

1 Algorithme PageRank de Google

L'histoire de Google a tous les traits d'une histoire de succès typiquement américaine. Une jeune entreprise start-up dans un garage ; un succès fulgurant auprès du public ; et une entrée triomphale des fondateurs dans la liste des milliardaires quelques années plus tard.

En 1995, deux étudiants de Stanford, Sergey Brin (23 ans) et Lawrence Page (à peu près le même âge je pense), se rencontrent pour discuter les problèmes des moteurs de recherche. Ils comprennent très vite la difficulté principale. Voilà ce qui se passe d'habitude pendant une séance de recherche :

- on lance une requête auprès d'un moteur de recherche – quelque chose comme “dessins d'enfants” par exemple ;
- on reçoit quelque 74 000 liens correspondants ;
- et on n'en regarde jamais plus que la première trentaine.

Il est donc primordiale d'afficher les liens plus importants d'abord. Pour cela, il faut ranger les pages web.

Les deux jeunes hommes possèdent deux atouts principaux pour relever la tâche : ils sont informaticiens ; et ils connaissent la notion de vecteur propre. Il n'est pas à exclure même qu'ils connaissent le théorème de Perron–Frobenius : le père de Sergey Brin est un mathématicien à Maryland ; cela peut aider...

Une page est déclarée *importante* s'il y a beaucoup d'autres pages importantes qui pointent vers elle. La nature apparemment “circulaire” de cette définition s'explique exactement par la notion de vecteur propre. Voici une citation du premier article de Brin et Page [1] :

We assume page A has pages $T_1 \dots T_n$ which point to it. The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. Also $C(A)$ is defined as the number of links going out of page A . The PageRank of a page A is given as follows :

$$PR(A) = (1 - d) + d(PR(T_1)/C(T_1) + \dots + PR(T_n)/C(T_n)).$$

PageRank or $PR(A)$ can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web.¹

Traduisons.

¹Cette dernière phrase, affirmant que PR est le vecteur propre de la matrice du web, est en fait inexacte, comme on le verra plus tard.

Tout d'abord, le graphe de web est orienté, et la notion habituelle de la matrice d'adjacence ne nous convient pas. Soit $A = (a_{ij})$ la matrice telle que $a_{ij} \neq 0$ si et seulement si il existe un arc du sommet j vers le sommet i . Prêtez attention à la direction : l'arc va de j à i et non pas à l'envers. Dans ce cas-là on pose

$$a_{ij} = \frac{1}{\#(\text{arcs sortants du sommets } j)}.$$

Ainsi, la somme des éléments de chaque colonne de A est égale à 1. (En fait, il y a des pages qui n'ont pas de liens sortants ; pour celles-ci, la somme des éléments de la colonne correspondante vaut 0. Mais nous allons pour l'instant négliger cette circonstance.) Finalement, le rang r_i du sommet i est calculé comme

$$r_i = (1 - d) + d \cdot \sum_j a_{ij} r_j,$$

ou, sous forme vectorielle,

$$\vec{r} = (1 - d) \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + d \cdot A \vec{r}. \quad (1)$$

Si nous oublions temporairement l'histoire du "damping factor" d , nous obtiendrons l'équation

$$\vec{r} = A \cdot \vec{r};$$

le vecteur \vec{r} de PageRank n'est donc rien d'autre que le vecteur propre de la matrice A qui correspond à la valeur propre égale à 1. Mais pourquoi 1 est une valeur propre ? Qu'est-ce que nous dit la théorie sur ce sujet ?

Théorème 1.1 (Perron–Frobenius) *Soit A une matrice dont tous les éléments vérifient $a_{ij} \geq 0$. Alors :*

1. *La valeur propre de A du module maximal λ_{\max} est réelle et positive.*
2. *Parmis les vecteurs propres correspondant à λ_{\max} il existe un dont toutes les composantes sont positives ou nulles.*
3. *"Normalement", c'est-à-dire, sauf pour des exemples spécialement conçus, la valeur propre λ_{\max} est simple, et le module de toute autre valeur propre est strictement inférieur à λ_{\max} . (Dans ce cas-là, le vecteur propre correspondant est aussi unique.)*

4. Si les sommes de toutes les lignes de A sont égales, alors λ_{\max} est égale à cette somme commune. La même chose si les sommes de toutes les colonnes de A sont égales.

La dernière proposition mérite un commentaire. Elle est évidente pour les *lignes* à sommes égales. En effet, notons cette somme k . Alors le vecteur propre correspondant est le vecteur dont toutes les composantes sont égales à 1 : il est clair que multiplier ce vecteur par la matrice A revient à le multiplier par k . D'autre part, aucune valeur propre ne peut être supérieure à k . En effet, soit \vec{x} un vecteur, et soit x_i sa composante maximale ; étant donné que la i -ème composante de $A\vec{x}$ est une "somme pondérée" de quelques composantes de \vec{x} , avec le poids total k , il est clair que x_i ne peut pas être multipliée par un nombre supérieur à k .

Pour les *colonnes* à sommes égales, trouver le vecteur propre correspondant à k n'est pas évident ; c'est d'ailleurs ce que nous cherchons. Par contre, le $\det(A - \lambda I)$ reste le même quand on remplace une matrice par sa transposée ! Par conséquent, le spectre reste inchangé.

Ce raisonnement montre aussi que si, au lieu d'être toujours égales à 1, les sommes des éléments de certaines colonnes de A sont nulles, alors λ_{\max} risque de devenir inférieure à 1. Dans ce cas-là, le "simple algorithme itératif" évoqué par les auteurs peut conduire à un vecteur de PageRank qui tend vers zéro. Le facteur d est introduit, entre autre, pour pallier à ce défaut. Selon la définition (1), la valeur minimale possible de PageRank est égale à $1 - d$, soit, en pratique, 0.15.

Maintenant, en revenant à la définition principale, on doit se poser la question suivante : est-il vrai que les itérations successives de la fonction affine

$$\vec{x} \mapsto (1 - d) \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + d \cdot A\vec{x}$$

tendent vers la solution de l'équation (1) ? Ainsi on passe à une autre théorie, celle des itérations de fonctions, ou des *systèmes dynamiques discrets*.

* * *

L'article de Rogers [2] contient quelques exemples concrets, un logiciel interactif pour calculer le PageRank pour des petits graphes, ainsi qu'une discussion du comportement de la solution en fonction de la structure du graphe, y compris des méthodes d'augmentation artificielle du PageRank

d'une page – par exemple, en créant plusieurs autres pages, du contenu vide et dont le seul objectif serait de pointer vers la page principale. (L'auteur est quand même convaincu que la version actuelle de Google possède des remèdes contre cela.)

Pour finir, je cite deux premiers paragraphes de cet article :

Page Rank is a topic much discussed by Search Engine Optimisation (SEO) experts. At the heart of PageRank is a mathematical formula that seems scary to look at but is actually fairly simple to understand.

Despite this many people seem to get it wrong! In particular "Chris Ridings of www.searchenginesystems.net" has written a paper entitled "Page Rank Explained : Everything you've always wanted to know about PageRank", pointed to by many people, that contains a fundamental mistake early on in the explanation! Unfortunately this means some of the recommendations in the paper are not quite accurate.

Apparemment, les vecteurs propres ne sont pas encore devenus un lieu commun dans la communauté des gens qui travaillent sur le web.

Références

- [1] **Sergey Brin, Lawrence Page.** The anatomy of a large-scale hyper-textual web search engine.
<http://www-db.stanford.edu/~backrub/google.html>
- [2] **Ian Rogers.** Page rank explained. The Google pagerank algorithm and how it works. (May 2002)
<http://www.iprcom.com/papers/pagerank>