L'ALGORITHME PAGERANK DE GOOGLE: UNE PROMENADE SUR LA TOILE

MICHAEL EISERMANN

Depuis plus d'une décennie Google domine le marché des moteurs de recherche sur internet. Son point fort est qu'il trie intelligemment ses résultats par ordre de pertinence. Comment est-ce possible ? Depuis sa conception en 1998, Google continue à évoluer et la plupart des améliorations demeurent des secrets bien gardés. L'idée principale, par contre, a été publiée [1] : le pilier de son succès est une judicieuse modélisation mathématique.

QUE FAIT UN MOTEUR DE RECHERCHE ?

Une base de données a une structure prédéfinie qui permet d'en extraire des informations, par exemple « nom, rue, code postal, téléphone, … ». L'internet, par contre, est peu structuré : c'est une immense collection de textes de nature variée. Toute tentative de classification semble vouée à l'échec, d'autant plus que le web évolue rapidement : une multitude d'auteurs ajoutent constamment de nouvelles pages et modifient les pages existantes.

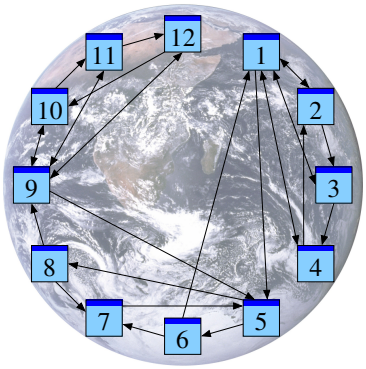
Pour trouver une information dans ce tas amorphe, l'utilisateur pourra lancer une recherche de mots-clés. Ceci nécessite une certaine préparation pour être efficace : le moteur de recherche copie préalablement les pages web en mémoire locale et trie les mots par ordre alphabétique. Le résultat est un annuaire de mots-clés avec leurs pages web associées.

Pour un mot-clé donné il y a typiquement des milliers de pages correspondantes (plus d'un million pour « tangente », par exemple). Comment aider l'utilisateur à repérer les résultats potentiellement intéressants ? C'est ici que Google a apporté sa grande innovation.

LE WEB EST UN GRAPHE !

Profitons du peu de structure qui soit disponible. L'internet n'est pas une collection de textes indépendants mais un immense *hypertexte* : les pages se citent mutuellement.

Afin d'analyser cette structure nous allons négliger le contenu des pages et ne tenir compte que des liens entre elles. Ce que nous obtenons est la structure d'un graphe. La figure suivante montre un exemple en miniature.



Dans la suite je note les pages web par $P_1, P_2, P_3, \ldots, P_n$ et j'écris $j \to i$ si la page P_j cite la page P_i. Dans notre graphe nous avons un lien $1 \to 5$, par exemple, mais pas de lien $5 \to 1$.

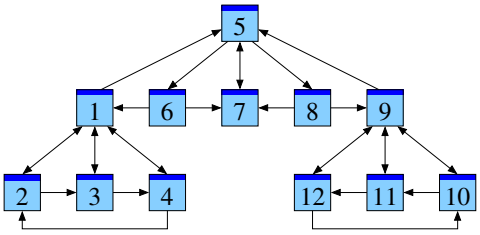
COMMENT EXPLOITER CE GRAPHE ?

Les liens sur internet ne sont pas aléatoires mais ont été édités avec soin. Quels renseignements pourrait nous donner ce graphe ?

L'idée de base, encore à formaliser, est qu'un lien $j \to i$ est une recommandation de la page P_j d'aller lire la page P_i. C'est ainsi un vote de P_j en faveur de l'autorité de la page P_i.

Date: première version juin 2009. *Dernière mise à jour:* 17 juillet 2009.
Document en support de mon exposé aux Journées APMEP à Rouen en octobre 2009.

Analysons notre exemple sous cet aspect. La présentation suivante de notre graphe suggère une hiérarchie possible — encore à justifier.



Parmi les pages P_1, P_2, P_3, P_4 la page P_1 sert de référence commune et semble un bon point de départ pour chercher des informations. Il en est de même dans le groupe $P_9, P_{10}, P_{11}, P_{12}$ où la page P_9 sert de référence commune. La structure du groupe P_5, P_6, P_7, P_8 est similaire, où P_7 est la plus citée. À noter toutefois que les pages P_1 et P_9, déjà reconnues comme importantes, font référence à la page P_5. On pourrait ainsi soupçonner que la page P_5 contient de l'information essentielle pour l'ensemble, qu'elle est la plus pertinente.

Premier modèle : comptage naïf

Il est plausible qu'une page importante reçoit beaucoup de liens. Avec un peu de naïveté, on croira aussi l'affirmation réciproque : si une page reçoit beaucoup de liens, alors elle est importante. Ainsi on pourrait définir l'importance μ_i de la page P_i comme le nombre des liens $j \to i$. En formule ceci s'écrit comme suit :

$$\mu_i := \sum_{j \to i} 1. \tag{1}$$

Autrement dit, μ_i est égal au nombre de « votes » pour la page P_i, où chaque vote contribue par la même valeur 1. C'est facile à définir et à calculer, mais ne correspond souvent pas à l'importance ressentie par l'utilisateur : dans notre exemple on trouve $\mu_1 = \mu_9 = 4$ devant $\mu_5 = \mu_7 = 3$. Ce qui est pire, ce comptage naïf est trop facile à manipuler en ajoutant des pages sans intérêt recommandant une page quelconque.

Second modèle : comptage pondéré

Certaines pages émettent beaucoup de liens : ceux-ci semblent moins spécifiques et leur poids sera plus faible. Nous partageons donc le vote de la page P_j en ℓ_j parts égales, où ℓ_j dénote le nombre de liens émis. Ainsi on pourrait définir une mesure plus fine :

$$\mu_i := \sum_{j \to i} \frac{1}{\ell_j}. \tag{2}$$

Autrement dit, μ_i compte le nombre de « votes pondérés » pour la page P_i. C'est facile à définir et à calculer, mais ne correspond toujours pas bien à l'importance ressentie : dans notre exemple on trouve $\mu_1 = \mu_9 = 2$ devant $\mu_5 = 3/2$ et $\mu_7 = 4/3$. Et comme avant ce comptage est trop facile à truquer.

Troisième modèle : comptage récursif

Heuristiquement, une page P_i paraît importante si beaucoup de pages importantes la citent. Ceci nous mène à définir l'importance μ_i de manière récursive comme suit :

$$\mu_i = \sum_{j \to i} \frac{1}{\ell_j} \mu_j. \tag{3}$$

Ici le poids du vote $j \to i$ est proportionnel au poids μ_j de la page émettrice. C'est facile à formuler mais moins évident à calculer. (Une méthode efficace sera expliquée dans la suite.) Pour vous rassurer vous pouvez déjà vérifier que notre exemple admet bien la solution

$$\begin{array}{c} P_1\ P_2\ P_3\ P_4\ P_5\ P_6\ P_7\ P_8\ P_9\ P_{10}\ P_{11}\ P_{12} \\ \mu = (\ 2,\ 1,\ 1,\ 1,\ 3,\ 1,\ 2,\ 1,\ 2,\ 1,\ 1,\ 1\ ). \end{array}$$

Contrairement aux modèles précédents, la page P_5 est repérée comme la plus importante. C'est bon signe, nous sommes sur la bonne piste…

Remarquons que (3) est un système de n équations linéaires à n inconnues. Dans notre exemple, où $n = 12$, il est déjà pénible à résoudre à la main, mais encore facile sur ordinateur. Pour les graphes beaucoup plus grands nous aurons besoin de méthodes spécialisées.

PROMENADE ALÉATOIRE

Avant de tenter de résoudre l'équation (3), essayons d'en développer une intuition. Pour ceci imaginons un surfeur aléatoire qui se balade sur internet en cliquant sur les liens au hasard. Comment évolue sa position ?

À titre d'exemple, supposons que notre surfeur démarre au temps $t = 0$ sur la page P_7. Le seul lien pointe vers P_5, donc au temps $t = 1$ le surfeur s'y retrouve avec probabilité 1. D'ici partent trois liens, donc au temps $t = 2$ il se trouve sur une des pages P_6, P_7, P_8 avec probabilité $1/3$. Voici les probabilités suivantes :

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
t=0	.000	.000	.000	.000	.000	.000	1.00	.000	.000	.000	.000	.000
t=1	.000	.000	.000	.000	1.00	.000	.000	.000	.000	.000	.000	.000
t=2	.000	.000	.000	.000	.000	.333	.333	.333	.000	.000	.000	.000
t=3	.167	.000	.000	.000	.333	.000	.333	.000	.167	.000	.000	.000
t=4	.000	.042	.042	.042	.417	.111	.111	.111	.000	.042	.042	.042
t=5	.118	.021	.021	.021	.111	.139	.250	.139	.118	.021	.021	.021
...												
t=29	.117	.059	.059	.059	.177	.059	.117	.059	.117	.059	.059	.059
t=30	.117	.059	.059	.059	.177	.059	.117	.059	.117	.059	.059	.059

On observe une diffusion qui converge assez rapidement vers une distribution stationnaire. Vérifions cette observation par un second exemple, partant cette fois-ci de la page P_1 :

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
t=0	1.00	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
t=1	.000	.250	.250	.250	.250	.000	.000	.000	.000	.000	.000	.000
t=2	.375	.125	.125	.125	.000	.083	.083	.083	.000	.000	.000	.000
t=3	.229	.156	.156	.156	.177	.000	.083	.000	.042	.000	.000	.000
t=4	.234	.135	.135	.135	.151	.059	.059	.059	.000	.010	.010	.010
t=5	.233	.126	.126	.126	.118	.050	.109	.050	.045	.005	.005	.005
...												
t=69	.117	.059	.059	.059	.177	.059	.117	.059	.117	.059	.059	.059
t=70	.117	.059	.059	.059	.177	.059	.117	.059	.117	.059	.059	.059

Bien que la diffusion mette plus de temps, la mesure stationnaire est la même ! Elle coïncide d'ailleurs avec notre solution $\mu = (2,1,1,1,3,1,2,1,2,1,1,1)$, ici divisée par 17 pour normaliser la somme à 1. Les pages où μ_i est grand sont les plus « fréquentées » ou les plus « populaires ». Dans la quête de classer les pages web, c'est encore un argument pour utiliser la mesure μ comme indicateur.

LA LOI DE TRANSITION

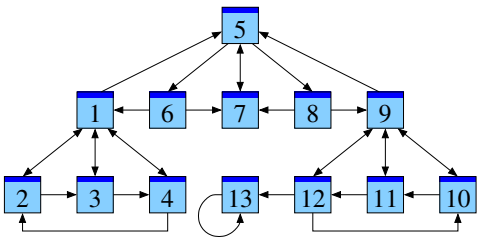
Comment formaliser la diffusion illustrée ci-dessus ? Supposons qu'au temps t notre surfeur aléatoire se trouve sur la page P_j avec une probabilité p_j. La probabilité de partir de P_j et de suivre le lien $j \to i$ est alors $\frac{1}{\ell_j} p_j$. La probabilité d'arriver au temps $t+1$ sur la page P_i est donc

$$p'_i := \sum_{j \to i} \frac{1}{\ell_j} p_j. \tag{4}$$

Étant donnée la distribution initiale p, la loi de transition (4) définit la distribution suivante $p' = T(p)$. C'est ainsi que l'on obtient la ligne $t+1$ à partir de la ligne t dans nos exemples. (En théorie des probabilités ceci s'appelle une *chaîne de Markov*.) La mesure stationnaire est caractérisée par l'équation d'équilibre $\mu = T(\mu)$, qui est justement notre équation (3).

ATTENTION AUX TROUS NOIRS

Que se passe-t-il quand notre graphe contient une page (ou un groupe de pages) sans issue ? Pour illustration, voici notre graphe modifié :



L'interprétation comme marche aléatoire permet de résoudre l'équation (3) sans aucun calcul : la page P_{13} absorbe toute la probabilité car notre surfeur aléatoire tombera tôt ou tard sur cette page, où il demeure pour le reste de sa vie. Ainsi la solution est $\mu = (0,0,0,0,0,0,0,0,0,0,0,0,1)$. Notre modèle n'est donc pas encore satisfaisant.

LE MODÈLE UTILISÉ PAR GOOGLE

Pour échapper aux trous noirs, Google utilise un modèle plus raffiné : avec une probabilité fixée c le surfeur abandonne sa page actuelle P_j et recommence sur une des n pages du web, choisie de manière équiprobable ; sinon, avec probabilité $1-c$, le surfeur suit un des liens de la page P_j, choisi de manière équiprobable.

Cette astuce de « téléportation » évite de se faire piéger par une page sans issue, et garantit d'arriver n'importe où dans le graphe, indépendamment des questions de connexité.

Dans ce modèle la transition est donnée par

$$p'_i := \frac{c}{n} + \sum_{j \to i} \frac{1-c}{\ell_j} p_j. \tag{5}$$

Le premier terme $\frac{c}{n}$ provient de la téléportation, le second terme est la marche aléatoire précédente. La mesure d'équilibre vérifie donc

$$\mu_i = \frac{c}{n} + \sum_{j \to i} \frac{1-c}{\ell_j} \mu_j. \tag{6}$$

Le paramètre c est encore à calibrer. Pour $c = 0$ nous obtenons le modèle précédent. Pour $0 < c \leq 1$ la valeur $1/c$ est le nombre moyen de pages visitées, c'est-à-dire le nombre de liens suivis plus un, avant de recommencer sur une page aléatoire (processus de Bernoulli).

Par exemple, le choix $c = 0.15$ correspond à suivre environ 6 liens en moyenne, ce qui semble une description réaliste.

Pour conclure l'analyse de notre exemple, voici la marche aléatoire partant de la page P_1 :

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
t=0	1.00	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
t=1	.013	.225	.225	.225	.225	.013	.013	.013	.013	.013	.013	.013
t=2	.305	.111	.111	.111	.028	.076	.087	.076	.034	.020	.020	.020
t=3	.186	.124	.124	.124	.158	.021	.085	.021	.071	.028	.028	.028
t=4	.180	.105	.105	.105	.140	.057	.075	.057	.057	.040	.040	.040
t=5	.171	.095	.095	.095	.126	.052	.101	.052	.087	.042	.042	.042
⋮												
t=29	.120	.066	.066	.066	.150	.055	.102	.055	.120	.066	.066	.066
t=30	.120	.066	.066	.066	.150	.055	.102	.055	.120	.066	.066	.066

La mesure stationnaire est vite atteinte, et la page P_5 arrive en tête avec $\mu_5 = 0.15$ avant les pages P_1 et P_9 avec $\mu_1 = \mu_9 = 0.12$.

LE THÉORÈME DU POINT FIXE

Afin de développer un modèle prometteur nous avons utilisé des arguments heuristiques et des illustrations expérimentales. Fixons maintenant ce modèle et posons-le sur un solide fondement théorique. Nos calculs aboutissent bel et bien dans notre exemple miniature, mais est-ce toujours le cas ? Le beau résultat suivant y répond en toute généralité :

Théorème du point fixe. *Considérons un graphe fini quelconque et fixons le paramètre c tel que $0 < c \leq 1$. Alors l'équation* (6) *admet une unique solution vérifiant $\mu_1 + \cdots + \mu_n = 1$. Dans cette solution $\mu_1, \ldots, \mu_n$ sont tous positifs. Pour toute distribution de probabilité initiale le processus de diffusion* (5) *converge vers cette unique mesure stationnaire μ. La convergence est au moins aussi rapide que celle de la suite géométrique $(1-c)^n$ vers 0.*

L'idée de la preuve est simple : on montre que la loi de transition (5) définit une application $T : p \mapsto p'$ qui est contractante de rapport $1 - c$. Le résultat découle ainsi du théorème du point fixe de Banach.

CONCLUSION

Pour être utile, un moteur de recherche doit non seulement énumérer les résultats d'une requête mais les classer par ordre d'importance. Or, estimer la pertinence des pages web est un profond défi de modélisation.

En première approximation Google analyse le graphe formé par les liens entre pages web. Interprétant un lien $j \to i$ comme « vote » de la page P_j en faveur de la page P_i, le modèle PageRank (6) définit une mesure de « popularité ».

Le théorème du point fixe assure que cette équation admet une unique solution, et justifie l'algorithme itératif (5) pour l'approcher. Celui-ci est facile à implémenter et assez efficace pour les graphes de grandeur nature.

Muni de ces outils mathématiques et d'une habile stratégie d'entreprise, Google gagne des milliards de dollars. Il fallait y penser !

RÉFÉRENCES

[1] S. Brin, L. Page : *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Stanford University 1998, `http://infolab.stanford.edu/pub/papers/google.pdf` (20 pages).

[2] M. Eisermann : *Comment fonctionne Google ?* Quadrature, no. 68, avril 2008, version étendue sur `http://www-fourier.ujf-grenoble.fr/~eiserm/enseignement#google` (15 pages).

DÉVELOPPEMENT MATHÉMATIQUE

L'objectif de cet appendice est de démontrer le théorème du point fixe énoncé ci-dessus. Les outils nécessaires sont de niveau licence : nous aurons besoin d'un peu de calcul matriciel (essentiellement pour une notation commode) et du théorème de point fixe de Banach pour les fonctions contractantes $f\colon \mathbb{R}^n \to \mathbb{R}^n$. Je reprends ici le développement de mon article [2].

REFORMULATION MATRICIELLE

Remarquons d'abord que l'équation (3) n'est rien d'autre qu'un système d'équations linéaires. Plus explicitement, pour tout couple d'indices $i, j \in \{1, \dots, n\}$, on définit a_{ij} par

$$a_{ij} := \begin{cases} \frac{1}{\ell_j} & \text{si } j \to i, \\ 0 & \text{sinon.} \end{cases} \tag{7}$$

On obtient ainsi une matrice $A = (a_{ij})$, et notre équation d'équilibre (3) s'écrit comme

$$\mu = A\mu \tag{8}$$

ou encore

$$(A - I)\mu = 0, \tag{9}$$

ce qui est un honnête système linéaire à n équations et n inconnues $\mu_1, \dots, \mu_n$.

Dans notre exemple miniature discuté ci-dessus, A est la matrice 12×12 suivante :

$$A = \begin{pmatrix}
\circ & 1/2 & 1/2 & 1/2 & \circ & 1/2 & \circ & \circ & \circ & \circ & \circ & \circ \\
1/4 & \circ & \circ & 1/2 & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
1/4 & 1/2 & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
1/4 & \circ & 1/2 & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
1/4 & \circ & \circ & \circ & \circ & \circ & 1 & \circ & 1/4 & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & 1/3 & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & 1/3 & 1/2 & \circ & 1/2 & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & 1/3 & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & 1/2 & \circ & 1/2 & 1/2 & 1/2 \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & 1/4 & \circ & \circ & 1/2 \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & 1/4 & 1/2 & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & 1/4 & \circ & 1/2 & \circ
\end{pmatrix}.$$

Comme énoncé, dans cet exemple l'équation $\mu = A\mu$ admet comme solution le vecteur

$$\mu = (2, 1, 1, 1, 3, 1, 2, 1, 2, 1, 1, 1)^{\dagger}.$$

MATRICES STOCHASTIQUES

Bien que nous n'utilisions que des arguments d'algèbre linéaire et un peu d'analyse dans $\mathbb{R}^n$, nous ne nous priverons pas du vocabulaire stochastique, car c'est le point de vue et le langage naturel de notre développement.

Par définition, notre matrice $A = (a_{ij})$ vérifie

$$\begin{aligned} a_{ij} &\geq 0 \qquad \text{pour tout } i, j \text{ et} \\ \textstyle\sum_i a_{ij} &= 1 \qquad \text{pour tout } j, \end{aligned}$$

ce que l'on appelle une *matrice stochastique*. (La somme de chaque colonne vaut 1, mais on ne peut en général rien dire sur la somme dans une ligne.) Nous supposons ici que toute page emet des liens. Ce n'est pas une restriction sérieuse : si jamais une page n'émet aucun lien on peut la faire pointer vers elle-même.

Nous interprétons a_{ij} comme la probabilité d'aller de la page P_j à la page P_i, en suivant un des ℓ_j liens au hasard. La marche aléatoire associée consiste à se balader sur le graphe suivant les probabilités a_{ij}.

MARCHE ALÉATOIRE

Supposons qu'un vecteur $x \in \mathbb{R}^n$ vérifie

$$x_j \geq 0 \quad \text{pour tout } j \text{ et} \quad \textstyle\sum_j x_j = 1,$$

ce que l'on appelle un *vecteur stochastique* ou une *mesure de probabilité* sur les pages $P_1, \dots, P_n$: on interprète x_j comme la probabilité de se trouver sur la page P_j.

Effectuons un pas dans la marche aléatoire : avec probabilité x_j on démarre sur la page P_j, puis on suit le lien $j \to i$ avec probabilité a_{ij}. Ceci nous fait tomber sur la page P_i avec une probabilité $a_{ij}x_j$. Au total, la probabilité d'arriver sur la page P_i par n'importe quel lien est

$$y_i = \textstyle\sum_j a_{ij}x_j. \tag{10}$$

Autrement dit, un pas dans la marche aléatoire correspond à l'application linéaire

$$T\colon \mathbb{R}^n \to \mathbb{R}^n, \quad x \mapsto y = Ax. \tag{11}$$

La marche aléatoire partant d'une probabilité initiale x^0 est l'itération de la transition $x^{t+1} = T(x^t)$ pour $t \in \mathbb{N}$.

Préservation de la masse

Si x est un vecteur stochastique, alors son image $y = Ax$ l'est aussi. Effectivement, $y_i \geq 0$ car $y_i = \sum_j a_{ij} x_j$ est une somme de termes positifs ou nuls. De plus on trouve

$$\begin{aligned}\sum_i y_i &= \sum_i \sum_j a_{ij} x_j = \sum_j \sum_i a_{ij} x_j \\ &= \sum_j \Big(\sum_i a_{ij} \Big) x_j = \sum_j x_j = 1.\end{aligned}$$

Mesure invariante

Une mesure de probabilité μ vérifiant $\mu = T(\mu)$ est appelée une *mesure invariante* ou une *mesure stationnaire* ou encore une *mesure d'équilibre*. En termes d'algèbre linéaire (8) c'est un vecteur propre associé à la valeur propre 1. En termes d'analyse, c'est un point fixe de l'application T. C'est ce dernier point de vue que nous allons exploiter ici.

Le modèle PageRank

Dans le modèle PageRank la loi de transition (5) se formalise comme l'application affine

$$T \colon \mathbb{R}^n \to \mathbb{R}^n, \quad x \mapsto c\varepsilon + (1-c)Ax. \tag{12}$$

Ici le vecteur stochastique $\varepsilon = (\frac{1}{n}, \ldots, \frac{1}{n})$ correspond à l'équiprobabilité, et A est la matrice stochastique définie par (7).

Remarque. Restreinte aux vecteurs stochastiques, l'application T est donnée par

$$T(x) = cEx + (1-c)Ax \tag{13}$$

où E est la matrice dont tous les coefficients valent $1/n$. Effectivement, sur le sous-espace affine des vecteurs $x \in \mathbb{R}^n$ vérifiant $\sum_j x_j = 1$ nous avons $Ex = \varepsilon$. La restriction de T coïncide donc avec l'application induite par la matrice stochastique $A_c = cE + (1-c)A$.

Le théorème du point fixe

Pour un vecteur $x \in \mathbb{R}^n$ on définit sa *norme* par $|x| := \sum_i |x_i|$. C'est une honnête norme, qui a toutes les bonnes propriétés usuelles. Ainsi $|x-y|$ mesure la distance entre deux points $x, y \in \mathbb{R}^n$ relative à la norme $|\cdot|$.

Définition. Une fonction $f \colon \mathbb{R}^n \to \mathbb{R}^n$ est dite *contractante* de rapport $k < 1$ si elle vérifie $|f(x) - f(y)| \leq k|x-y|$ pour tout $x, y \in \mathbb{R}^n$.

Théorème du point fixe (S. Banach 1922). *Si $f \colon \mathbb{R}^n \to \mathbb{R}^n$ est une fonction contractante de rapport $k < 1$, alors :*

- *Il existe un et un seul point $\mu \in \mathbb{R}^n$ vérifiant $f(\mu) = \mu$.*
- *Pour tout vecteur initial $x^0 \in \mathbb{R}^n$ la suite itérative $x^{m+1} = f(x^m)$ converge vers μ.*
- *On a $|x^m - \mu| \leq k^m |x^0 - \mu|$, la convergence de x^m vers μ est donc au moins aussi rapide que celle de la suite géométrique k^m vers 0.*
- *Pour le calcul concret on a l'estimation de l'écart $|x^m - \mu| \leq \frac{k}{1-k} |x^m - x^{m-1}|$.* □

Dans la pratique, on ignore la limite μ mais on peut facilement calculer la suite itérative x^m. Pour contrôler la qualité de l'approximation x^m, on majore l'écart $|x^m - \mu|$ entre x^m et la limite inconnue par la quantité $\frac{k}{1-k} |x^m - x^{m-1}|$.

Application au modèle PageRank

Nous disposons maintenant de tous les outils nécessaires pour montrer que le modèle PageRank admet un unique solution :

Proposition. *Soit $A \in \mathbb{R}^{n \times n}$ une matrice stochastique quelconque et soit c une constante vérifiant $0 < c \leq 1$. Alors l'application affine $T \colon \mathbb{R}^n \to \mathbb{R}^n$ définie par* (12) *est contractante de rapport $k = 1 - c$.*

Démonstration. Regardons deux vecteurs $x, y \in \mathbb{R}^n$ et majorons la norme de $z := Tx - Ty$ en fonction de $|x-y|$. On a $z = kA(x-y)$ donc $z_i = k \sum_j a_{ij}(x_j - y_j)$ pour tout $i = 1, \ldots, n$.

Ceci nous permet de calculer la norme :

$$\begin{aligned}|Tx-Ty| = |z| &= \sum_i |z_i| \\ &= \sum_i \Big| k \sum_j a_{ij}(x_j - y_j) \Big| \\ &\leq k \sum_i \sum_j |a_{ij}(x_j - y_j)| \\ &= k \sum_j \sum_i a_{ij} |x_j - y_j| \\ &= k \sum_j \Big(\sum_i a_{ij} \Big) |x_j - y_j| \\ &= k \sum_j |x_j - y_j| \\ &= k|x-y|.\end{aligned}$$

Ceci prouve que $T : \mathbb{R}^n \to \mathbb{R}^n$ est contractante de rapport k comme énoncé. □

Remarque. La proposition inclut le cas trivial $c = 1$: dans ce cas $T(x) = \varepsilon$ est constante, donc $x = \varepsilon$ est l'unique point fixe. Dans l'autre extrême on pourrait considérer $c = 0$, mais $T = A$ n'est pas forcément contractante. Par exemple pour un graphe à n sommets sans arêtes entre eux, nous obtenons la matrice identité, $A = I$, qui admet tout vecteur $x \in \mathbb{R}^n$ comme point fixe. Un bon choix de c se situe donc quelque part entre 0 et 1.

Corollaire. *Pour $0 < c \leq 1$ l'application T admet une unique mesure invariante $\mu = T(\mu)$ et pour tout vecteur initial x^0 la suite itérative $x^{m+1} = T(x^m)$ converge vers le point fixe μ, au moins aussi rapidement que $(1-c)^m \to 0$.*

Démonstration. L'application T étant contractante, elle admet un unique point fixe $\mu \in \mathbb{R}^n$. Il ne reste qu'à vérifier que le point fixe est un vecteur stochastique, c'est-à-dire qu'il satisfait $\mu_i \geq 0$ et $\sum_i \mu_i = 1$: si l'on démarre avec un vecteur stochastique x^0, alors tous les itérés x^m restent stochastiques, donc leur limite μ l'est aussi. (Exercice.) □

Remarque. Le résultat précédent se généralise au théorème de Perron–Frobenius : si une matrice réelle A a tous ses coefficients positifs, $a_{ij} > 0$ pour $i, j = 1, \ldots, n$, alors le rayon spectral de A est donné par une valeur propre $\lambda \in \mathbb{R}_+$, l'espace propre associé E_λ est de dimension 1, et il existe un vecteur propre $v \in E_\lambda$ dont tous les coefficients sont positifs.

Remarque. L'algorithme itératif correspondant est souvent appelé la « méthode de la puissance ». Il se généralise à une matrice A quelconque et permet d'approcher numériquement un vecteur propre v associé à la valeur propre λ de module $|\lambda|$ maximal, pourvu que cette valeur propre soit unique et simple.

QUELQUES APPROFONDISSEMENTS

DE L'ALGORITHME À L'IMPLÉMENTATION

Rappelons que la matrice A représentant le graphe du web est très grande : en 2004 Google affirmait que « le classement est effectué grâce à la résolution d'une équation de 500 millions de variables et de plus de 3 milliards de termes. » Comment est-ce possible ?

La manière usuelle de stocker une matrice de taille $n \times n$ est un grand tableau de n^2 coefficients indexés par $(i, j) \in \{1, \ldots, n\}^2$. Il est envisageable de stocker ainsi une matrice 1000×1000, c'est-à-dire un million de coefficients mais ceci est hors de question pour une matrice $n \times n$ où $n \approx 10^6$, voire $n \approx 10^8$.

Dans notre cas la plupart des coefficients de la matrice valent zéro car une page n'émet que quelques douzaines de liens typiquement. Dans ce cas, il suffit de stocker les coefficients non nuls, dont le nombre est d'ordre n et non n^2. Une telle matrice est appelée *creuse*.

Pour des applications réalistes, il est donc nécessaire d'implémenter des structures de données et des méthodes adaptées aux matrices creuses. La méthode du point fixe est faite sur mesure pour ce genre d'application, et la loi de transition (5) est facile à implémenter, voir [2].

CHAÎNES DE MARKOV ET ERGODICITÉ

Ce que nous venons d'étudier sont des chaînes de Markov, à temps discret et ici à espace d'états fini. En plus nos chaînes de Markov sont homogènes dans le sens que la loi de transition ne change pas au cours du temps.

Le choix du paramètre $c \in]0,1]$, qui gère la téléportation sur le graphe, garantit que notre chaîne de Markov est irréductible et apériodique. Dans cette situation on a toujours convergence vers une unique mesure stationnaire μ : les puissances A^t, où $t \in \mathbb{N}$, convergent vers la matrice dont chaque colonne est μ. En particulier, la mesure $x^t = A^t x^0$ converge vers μ indépendamment de la mesure initiale x^0.

Dans cette situation dite « ergodique » la loi des grands nombres est en vigueur : la moyenne « en temps » d'une observable h le long d'une trajectoire est égale à sa moyenne « en espace ». Plus précisément, pour presque toute trajectoire $(\omega_t)_{t\in\mathbb{N}}$ on a l'égalité

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} h(\omega_t) = \sum_j h(j)\mu_j. \tag{14}$$

En particulier, μ_i est la fréquentation moyenne de la page P_i. Ceci justifie notre interprétation que les pages avec une grande probabilité μ_i sont les plus fréquentées, autrement dit les plus populaires.

QUELQUES POINTS DE RÉFLEXION

LE MODÈLE EST-IL PLAUSIBLE ?

La structure caractéristique des documents hypertextes sont les citations mutuelles : l'auteur d'une page web ajoute des liens vers les pages qu'il considère utiles ou intéressantes.

L'hypothèse à la base du modèle PageRank est que l'on peut interpréter un lien comme un vote ou une recommandation. Des millions d'auteurs de pages web lisent et jugent mutuellement leurs pages, et leurs jugements s'expriment par leurs liens. Le modèle de la marche aléatoire en profite en transformant l'évaluation mutuelle en une mesure globale de popularité.

Cet argument de plausibilité sera à débattre et à analyser plus en détail. L'ultime argument en faveur du modèle PageRank, par contre, est son succès : le classement des résultats semble bien refléter les attentes des utilisateurs.

DESCRIPTIF OU NORMATIF ?

Au début de son existence, Google se voulait un outil *descriptif* : si une page est importante, alors elle figure en tête du classement.

Son écrasant succès a fait de Google une référence *normative* : si une page figure en tête du classement, alors elle est importante.

Pour des sites web commerciaux, l'optimisation de leur classement PageRank est ainsi devenue un enjeu vital. Afin d'améliorer son classement, il suffit d'attirer des liens, de préférence ceux émis des pages importantes, et il vaut mieux en émettre très peu.

Ces stratégies et astuces sont devenues un domaine très actif, dit « *search engine optimization* » (SEO). Cette évolution rend l'évaluation des pages web encore plus difficile : comme l'approche et l'importance de Google sont mondialement connues, les liens s'utilisent différemment de nos jours.

Ainsi l'omniprésence de Google change l'utilisation des liens par les auteurs des pages web… ce qui remet en question l'hypothèse à la base même du modèle PageRank.

E-mail address: Michael.Eisermann@ujf-grenoble.fr

INSTITUT FOURIER, UNIVERSITÉ GRENOBLE I, FRANCE

URL: www-fourier.ujf-grenoble.fr/~eiserm

COMMENT FONCTIONNE GOOGLE?

MICHAEL EISERMANN

RÉSUMÉ. Le point fort du moteur de recherche Google est qu'il trie intelligemment ses résultats par ordre d'importance. Nous expliquons ici l'algorithme PageRank qui est à la base de ce classement. Il faut d'abord établir un modèle qui permet de définir ce que l'on entend par « importance ». Une fois ce modèle formalisé, il s'agit de résoudre astucieusement un immense système d'équations linéaires.

Il va sans dire que l'application pratique est devenue très importante. Bien qu'élémentaires, les arguments mathématiques sous-jacents n'en sont pas moins intéressants : l'approche fait naturellement intervenir l'algèbre linéaire, la « marche aléatoire » sur un graphe et le théorème du point fixe. Tout ceci en fait un très beau sujet pour la culture des mathématiques et leurs applications.

TABLE DES MATIÈRES

Introduction 1
1. Que fait un moteur de recherche ? 2
2. Comment mesurer l'importance d'une page web ? 3
3. Marche aléatoire sur la toile 6
4. Existence et unicité d'une solution 8
5. Implémentation efficace 11
6. Quelques points de réflexion 12
Références 15

INTRODUCTION

Cet article discute les mathématiques utilisées par Google, un moteur de recherche généraliste qui a eu un succès fulgurant depuis sa création en 1998. Le point fort de Google est qu'il trie *par ordre d'importance* les résultats d'une requête, c'est-à-dire les pages web associées aux mots-clés cherchés. L'étonnante efficacité de cette méthode a fait le succès de Google et la fortune de ses fondateurs, Sergey Brin et Lawrence Page. L'idée est née lors de leur thèse de doctorat, puis publiée dans leur article [1]. Il s'agit essentiellement de résoudre un grand système d'équations linéaires et fort heureusement l'algorithme itératif qui en découle est aussi simple que puissant. On s'intéresse ici de plus près à cet algorithme, à la fois simple et ingénieux. En conjonction avec une habile stratégie d'entreprise, on pourrait dire que Google gagne des milliards de dollars avec l'algèbre linéaire !

Ajoutons que Google a eu la chance de naître dans une situation favorable, quand la « nouvelle économie » était encore en pleine croissance : le volume d'internet explosait et les moteurs de recherche de première génération avaient du mal à s'adapter aux exigences grandissantes. Si vous voulez savoir plus sur la foudroyante histoire de l'entreprise Google, ses légendes et anecdotes, vous lirez avec profit le livre de David Vise et Mark Malseed [2].

Date: 16 mai 2006. *Dernière mise à jour*: 13 mai 2009. *URL*: `www-fourier.ujf-grenoble.fr/~eiserm`
Université Joseph Fourier • Licence de Mathématiques • cours « Mathématiques assistées par ordinateur ».

1. Que fait un moteur de recherche ?

1.1. **Fouille de données.** À première vue, le principe d'un moteur de recherche est simple : on copie d'abord les pages web concernées en mémoire locale, puis on trie le contenu (les mots-clés) par ordre alphabétique afin d'effectuer des recherches lexiques. Une *requête* est la donnée d'un ou plusieurs mots-clés ; la *réponse* est une liste des pages contenant les mots-clés recherchés. C'est en gros ce que faisaient les moteurs de recherche, dits de première génération, dans les années 1990. Après réflexion, cette démarche simpliste n'est pas si évidente car la quantité des documents à gérer est énorme et rien que le stockage et la gestion efficaces posent des défis considérables. Et cela d'autant plus que les requêtes doivent être traitées en temps réel : on ne veut pas la réponse dans une semaine, mais *tout de suite*.

Une implémentation opérationnelle à cette échelle doit donc employer la force brute d'un réseau puissant, afin de répartir les données et les tâches sur plusieurs ordinateurs travaillant en parallèle. Plus important encore sont les algorithmes, hautement spécialisés et optimisés, sans lesquels même un réseau de quelques milliers d'ordinateurs resterait impuissant devant cette tâche herculéenne. Pour se faire une idée de l'ordre de grandeur, voici quelques chiffres sur l'entreprise Google :

Cerveaux: environ 6 000 employés (début 2006)
Ordinateurs: plus de 60 000 PC en réseau sous Linux (on ignore les chiffres exacts)
Mémoire vive: plus de 130 000 Go de RAM pour les calculs (on ignore les chiffres exacts)
Disques dures: plus de 5 000 000 Go pour stocker les données (on ignore les chiffres exacts)
Trafic en ligne: quelques milliers de requêtes par secondes (on ignore les chiffres exacts)
Part du marché: plus de 50% aux États-Unis et dans de nombreux autres pays
Chiffre d'affaires: plus de $6 000 000 000 en 2005, dont $1 465 400 000 bénéfices net
Cotation boursière: environ $80 000 000 000 en août 2005

Précisons que la recherche sur Google est un service gratuit. En 1998 il n'était pas du tout évident comment gagner de l'argent avec un produit gratuit, aussi apprécié qu'il soit. Jusqu'en 2000 l'entreprise accumulait des pertes et fut même menacée de faillite. L'idée qui l'a sauvée a été la vente de *liens commerciaux* et depuis 2001 ces placements publicitaires génèrent de plus en plus de bénéfices.

1.2. **Classement des résultats.** L'énorme quantité des données entraîne un deuxième problème, plus délicat encore : les pages trouvées sont souvent trop nombreuses, il faut donc en choisir les plus pertinentes. La grande innovation apportée par Google en 1998 est le tri des pages par ordre d'importance. Ce qui est frappant est que cet ordre correspond assez précisément aux attentes des utilisateurs.

Par exemple, si vous vous intéressez à la programmation et vous faites chercher les mots-clés « C++ compiler », vous trouverez quelques millions de pages. Des pages importantes comme `gcc.gnu.org` se trouvent quelque part en tête du classement, ce qui est très raisonnable. Par contre, une petite page personnelle, où l'auteur mentionne qu'il ne connaît rien du C++ et n'arrive pas à compiler, ne figurera que vers la fin de la liste, ce qui est également raisonnable. Comment Google distingue-t-il les deux ?

Selon les informations fournies par l'entreprise elle-même, l'index de Google porte sur plus de 8 milliards de documents web. Une bonne partie des informations répertoriées, pages web et documents annexes, changent fréquemment. Il est donc hors de question de les classer manuellement, par des êtres humains : ce serait trop coûteux, trop lent et jamais à jour. L'importance d'une page doit donc être déterminée de manière automatisée, par un algorithme. Comment est-ce possible ?

2. COMMENT MESURER L'IMPORTANCE D'UNE PAGE WEB ?

2.1. **Le web est un graphe.** La particularité des documents *hypertexte* est qu'ils fournissent des liens, des références mutuelles pointant de l'une vers l'autre. Ainsi on peut considérer le web comme un immense *graphe*, dont chaque page web j est un *sommet* et chaque lien $j \to i$ est une *arête*.

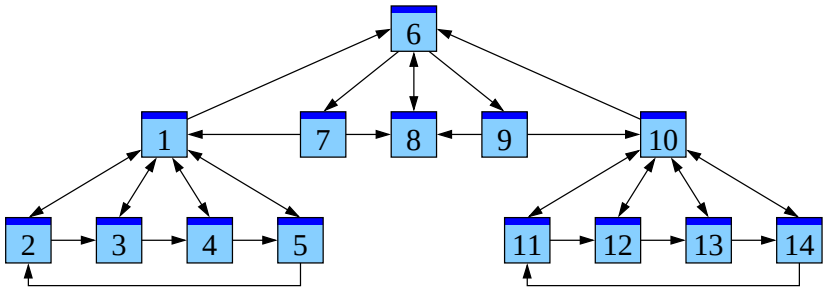


FIG. 1. Le web vu comme un graphe

Dans la suite on numérote les pages par $1,2,3,\ldots,n$ et on écrit $j \to i$ si la page j pointe vers la page i (au moins une fois ; on ne compte pas les liens multiples). Ainsi chaque page j émet un certains nombre ℓ_j de liens vers des pages « voisines ». À noter que les arêtes sont orientées : si l'on a $j \to i$, on n'a pas forcément le sens inverse $i \to j$. Le graphe de la figure 1, par exemple, s'écrit comme suit :

$1 \to 2,3,4,5,6$; $2 \to 1,3$; $3 \to 1,4$; $4 \to 1,5$; $5 \to 1,2$; $6 \to 7,8,9$; $7 \to 8,1$; $8 \to 6$;
$9 \to 8,10$; $10 \to 6,11,12,13,14$; $11 \to 10,12$; $12 \to 10,13$; $13 \to 10,14$; $14 \to 10,11$.

2.2. **Comment repérer des pages importantes ?** Dans une première approximation nous allons négliger le contenu des pages et ne tenir compte que de la structure du graphe.

- Regardons d'abord le groupe des pages $1,2,3,4,5$. Le dessin suggère que la page 1 sert de racine tandis que les pages $2,3,4,5$ sont subordonnées. Dans ce sens, la page 1 sera sans doute un bon point de départ si vous cherchez des informations.
- Il en est de même pour le groupe $10,11,12,13,14$, où la page 10 sert de racine alors que $11,12,13,14$ sont subordonnées. À titre d'exemple, il pourrait s'agir d'une page d'accueil et quatre pages annexes, ou d'une introduction et quatre chapitres d'un ouvrage.
- La structure du groupe $6,7,8,9$ est similaire. À noter toutefois que les pages 1 et 10, déjà reconnues comme importantes, font toutes deux référence à la page 6. On pourrait ainsi soupçonner que la page 6 contient de l'information essentielle pour tout l'ensemble.

Heuristiquement, on conclut que les pages $1,6,10$ semblent les plus importantes, avec une légère préférence pour la page 6. Soulignons toutefois que notre dessin dans le plan suggère une organisation hiérarchique qui n'est qu'artificielle. Un ordinateur qui analyse cette situation n'a que l'information brute des liens $1 \to 2,3,4,5,6$; $2 \to 1,3$; etc.

Question 1. Est-il possible, par un algorithme, d'associer à chaque page $i = 1,\ldots,n$ une *mesure d'importance* ? Plus explicitement, on souhaite que cette mesure soit un nombre réel $\mu_i \geq 0$ avec la convention que plus μ_i est grand, plus la page i est « importante ».

Remarque 2. La notion d'importance d'une page est nécessairement vague. Qu'est-ce que l'importance ? Peut-il y avoir une mesure objective ? Si oui, comment la définir ? Cette question semble au cœur de toute la problématique. Si vous avez une nouvelle idée pertinente à ce sujet, implémentez-la et devenez riche ! (Ou bien venez en discuter avec moi.)

Dans la suite notre but sera modeste : le mieux que l'on puisse espérer est que notre analyse dégage un résultat qui *approche* bien l'importance *ressentie* par les utilisateurs. Pour toute application professionnelle les résultats numériques seront à tester et à calibrer empiriquement.

2.3. **Première idée : comptage des liens.** Il est plausible qu'une page importante reçoit beaucoup de liens. Avec un peu de naïveté, on croira aussi l'affirmation réciproque : si une page reçoit beaucoup de liens, alors elle est importante. Ainsi on pourrait définir l'importance μ_i de la page i comme suit :

$$\mu_i = \sum_{j \to i} 1. \tag{1}$$

Interprétation: La somme (1) veut juste dire que μ_i est égal au nombre de liens $j \to i$ reçus par i. C'est facile à définir et facile à calculer : il suffit de compter.

Exemple: Dans notre exemple, les pages 1 et 10 reçoivent 5 liens chacune, alors que la page 6 n'en reçoit que 3. Ainsi $\mu_1 = \mu_{10} = 5$ mais seulement $\mu_6 = 3$.

Inconvénient: La mesure μ ainsi définie ne correspond pas à l'importance ressentie par les utilisateurs : elle sous-estime l'importance de la page 6.

Manipulation: On peut artificiellement augmenter l'importance d'une page i en créant des pages « vides de sens » pointant vers i. Cette faiblesse fait du comptage une approche peu fiable.

2.4. **Seconde idée : comptage pondéré.** Certaines pages j émettent beaucoup de liens : ceux-ci sont donc moins spécifiques et dans un certain sens leur *poids* est plus faible. Ainsi on pourrait définir une mesure d'importance plus fine comme suit :

$$\mu_i = \sum_{j \to i} \frac{1}{\ell_j}. \tag{2}$$

Interprétation: Comme avant, la somme (2) compte les liens reçus par la page i, mais maintenant chaque lien $j \to i$ n'est compté qu'avec un poids $\frac{1}{\ell_j}$. Il suffit de sommer.

Exemple: Dans notre exemple, on trouve des sommes $\mu_1 = \mu_{10} = 2.5$ et $\mu_6 = 1.4$.

Inconvénient: La mesure μ ainsi définie ne correspond toujours pas bien à l'importance ressentie par les utilisateurs : elle sous-estime à nouveau l'importance de la page 6.

Manipulation: Comme avant, on peut artificiellement augmenter l'importance d'une page i en créant une foule de pages « vides » pointant vers i. De nouveau, la mesure n'est pas fiable.

2.5. **Troisième idée : définition récursive.** La dernière idée en date, finalement, est celle utilisée par Google. Le principe : *une page i est importante si beaucoup de pages importantes pointent vers i.* Ainsi on est amené à définir l'importance μ_i de manière récursive comme suit :

$$\mu_i = \sum_{j \to i} \frac{1}{\ell_j} \mu_j. \tag{3}$$

Interprétation: La somme (3) compte chaque lien reçu par i avec poids $\frac{1}{\ell_j}\mu_j$: ceci tient compte de l'importance μ_j de la page d'origine j, et du nombre ℓ_j des liens qui en sont émis.

Exemple: Dans notre exemple, on trouve, après calcul, les valeurs $\mu_6 = 6$ et $\mu_1 = \mu_{10} = 5$ puis $\mu_8 = 4$. Les autres pages suivent avec un grand écart et n'obtiennent que $\mu_i = 2$.

Plausibilité: Les pages $6, 1, 10, 8$ sont effectivement repérées comme les plus importantes. Ceci veut dire que la mesure μ ainsi obtenue correspond assez bien à l'importance ressentie par les utilisateurs, comme motivée ci-dessus. (On discutera pourquoi au §6.)

Robustesse: Si l'on ajoute des pages « vides de sens » elles recevront l'importance 0 et ne contribueront pas au calcul. Ainsi la manipulation évidente n'influence plus le résultat.

L'équation (3) est facile à écrire mais moins évidente à résoudre : naïvement parlant, pour calculer μ_i il faut d'abord connaître les termes de droite, donc les μ_j, ce qui a l'air circulaire… Notre objectif est donc d'expliquer pourquoi une solution existe et comment la calculer de manière efficace.

2.6. **Apparaît l'algèbre linéaire…** Après réflexion, l'équation (3) n'est rien autre qu'un système d'équations linéaires. Plus explicitement, pour tout couple d'indices $i, j \in \{1, \dots, n\}$, on définit a_{ij} par

$$a_{ij} := \begin{cases} \frac{1}{\ell_j} & \text{si } j \to i, \\ 0 & \text{sinon.} \end{cases} \tag{4}$$

On obtient ainsi une matrice $A = (a_{ij})$, et notre équation (3) s'écrit comme

$$\mu = A\mu \qquad \text{ou encore} \qquad (A - I)\mu = 0,$$

ce qui est un honnête système linéaire, que l'on peut résoudre par des méthodes adéquates.

Exemple 3. Dans notre exemple, A est la matrice 14×14 explicitée ci-dessous et l'équation $\mu = A\mu$ admet la solution énoncée. (Le vérifier !) C'est même la seule à multiplication par un scalaire près.

$$A = \begin{pmatrix}
0 & 1/2 & 1/2 & 1/2 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/5 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/5 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/5 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/5 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1/5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1/5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/3 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 1/2 & 1/2 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/5 & 0 & 0 & 0 & 1/2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/5 & 1/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/5 & 0 & 1/2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/5 & 0 & 0 & 1/2 & 0
\end{pmatrix}, \quad \mu = \begin{pmatrix} 5 \\ 2 \\ 2 \\ 2 \\ 2 \\ 6 \\ 2 \\ 4 \\ 2 \\ 5 \\ 2 \\ 2 \\ 2 \\ 2 \end{pmatrix}.$$

Définition 4. Soit $A \in \mathbb{R}^{n \times n}$ une matrice et soit $v \in \mathbb{R}^n \smallsetminus \{0\}$ un vecteur non nul. Si $Av = \lambda v$ pour un scalaire $\lambda \in \mathbb{R}$, alors on dit que v est un *vecteur propre* de la matrice A, associé à la *valeur propre* λ.

Pour notre application, nous nous intéressons donc aux vecteurs propres de A associé à $\lambda = 1$. On montrera plus bas qu'un tel vecteur propre existe et que la solution est essentiellement unique (§4.2).

3. Marche aléatoire sur la toile

3.1. **Matrices stochastiques.** Les arguments du §2 nous mènent à étudier une certaine matrice A qui code la structure du web. Avant de résoudre l'équation $A\mu = \mu$, on va essayer d'en développer une intuition. L'idée est de réinterpréter μ comme une mesure de « popularité » des pages web.

Chaque page j émet un certain nombre ℓ_j de liens, ce que l'on code par des coefficients a_{ij} suivant l'équation (4) ci-dessus. Par la suite nous supposons que $\ell_j \geq 1$, ce qui n'est pas une restriction sérieuse : si jamais une page n'émet pas de liens on peut la faire pointer vers elle-même.

Selon sa définition, notre matrice $A = (a_{ij})$ vérifie

$$\begin{aligned} a_{ij} &\geq 0 && \text{pour tout } i, j \text{ et} \\ \sum_i a_{ij} &= 1 && \text{pour tout } j, \end{aligned}$$

ce que l'on appelle une *matrice stochastique*. (À noter que la somme de chaque colonne vaut 1, mais on ne peut en général rien dire sur la somme dans une ligne.)

On peut interpréter a_{ij} comme la probabilité d'aller de la page j à la page i, en suivant un des ℓ_j liens au hasard. La *marche aléatoire* associée consiste à se balader sur le graphe suivant les probabilités a_{ij}. Notre modèle admet ainsi une étonnante interprétation probabiliste : aussi étrange que cela puisse apparaître, on modélise un surfeur aléatoire, qui ne lit jamais rien mais qui clique au hasard !

Soulignons donc à nouveau que ce n'est pas le contenu des pages web qui soit pris en compte pour le calcul de « l'importance », mais uniquement la structure du graphe formé par les pages et les liens entre elles. (Ne vous faites pas trop de souci pour l'instant, on discutera plus bas, au §6, pourquoi ce modèle est tout de même plausible.)

3.2. **Convergence vers une mesure invariante.** Supposons qu'un vecteur $x \in \mathbb{R}^n$ vérifie

$$x_j \geq 0 \quad \text{pour tout } j \text{ et} \quad \sum_j x_j = 1,$$

ce que l'on appelle un *vecteur stochastique* ou une *mesure de probabilité* sur les pages $1, \ldots, n$: on interprète x_j comme la probabilité de se trouver sur la page j.

Effectuons un pas dans la marche aléatoire : avec probabilité x_j on démarre sur la page j, puis on suit le lien $j \to i$ avec probabilité a_{ij}. Ce chemin nous fait tomber sur la page i avec une probabilité $a_{ij}x_j$. Au total, la probabilité d'arriver sur la page i, par n'importe quel chemin, est la somme

$$y_i = \sum_j a_{ij} x_j.$$

Autrement dit, un pas dans la marche aléatoire correspond à l'application linéaire

$$T : \mathbb{R}^n \to \mathbb{R}^n, \quad x \mapsto y = Ax.$$

Remarque 5. Si x est un vecteur stochastique, alors son image $y = Ax$ aussi. Effectivement, $y_i \geq 0$ car $y_i = \sum_j a_{ij}x_j$ est une somme de termes positifs ou nuls et, de plus,

$$\sum_i y_i = \sum_i \sum_j a_{ij}x_j = \sum_j \sum_i a_{ij}x_j = \sum_j \Big(\sum_i a_{ij}\Big) x_j = \sum_j x_j = 1.$$

Définition 6. Une mesure de probabilité μ vérifiant $\mu = T(\mu)$ est appelée une *mesure invariante* ou une *mesure d'équilibre*. En termes d'algèbre linéaire c'est un vecteur propre associé à la valeur propre 1. En termes d'analyse, μ est un point fixe de l'application T.

Exemple 7. Itérer la marche aléatoire avec une probabilité initiale u_0 veut dire que l'on considère les mesures de probabilités successives $u_1, u_2, u_3, \ldots$ définies par $u_{t+1} = Au_t$. Voici un exemple démarrant sur la page 8, c'est-à-dire $u_0 = (0,0,0,0,0,0,0,1,0,0,0,0,0,0)$:

temps	page 1	page 2	page 3	page 4	page 5	page 6	page 7	page 8	page 9	page 10	page 11	page 12	page 13	page 14
t=0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000
t=1	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
t=2	0.000	0.000	0.000	0.000	0.000	0.000	0.333	0.333	0.333	0.000	0.000	0.000	0.000	0.000
t=3	0.167	0.000	0.000	0.000	0.000	0.333	0.000	0.333	0.000	0.167	0.000	0.000	0.000	0.000
t=4	0.000	0.033	0.033	0.033	0.033	0.400	0.111	0.111	0.111	0.000	0.033	0.033	0.033	0.033
t=5	0.122	0.017	0.017	0.017	0.017	0.111	0.133	0.244	0.133	0.122	0.017	0.017	0.017	0.017
t=6	0.100	0.033	0.033	0.033	0.033	0.293	0.037	0.170	0.037	0.100	0.033	0.033	0.033	0.033
t=7	0.084	0.036	0.036	0.036	0.036	0.210	0.098	0.135	0.098	0.084	0.036	0.036	0.036	0.036
t=8	0.122	0.035	0.035	0.035	0.035	0.168	0.070	0.168	0.070	0.122	0.035	0.035	0.035	0.035
t=9	0.105	0.042	0.042	0.042	0.042	0.217	0.056	0.126	0.056	0.105	0.042	0.042	0.042	0.042
...														
t=28	0.125	0.050	0.050	0.050	0.050	0.151	0.050	0.100	0.050	0.125	0.050	0.050	0.050	0.050
t=29	0.125	0.050	0.050	0.050	0.050	0.150	0.050	0.100	0.050	0.125	0.050	0.050	0.050	0.050
t=30	0.125	0.050	0.050	0.050	0.050	0.150	0.050	0.100	0.050	0.125	0.050	0.050	0.050	0.050

On observe un phénomène de *diffusion*, très plausible après réflexion :
- On commence au temps $t = 0$ sur la page 8 avec probabilité 1.000.
- Au temps $t = 1$, on se trouve sur la page 6 avec probabilité 1.000, suivant le seul lien $8 \to 6$.
- Pour $t = 2$, on tombe sur une des pages voisines suivant $6 \to 7, 8, 9$, chacune avec probabilité $\frac{1}{3}$.
- Dans les itérations suivantes la probabilité se propage sur tout le graphe.
  On constate qu'à partir de $t = 5$ la distribution est partout non nulle.
- Après 30 itérations, on est très proche (à 10^{-3} près) de la solution μ déjà exhibée ci-dessus.

On conclut, au moins empiriquement, que la probabilité tend vers notre distribution d'équilibre μ. À noter qu'il ne s'agit pas de l'équiprobabilité : certaines pages sont plus fréquentées que d'autres ! Comme motivé plus haut, ceci reflète bien le rôle particulier des pages $6, 1, 10, 8$.

Remarque 8. L'interprétation de la limite $u_t \to \mu$ est la suivante : μ_i est la probabilité de se trouver sur la page i après une très longue marche aléatoire. Ainsi les pages avec une grande probabilité μ_i sont les plus fréquentées ou les plus « populaires ». Dans la quête de classer les pages web par ordre d'importance, c'est encore un argument pour utiliser la mesure μ comme indicateur.

3.3. **Le modèle PageRank utilisé par Google.** Il se trouve que notre modèle a encore un grave défaut, quant aux propriétés mathématiques ainsi qu'à son utilité pratique :

Exemple 9. Le graphe suivant est une légère variante de l'exemple donné au §2.1, où s'ajoute la page 15 qui n'émet pas de liens. Pourtant, le résultat diffère drastiquement : la seule mesure invariante est $\mu = (0, \ldots, 0, 1)$, car notre surfeur aléatoire tombera tôt ou tard sur la page 15, où il demeure pour le reste de sa vie. Ce résultat ne reflète évidemment pas l'importance des pages, qui devrait rester inchangée (ou presque).

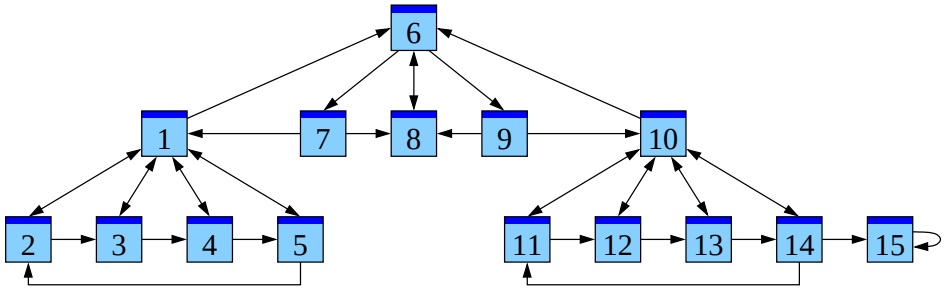


FIG. 2. Une variante du graphe initial

Pour cette raison Google utilise un modèle plus raffiné, dépendant d'un paramètre $c \in [0,1]$:

- Avec probabilité c, le surfeur abandonne la page actuelle et recommence sur une des n pages du web, choisie de manière équiprobable.
- Avec probabilité $1-c$, le surfeur suit un des liens de la page actuelle j, choisi de manière équiprobable parmi tous les ℓ_j liens émis. (C'est la marche aléatoire discutée ci-dessus.)

Cette astuce évite en particulier de se faire piéger par une page sans issue. Plus généralement, elle garantit d'arriver n'importe où dans le graphe, indépendamment des questions de connexité.

Ce nouveau modèle probabiliste se formalise comme l'application affine

$$T : \mathbb{R}^n \to \mathbb{R}^n, \quad x \mapsto c\varepsilon + (1-c)Ax.$$

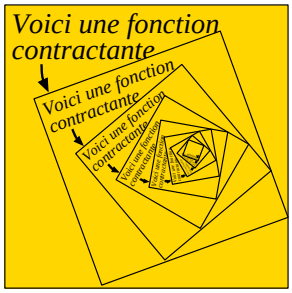
Ici A est la matrice stochastique définie par l'équation (4). Le vecteur stochastique $\varepsilon = (\frac{1}{n}, \ldots, \frac{1}{n})$ correspond à l'équiprobabilité sur toutes les pages. La constante $c \in [0,1]$ est un paramètre du modèle.

Remarque 10. La valeur $\frac{1}{c}$ est le *nombre moyen* de pages visitées (= liens suivis plus 1) avant de recommencer sur une page aléatoire. En général, on choisira la constante c positive mais proche de zéro. Par exemple, $c = 0.15$ correspond à suivre environ 6 liens en moyenne. (On pourrait argumenter que ceci correspond empiriquement au comportement des utilisateurs… à débattre.)

Exercice 11. Si vous vous y connaissez en probabilité, prouvez la remarque précédente.

4. EXISTENCE ET UNICITÉ D'UNE SOLUTION

4.1. **Le théorème du point fixe.** Une fonction $f : \mathbb{R} \to \mathbb{R}$ est *contractante* de rapport $k < 1$ si elle vérifie $|f(x) - f(y)| \leq k|x-y|$ pour tout $x, y \in \mathbb{R}$. Sous cette hypothèse, f admet un et un seul point fixe $\mu \in \mathbb{R}$, $f(\mu) = \mu$, et pour tout $u_0 \in \mathbb{R}$ la suite itérative $u_{m+1} = f(u_m)$ converge vers μ.



C'est exactement l'argument qu'il nous faut pour notre application. On a déjà vu, d'ailleurs, que la convergence se produisait dans notre exemple ci-dessus. Est-ce une coïncidence ? Non, c'est encore une manifestation du fameux théorème du point fixe. Comme nous travaillons sur les vecteurs $x \in \mathbb{R}^n$, nous sommes amenés à le généraliser convenablement :

Définition 12. Pour un vecteur $x \in \mathbb{R}^n$ on définit sa *norme* par $|x| := \sum_i |x_i|$. (C'est une honnête norme, qui a toutes les bonnes propriétés usuelles.) Une fonction $f : \mathbb{R}^n \to \mathbb{R}^n$ est dite *contractante* de rapport $k < 1$ (par rapport à la norme $|\cdot|$) si elle vérifie $|f(x) - f(y)| \leq k|x-y|$ pour tout $x, y \in \mathbb{R}^n$.

Théorème 13 (le théorème du point fixe)**.** *Si $f\colon \mathbb{R}^n \to \mathbb{R}^n$ est une fonction contractante de rapport $k<1$, alors :*

(1) *Il existe un et un seul point $\mu \in \mathbb{R}^n$ vérifiant $f(\mu)=\mu$.*

(2) *Pour toute valeur initiale $u_0 \in \mathbb{R}^n$ la suite itérative $u_{m+1}=f(u_m)$ converge vers μ.*

(3) *On a $|u_m-\mu| \le k^m|u_0-\mu|$, la convergence vers μ est donc au moins aussi rapide que celle de la suite géométrique k^m vers 0. Pour le calcul concret on a l'estimation de l'écart*

$$|u_m-\mu| \le \frac{k}{1-k}|u_m-u_{m-1}|.$$

Remarque 14. Dans la pratique, on ignore souvent la limite μ mais on peut facilement calculer la suite itérative u_m. Pour contrôler la qualité de l'approximation u_m, on majore l'écart $|u_m-\mu|$ entre u_m et la limite inconnue par la quantité $\frac{k}{1-k}|u_m-u_{m-1}|$, très facile à calculer.

Démonstration. Comme il s'agit d'une très belle preuve, je ne peux m'empêcher de la refaire ici.

Unicité. — Si $x,y \in \mathbb{R}^n$ sont deux points fixes d'une fonction f qui est contractante de rapport $k<1$, alors $|x-y| = |f(x)-f(y)| \le k|x-y|$. Ceci n'est possible que pour $|x-y|=0$, donc $x=y$.

Existence. — Une récurrence facile montre que $|u_{m+1}-u_m| \le k^m|u_1-u_0|$ pour tout $m \in \mathbb{N}$, puis

$$\begin{aligned}|u_{m+p}-u_m| &\le |u_{m+p}-u_{m+p-1}|+\cdots+|u_{m+1}-u_m| \\ &\le (k^{p-1}+\cdots+k^0)|u_{m+1}-u_m| = \tfrac{1-k^p}{1-k}|u_{m+1}-u_m| \\ &\le \tfrac{1}{1-k}|u_{m+1}-u_m| \le \tfrac{k^m}{1-k}|u_1-u_0|\end{aligned}$$

pour tout $m,p \in \mathbb{N}$. La suite (u_m) est donc de Cauchy et converge puisque $(\mathbb{R}^n, |\cdot|)$ est complet. Notons $\mu := \lim u_m$ sa limite et vérifions qu'il s'agit d'un point fixe. Puisque f est contractante, elle est continue. L'équation de récurrence $u_{m+1}=f(u_m)$ donne donc

$$\mu = \lim u_{m+1} = \lim f(u_m) = f(\lim u_m) = f(\mu).$$

Vitesse de convergence. — Pour tout u_0 la suite itérative $u_{m+1}=f(u_m)$ vérifie $|u_m-\mu| \le k^m|u_0-\mu|$, donc $u_m \to \mu$. On a déjà établit la majoration $|u_{m+p}-u_m| \le \frac{k}{1-k}|u_m-u_{m-1}|$, et le passage à la limite $p \to \infty$ donne l'inégalité cherchée. □

Remarque 15. La propriété de f d'être contractante ne repose que sur la métrique de $\mathbb{R}^n$. Le théorème du point fixe et sa preuve se généralisent mot par mot à un espace métrique quelconque à condition qu'il soit complet, c'est-à-dire que toute suite de Cauchy converge.

Les espaces métriques complets sont très importants parce qu'ils permettent de *construire* certains objets comme limites de suites de Cauchy, l'existence étant assurée par l'hypothèse de complétude. Notre théorème en est un exemple fondamental aussi bien pour la théorie que pour le calcul numérique.

4.2. **Application au modèle PageRank.**

Proposition 16. *Soit $A \in \mathbb{R}^{n\times n}$ une matrice stochastique et $T\colon \mathbb{R}^n \to \mathbb{R}^n$ l'application définie par*

$$T(x) = c\varepsilon + (1-c)Ax$$

avec une constante $c \in]0,1]$. Alors l'application T est contractante de rapport $k=1-c<1$. Par conséquent, elle admet une unique mesure invariante $\mu = T(\mu)$ et, pour tout vecteur initial u_0, la suite itérative $u_{m+1}=T(u_m)$ converge vers le point fixe μ, avec la vitesse énoncée ci-dessus.

C'est cette mesure invariante μ qui nous intéressera dans la suite et que l'on interprétera comme mesure d'importance. On la calculera d'ailleurs par la méthode itérative de la proposition précédente.

Démonstration. Il suffit de prouver que T est contractante, de rapport $k = 1 - c < 1$, pour faire appel au théorème du point fixe. Regardons deux vecteurs $x, y \in \mathbb{R}^n$ et essayons de majorer $z := Tx - Ty$ en fonction de $|x - y|$. On a $z = kA(x - y)$ donc $z_i = k\sum_j a_{ij}(x_j - y_j)$ pour tout $i = 1, \ldots, n$. Ceci nous permet de calculer la norme :

$$\begin{aligned} |Tx - Ty| = |z| &= \sum_i |z_i| = \sum_i \Big| k \sum_j a_{ij}(x_j - y_j) \Big| \\ &\leq k \sum_i \sum_j |a_{ij}(x_j - y_j)| = k \sum_j \sum_i a_{ij} |x_j - y_j| \\ &= k \sum_j \Big(\sum_i a_{ij} \Big) |x_j - y_j| = k|x - y|. \end{aligned}$$

Ceci prouve que $T \colon \mathbb{R}^n \to \mathbb{R}^n$ est contractante de rapport k comme énoncé. L'application T admet donc un unique point fixe $\mu \in \mathbb{R}^n$. Remarquons finalement que le point fixe est un vecteur stochastique, c'est-à-dire qu'il satisfait $\mu_i \geq 0$ et $\sum_i \mu_i = 1$: si l'on démarre avec un vecteur stochastique u_0, alors tous les itérés u_m restent stochastiques, donc leur limite μ l'est aussi. (Exercice.) □

Remarque 17. La proposition inclut le cas trivial $c = 1$: dans ce cas $T(x) = \varepsilon$ est constante, donc $x = \varepsilon$ est l'unique point fixe. Dans l'autre extrême on pourrait considérer $c = 0$, mais $T = A$ n'est pas forcément contractante. Par exemple pour un graphe à n sommets sans arêtes entre eux, nous obtenons la matrice identité, $A = I$, qui admet tout vecteur $x \in \mathbb{R}^n$ comme point fixe. Un bon choix de c se situe donc quelque part entre 0 et 1 (voir la remarque 10).

Remarque 18. Le fait que la solution soit unique est fondamental : une fois que le modèle est établi, le théorème nous garantit une unique mesure μ, sans équivoque. Mieux encore, la suite itérative converge toujours vers μ, indépendamment du point de départ. En l'absence de toute autre information on pourra donc démarrer avec $u_0 = \varepsilon = (\frac{1}{n}, \ldots, \frac{1}{n})$ pour calculer la limite $u_m \to \mu$.

Remarquons à ce propos que Google est obligé de mettre à jour ses données régulièrement, car le web change sans cesse. Disons que Google met à jour le vecteur μ chaque semaine. Pour ce calcul, il serait maladroit de recommencer par $u_0 = \varepsilon$! Il est sans doute plus avantageux de recycler l'information déjà obtenue : on choisira $u_0 = \mu_{\text{ancien}}$, la mesure de la semaine d'avant. Ainsi peu d'itérations suffiront pour réajuster μ, en supposant que le graphe n'est que légèrement modifié.

La morale de cette histoire : l'unicité garantie par le théorème nous laisse la liberté de choisir parmi plusieurs méthodes de calcul — elles aboutissent toutes au même résultat ! On peut en profiter si l'on dispose d'informations supplémentaires, par exemple en choisissant judicieusement le point de départ de l'itération.

Remarque 19. La proposition précédente se généralise au théorème de Perron-Frobenius : si une matrice réelle A a tous ses coefficients positifs, $a_{ij} > 0$ pour $i, j = 1, \ldots, n$, alors le rayon spectral de A est donné par une valeur propre $\lambda \in \mathbb{R}_+$ et l'espace propre associé E_λ est de dimension 1. De plus, la matrice A admet un vecteur propre $v \in E_\lambda$ dont tous les coefficients sont positifs.

L'algorithme itératif correspondant est souvent appelé la « méthode de la puissance ». Il se généralise à une matrice A quelconque et permet d'approcher numériquement un vecteur propre v associé à la valeur propre λ de module $|\lambda|$ maximal, pourvu que cette valeur propre soit unique et simple.

5. IMPLÉMENTATION EFFICACE

Passons à l'implémentation de l'algorithme discuté ci-dessus. Le programme qui en résulte est plutôt court (moins de 100 lignes). Néanmoins il est important de réfléchir sur la meilleure façon de s'y prendre.

5.1. **Matrices creuses.** Rappelons que la matrice A représentant le web est très grande : en 2004 Google affirmait que « le classement est effectué grâce à la résolution d'une équation de 500 millions de variables et de plus de 3 milliards de termes. » Comment est-ce possible ?

La manière usuelle de stocker une matrice de taille $n \times n$ est un grand tableau de n^2 coefficients indexés par $(i,j) \in \{1,\dots,n\}^2$. Il est envisageable de stocker ainsi une matrice 1000×1000, c'est-à-dire un million de coefficients mais ceci est hors de question pour une matrice $n \times n$ avec $n \approx 10^6$, voire $n \approx 10^8$. L'approche naïve est donc prohibitive pour le modèle PageRank. Soulignons aussi que la méthode de Gauss, bien adaptée aux matrices de petite taille, s'avère inutilisable pour les grandes matrices. Cet algorithme effectue environ n^3 opérations, ce qui est trop coûteux si n est grand.

Dans notre cas la plupart des coefficients valent zéro car une page n'émet que quelques douzaines de liens typiquement. Dans ce cas, il suffit de stocker les coefficients non nuls, dont le nombre est d'ordre n et non n^2. Une telle matrice est appelée *creuse* (ou *sparse* en anglais). Pour des applications réalistes, il est donc nécessaire d'implémenter des structures et des méthodes adaptées aux matrices creuses. La méthode du point fixe est faite sur mesure pour ce genre d'application.

5.2. **Matrices provenant de graphes.** Pour simplifier, nous allons spécialiser notre implémentation aux matrices creuses provenant de graphes. Rappelons qu'un graphe peut commodément être codé sous forme de listes. Dans notre exemple initial cette description était :

$1 \to 2,3,4,5,6$; $2 \to 1,3$; $3 \to 1,4$; $4 \to 1,5$; $5 \to 1,3$; $6 \to 7,8,9$; $7 \to 8,1$; $8 \to 6$;
$9 \to 8,10$; $10 \to 6,11,12,13,14$; $11 \to 10,12$; $12 \to 10,13$; $13 \to 10,14$; $14 \to 10,11$.

Les pages sont numérotées par $1,\dots,n$ et, pour chaque page j, on énumère tous les liens $j \to i_1, i_2, \dots, i_\ell$ émanant de j vers les pages voisines. Notons $L_j = \{i_1, i_2, \dots, i_\ell\}$ leur ensemble et $\ell_j = |L_j|$ le nombre de liens émis par la page j. À noter que n peut être très grand alors que ℓ_j est en général très petit. Comme avant nous supposons toujours que $\ell_j \geq 1$.

Algorithme 1 Calcul efficace de $T(x) = c\varepsilon + (1-c)Ax$

Entrée: un vecteur $x \in \mathbb{R}^n$.
Sortie: le vecteur $y = Tx$.

```
Initialiser y_i ← c/n pour tout i = 1,...,n      // Ceci correspond au terme de base cε
pour j de 1 à n faire                            // On parcourt toutes les pages émettrices
   pour i ∈ L_j faire y_i ← y_i + ((1−c)/ℓ_j) x_j   // La page j pointe vers ℓ_j pages voisines
fin pour
retourner y
```

Remarque 20. La complexité de cet algorithme est optimale dans le sens que l'on traite chaque lien $j \to i$ exactement une fois : le nombre total d'opérations est donc proportionnel à $\sum_j \ell_j$. Autrement dit, si les pages émettent en moyenne $\bar{\ell} = \frac{1}{n}\sum_j \ell_j$ liens, nous avons a effectuer $n\bar{\ell}$ opérations au total, au lieu de n^2 pour une matrice dense.

À noter aussi que notre algorithme n'utilise que les deux vecteurs x et y : la matrice A ne figure pas explicitement dans l'implémentation. Effectivement, la construction explicite d'une matrice de taille $n \times n$ allouerait trop de mémoire et sera catastrophique quand n est grand !

Remarque 21. L'algorithme 1 est adapté à la structure des données choisie ci-dessus. Au lieu de stocker les liens émis $L_j = \{i \mid j \to i\}$ on pourrait stocker les liens reçus $L_i^* = \{j \mid j \to i\}$. Ceci correspond à passer de la matrice A à sa transposée A^*. Dans ce cas on inverse les deux boucles : pour i allant de 1 à n on parcourt $j \in L_i^*$ et calcule $y_i \leftarrow y_i + \frac{1-c}{\ell_j} x_j$. C'est essentiellement la même démarche, mais il faut faire un choix pour accorder structures des données et algorithmes utilisés.

Algorithme 2 Approximation de la mesure invariante $\mu = T\mu$

Entrée: la précision souhaitée $\delta > 0$.
Sortie: un vecteur x tel que $|x - \mu| \leq \delta$.

Initialiser $x_i \leftarrow \frac{1}{n}$ pour tout $i = 1, \ldots, n$ // Un vecteur stochastique initial
répéter $y \leftarrow x$, $x \leftarrow Tx$ **jusqu'à** $|x - y| \leq \frac{c\delta}{1-c}$ // Suivant le théorème du point fixe
retourner x

Exercice 22. Si vous vous intéressez à la programmation, vous pouvez essayer d'implémenter la méthode ci-dessus. Parallèlement, il sera intéressant de discuter ses aspects théoriques :

(1) Vérifier d'abord la correction des deux algorithmes. Pour le deuxième, montrer que la condition d'arrêt $|x - y| \leq \frac{c\delta}{1-c}$ garantit que le vecteur x est suffisamment proche de la limite cherchée, c'est-à-dire $|x - \mu| \leq \delta$ comme promis par la spécification de l'algorithme.

(2) Remarquons aussi que l'efficacité de cet algorithme dépend sensiblement du nombre d'itérations nécessaires. C'est ici que la vitesse de convergence, comme établie dans le théorème du point fixe, nous garantie une exécution rapide.

(3) Finalement, le fait que l'application T soit contractante nous assure aussi que notre algorithme est numériquement stable. Rappelons que par souci d'efficacité, on effectue tous les calculs avec des nombres à virgule flottante. Des erreurs d'arrondi sont donc inévitables. Fort heureusement, de telles erreurs ne sont pas amplifiées dans les calculs itératifs.

6. Quelques points de réflexion

On pourrait se contenter d'une conclusion pragmatique : « Bon, enfin ça marche. Tout le monde s'en sert. Il n'y a plus rien à ajouter. » Mais, d'autre part, l'approche est suffisamment simple et l'application importante pour se poser quelques questions.

6.1. Le modèle est-il plausible ?

La structure caractéristique d'un document *hypertexte* sont les *liens* vers d'autres documents. L'auteur d'une page web ajoute ainsi des liens vers les pages qu'il considère utiles ou « importantes ». Autrement dit, on peut interpréter un lien comme un *vote* ou une *recommandation*. Or, il ne suffit pas de compter les liens, car ils n'ont pas tous le même poids. Nous avons donc raffiné notre heuristique : une page est importante si beaucoup de pages importantes pointent vers elle. Cette définition peut sembler circulaire, mais le développement mathématique ci-dessus montre comment s'en sortir (par le théorème du point fixe).

Ainsi des millions d'auteurs de pages web lisent et jugent mutuellement leurs pages, puis leurs jugements s'expriment par les liens qu'ils mettent sur leurs pages. Le modèle de la marche aléatoire en profite en transformant l'évaluation mutuelle en une mesure globale de popularité. (Soulignons à nouveau que le surfeur aléatoire ignore le contenu et se fie uniquement à la structure des liens.)

6.2. **Hypothèses implicites.** D'après l'autoportrait de Google, « la technologie de Google utilise l'intelligence collective du web pour déterminer l'importance d'une page. » Nous venons de voir comment cette phrase peut s'interpréter mathématiquement. Une triple hypothèse y est implicite :

(1) Les liens reflètent fidèlement les appréciations des *auteurs* des pages web.

(2) Ces appréciations correspondent bien à celles des *lecteurs* des pages web.

(3) Le modèle du surfeur aléatoire les traduit fidèlement en une mesure de popularité.

En soutien de ces hypothèses, on mentionne parfois la « nature démocratique » du web pour dire que les lecteurs et les auteurs ne font qu'un et que l'échange des informations est libre. C'est une idéalisation de moins en moins plausible, surtout quant à l'aspect commercial du web. En 1993 seul 1,5% des sites web étaient dans le domaine `.com`, en 2003 ils représentaient plus de 50% du web et la fréquentation des pages devrait donner des proportions similaires.

6.3. **Descriptif ou normatif ?** Le statut de Google lui-même a complètement changé :

Google se veut descriptif: Au début de son existence, Google se voulait un outil purement *descriptif* : si une page est importante, alors elle figure en tête du classement.

En réalité il est devenu normatif: Aujourd'hui, son écrasant succès fait de Google une référence normative : si une page figure en tête du classement, alors elle est importante.

À titre d'illustration, citons un exemple devenu classique. Le mathématicien français Gaston Julia, né le 3 février 1893, devint célèbre pour ses contributions à la théorie de fractales, largement popularisée par son élève Benoît Mandelbrot depuis les années 1970. Pour son anniversaire le 3 février 2004, la page d'accueil de Google montrait une variation fantaisiste du logo usuel. Un clique dessus lançait la recherche d'images associées aux mots-clés « Julia » et « fractale ». Deux des pages en tête du classement étaient hébergées à un institut de l'université de Swinburne, à Melbourne en Australie. Comme tous les jours, des millions d'internautes ont visité la page de Google et, ce jour-là, une certaine fraction a suivi le lien du logo, pour tomber sur la page à Swinburne. Ce trafic soudain a suffit pour submerger le serveur australien, qui rendit l'âme aussitôt. Les images fractales durent être déplacées et une page explicative fut mise à la place [4]. Elle conclut par une question mémorable (d'après Job **1** 21) :

> Google giveth, and Google taketh away, blessed be Google ?
> [Google avait donné, Google a repris, que le nom de Google soit béni ?]

Bien que le trafic internet ne soit pas toujours une bénédiction, la plupart des webmestres seraient ravis d'accueillir des foules d'internautes sur leur site car la popularité peut potentiellement se transformer en bénéfice. Cet aspect rend l'évaluation des pages web encore plus difficile : comme l'approche et l'importance de Google sont mondialement connues, les liens s'utilisent sans doute différemment à nos jours. Après avoir compris l'algorithme de Google, les concepteurs de sites web pourraient appliquer cette connaissance afin d'améliorer leur classement…

6.4. **Peut-on manipuler Google ?** Pour des sites web commerciaux, l'optimisation de leur classement est devenue un enjeu important. Évidemment, le fournisseur d'un service commercial souhaite que son site soit le plus visité possible et ceci passe par Google : des millions de clients potentiels utilisent Google et suivent typiquement les liens en tête du classement. Comment améliorer son classement, son importance calculée par Google ? Voici ce qu'en dit l'entreprise Google :

> Les méthodes complexes et automatiques utilisées par les recherches Google rendent quasi impossible toute manipulation humaine des résultats. (…) Google ne pratique pas la vente des positions dans ces résultats ; autrement dit, il n'est pas possible d'acheter une valeur PageRank supérieure à la réalité du Web.

Pourtant, afin d'améliorer son classement par Google, il suffit d'attirer des liens, de préférence ceux émis par des pages importantes et il vaut mieux en émettre très peu, de manière bien choisie.

Exercice 23. La stratégie la plus évidente (et la plus honnête) pour attirer des liens est d'offrir des informations de qualité. Par exemple, si après lecture vous trouvez que cet article le mérite, faites-y pointer un lien Comment fonctionne Google ? depuis votre page web. Vous ferez ainsi monter son classement PageRank. À vérifier au bout de quelques semaines, après mise à jour de la base de données de Google.

Un grand merci à tous ceux qui ont déjà participé à cette expérience pratique. Depuis fin 2007 le présent document arrive en tête du classement pour la requête « comment fonctionne Google ».

Ces stratégies et astuces sont elles-mêmes devenues un domaine très actif, dit « *search engine optimization* » (SEO). Ceci confirme en particulier que l'omniprésence de Google change l'utilisation des liens par les auteurs… ce qui remet en question l'hypothèse à la base même du modèle.

Exercice 24. Qu'en pensez-vous : que peut-on faire pour améliorer son classement ? Avec votre implémentation expérimentale de l'algorithme PageRank, vous pouvez tester vos conjectures sur des exemples concrets. Vous pouvez aussi regarder ce qu'en disent des experts : cherchez par exemple « *google PageRank algorithm* » ou « *search engine optimization* » et vous verrez.

6.5. **Comment évolue Google ?** L'algorithme de base que nous venons de décrire fut mis en œuvre en 1998 et reste, semble-t-il, le fondement de l'efficacité légendaire de Google. (D'ailleurs, la méthode a été brevetée par l'université de Stanford en 2001, ainsi qu'une version raffinée en 2004, et le nom « PageRank » est une marque déposée de Google Inc. [3].)

La méthode actuellement utilisée a sans doute été adaptée et peaufinée au fil des années, afin de rendre le classement encore plus utile, c'est-à-dire plus proche des attentes des utilisateurs, et plus robuste contre des tentatives de manipulations. Contrairement à l'algorithme de base, toutes les modifications ultérieures restent un secret de l'entreprise Google.

Toujours est-il que les webmestres les plus inventifs arrivent souvent à influencer le classement en leur faveur pour se positionner sur les premières pages des résultats. En réaction, Google est obligé d'améliorer son algorithme pour rattraper les tricheurs, au moins les plus flagrants. Bref, c'est l'habituelle course du gendarme et du voleur, mais typiquement Google s'en sort bien.

Effectivement, Google a tout intérêt à maintenir la bonne qualité de ses résultats afin de défendre sa popularité qui, rappelons-le, est la source de ses revenus. Si l'on veut y voir un aspect positif, on pourrait dire que cette éternelle compétition fait évoluer les moteurs de recherche.

6.6. **Where next ?** Il n'est pas difficile d'imaginer des variantes et possibles améliorations, mais il est souvent délicat de les mettre en œuvre concrètement. À titre d'exemple, reprenons le modèle probabiliste de Google, formalisé par l'application $T(x) = c\varepsilon + (1-c)Ax$. La question de base reste l'évaluation des liens : par défaut la construction de la matrice A traite tous les liens émanant d'une page j comme équivalents, indépendamment de la requête, et cela donne déjà de bons résultats.

Le modèle pourrait être amélioré si l'on comprenait mieux quels liens sont pertinents à une requête donnée, afin d'adapter la matrice A. Autant que l'on puisse dire, Google implémente partiellement cette idée. Évidemment le modèle du surfeur aléatoire n'est qu'une première approximation. La prochaine étape serait donc de modéliser un surfeur « intelligent ».

Un autre sujet est la « personnalisation » : on pourrait par exemple remplacer ε, l'équiprobabilité sur toutes les pages, par une autre distribution δ, un « profil » dépendant des préférences de l'utilisateur. Ainsi la marche aléatoire serait l'itération de l'application $T(x) = c\delta + (1-c)Ax$. Ici le profil $\delta = (\frac{1}{n}, \ldots, \frac{1}{n})$ veut dire « aucune préférence ». Une distribution δ concentrée sur des sites à thèmes scientifiques, par exemple, en serait un autre profil, plus spécifique. La mesure invariante dépend évidemment du profil spécifié, et mettra plus de poids sur des pages proches du profil.

La difficulté réside dans la construction de (quelques) profils raisonnables, c'est-à-dire appréciés par les utilisateurs. De nouveau, un affinage itératif semble logique, basé sur des réactions des *utilisateurs* cette fois-ci. L'aspect prometteur de cette approche est qu'elle prend en compte simultanément les appréciations des auteurs *et* des lecteurs des pages web.

Vous voyez, il ne manque pas d'idées à explorer ! Vue l'importance du sujet, ce domaine est devenu extrêmement actif (et lucratif) depuis une dizaine d'années et réunit parfois recherches fondamentales et appliquées en mathématiques et informatique. Ne mentionnons ici que les travaux de Jon Kleinberg [5] qui a développé des méthodes théoriques et algorithmiques plus raffinées, ce qui lui valut le prix Nevanlinna en 2006. Mais cela serait une autre histoire...

Exercice 25. Trouvez une meilleure méthode pour extraire de l'information du web et devenez riche et/ou célèbre. Sachez à ce propos que l'Institut Fourier accepte les dons.

Remerciements. Je tiens à remercier mon collègue Tanguy Rivoal pour ses conseils linguistiques et surtout son encouragement à publier ces notes de cours sous forme d'article de vulgarisation. Une version abrégée de cet article est paru dans *Quadrature*, no. 68, avril 2008.

RÉFÉRENCES

[1] S. Brin, L. Page : *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, Stanford University, 1998. (Pour trouver ce texte en ligne cherchez-le avec Google.)

[2] D. Vise, M. Malseed : *Google story*, Dunod, Paris, 2006.

[3] Wikipedia : *PageRank*, `en.wikipedia.org/wiki/PageRank` et `fr.wikipedia.org/wiki/PageRank`. (La page francophone attend toujours une rédaction digne du sujet.)

[4] *The power of Google*, `local.wasp.uwa.edu.au/~pbourke/fractals/quatjulia/google.html`

[5] J.M. Kleinberg : *Authoritative sources in a hyperlinked environment*, `www.cs.cornell.edu/home/kleinber/auth.pdf`.

INSTITUT FOURIER, UNIVERSITÉ GRENOBLE I, FRANCE
URL: `www-fourier.ujf-grenoble.fr/~eiserm`
E-mail address: `Michael.Eisermann@ujf-grenoble.fr`