0688	.0688	.0688
30	.1290	.0694	.0694	.0694	.1255	.0658	.0685	.0658	.1289	.0694	.0694	.0694
50	.1290	.0694	.0694	.0694	.1255	.0658	.0685	.0658	.1290	.0694	.0694	.0694

Observations

Convergence vers le score !

Explication : point de vue probabiliste

Thème : chaîne de Markov (discrète).
Le vecteur de taille 12 se voit comme la probabilité

« le surfeur est sur la page i »

x_1 proba d'être sur page 1, x_2 proba d'être sur page 2, etc. Comme c'est une probabilité, les x_i sont positifs et leur somme vaut 1.

Sous certaines conditions, l'itération du processus « converge »

vers un état d'équilibre, une « mesure invariante par le processus markovien »

Cet état d'équilibre correspond au score cherché.

Rappel sur les matrices

Notre matrice A

$$\begin{pmatrix}
0 & 1/2 & 1/2 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/4 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1/4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 1/2 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 & 0 & 1/2 & 0
\end{pmatrix}$$

et avec le hasard

$$B = \alpha A + \frac{1-\alpha}{N} J$$

Explication : point de vue matricielle/analyste

Si x est le vecteur de départ (mis sous forme de ligne) :

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

Les itérations correspondent à calculer

$$Bx, BBx, BBBx, \text{ etc}$$

soit

$$Bx, B^2x, B^3x, B^4x, \ldots, B^{10}x, B^{20}x, \ldots$$

Sous certaines conditions, il y a « convergence » de $B^n x$ vers le score r ($Br = r$).

Itérations

Remplacer la recherche exacte par une suite ou un processus itératif est courant en mathématique :

- meilleure stabilité numérique
- recherche exacte trop longue, trop coûteuse
- à condition qu'il soit plus rapide de calculer $B^{30}x$.

exemple simple à tester sur votre machine à calculer

$\sqrt{2}$ peut être approché par la suite $u_0 = 1, u_{n+1} = u_n/2 + 1/u_n$.

Stockage

Une matrice est un tableau. Une première idée de stockage

- on stocke tout
- besoin : $n \times n$ « cases » mémoire

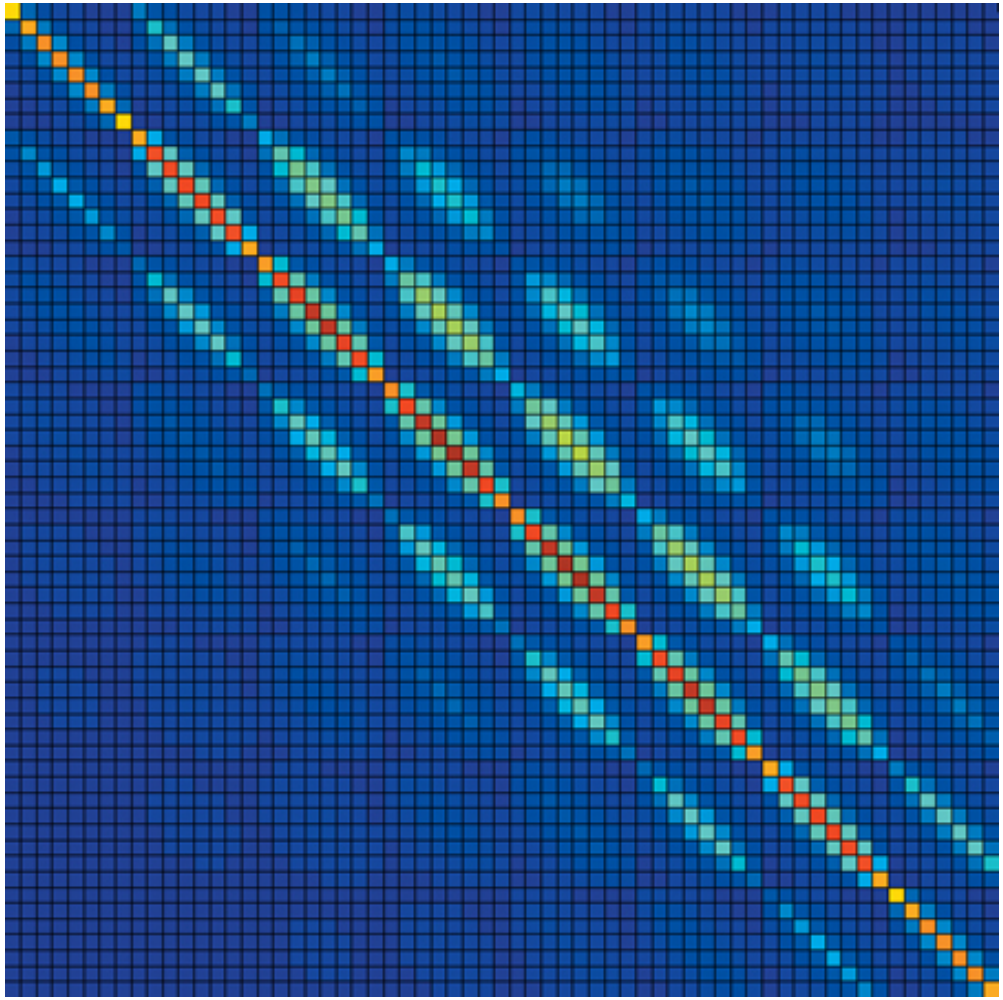
Pour notre cas, ce n'est pas optimal, c'est même très gourmand car

- il y a énormément de zéro dans la matrice relatif aux liens
- dans notre cas : 25 valeurs non nulles pour 144 valeurs !

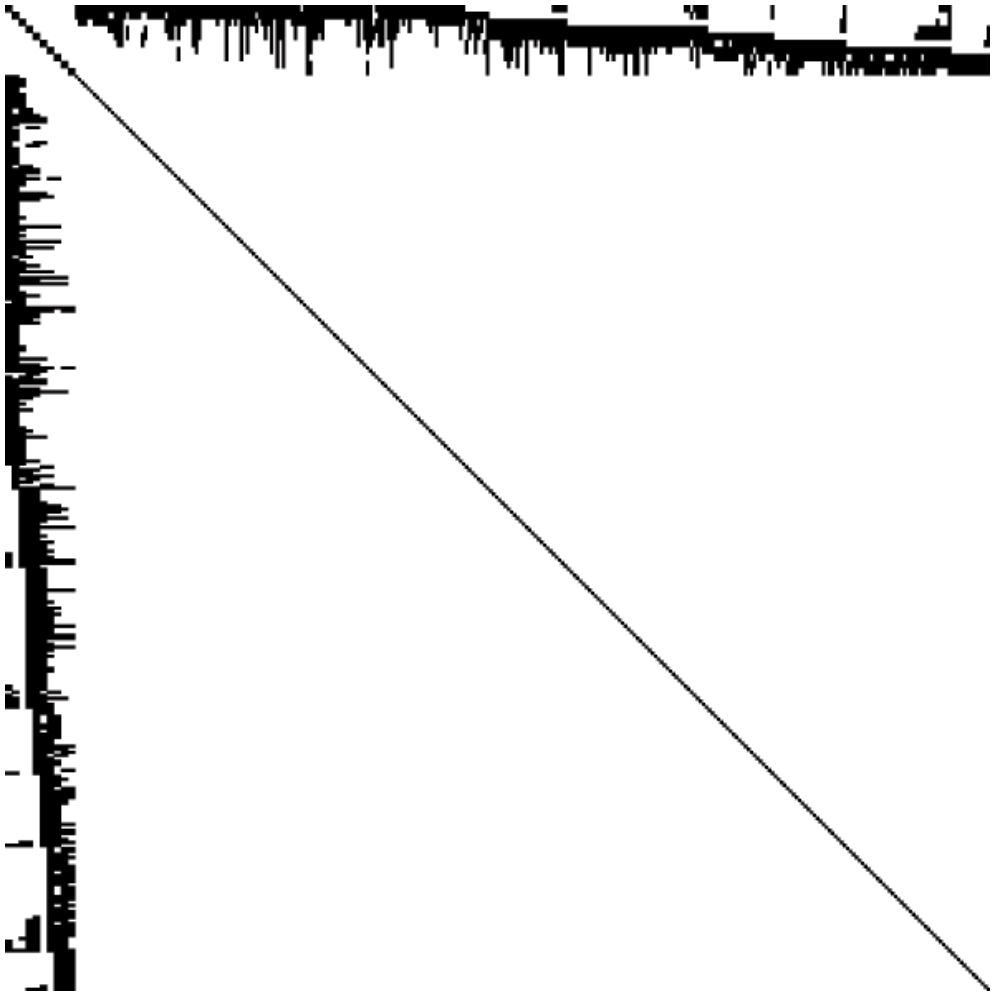
La matrice relatif aux liens est dite creuse. D'où un stocke particulier

- on stocke (i, j) valeur non nulle à ligne i/colonne j
- pour notre cas si la page P_i contient un lien vers la page P_j avec la pondération de vote

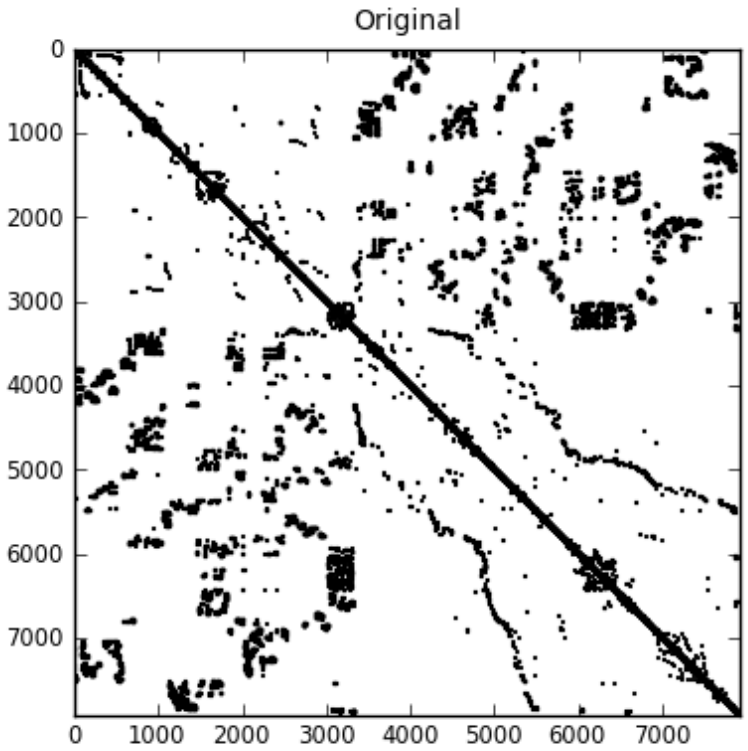
Matrice creuse en image

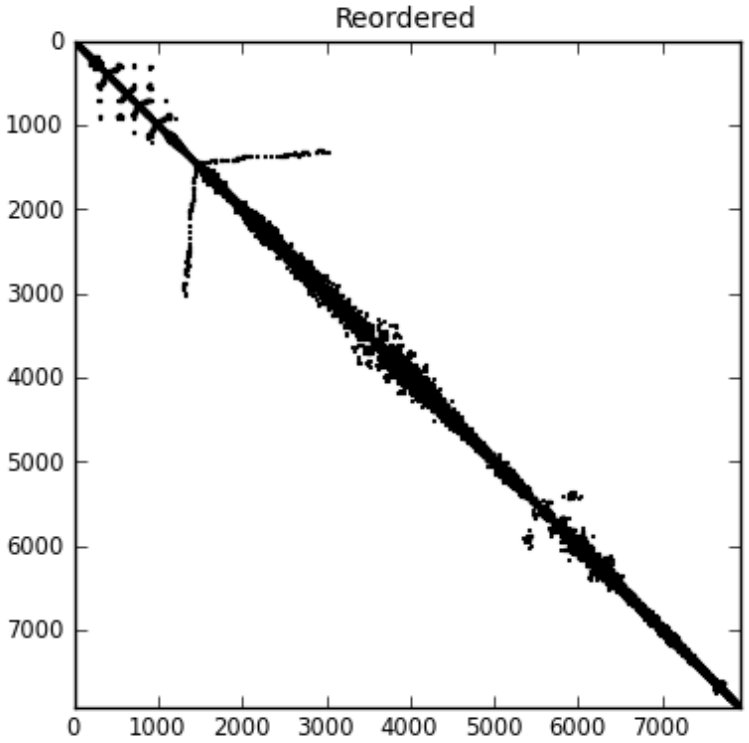


Matrice creuse en image



Matrice creuse en image





Optimisations

La structure creuse de A (même si B n'est pas creuse) et la méthode itérative (beaucoup d'astuces) font que le calcul du PageRank est très raisonnable en temps et en coût mémoire. Depuis l'algorithme a été amélioré

- minimisation de « Google bombing »
- plus de pertinence
- des choses publiées
- des choses secrètes

Influence de α

Comment est choisi le paramètre α. C'est un compromis entre

- grande valeur de α : processus converge lentement
- petite valeur de α : processus converge très rapidement
- petite valeur de α : liens comptent moins que le hasard
- grande valeur de α : liens comptent plus que le hasard

Ce que vous avez évité

Cette « nouvelle idée « (classement récursif) associée au côté « aléatoire » est le modèle PageRank :

- tout se passe bien
- ce classement est robuste (ajouter artificiellement des pages qui pointent vers une même page ne change pas beaucoup le classement)
- on utilise les matrices, l'algèbre linéaire, les marches aléatoires sur les graphes, le théorème du point fixe, etc
- du point de vue informatique comment gérer une telle quantité de données, les algorithme de calcul (certains sont secrets), etc

Merci