Google, les graphes et son algorithme de classement des pages web

25 mars 2014

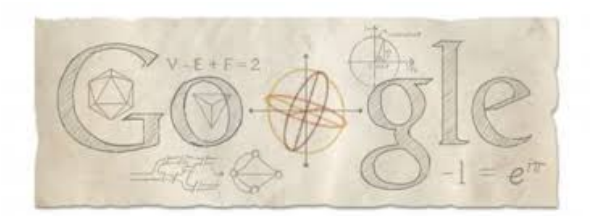


Table des matières

1 Introduction

Google est le moteur de recherche sur internet le plus utilisé dans le monde. En quelques années, il a supplanté les moteurs de recherche les plus populaires de l'époque comme Altavista ou Yahoo. Le mot "googler" a même fait son entrée dans le Petit Larousse !

Google a été développé par Sergei Brin et Larry Page au milieu des années 1990 alors qu'ils étaient tous les deux doctorants en informatique à l'université de Stanford.

Ce qui a fait son succès ? Les résultats présentés par Google étaient beaucoup plus satisfaisants pour l'utilisateur que ceux donnés par ses concurrents et ceci en raison d'un algorithme de classement des pages web révolutionnaire. L'algorithme utilisé actuellement par Google est bien évidemment tenu secret mais l'idée initiale a été publiée dans un article. C'est donc cette idée que nous allons explorer dans ce projet en nous basant sur les articles de M. Eisermann et M. Rigo.

2 Comment classer des pages web ?

La conception d'un moteur de recherche repose sur de nombreux éléments dont une base de données la plus grande possible contenant les pages web et un procédé d'indexation suivant des mots clefs. C'est un travail en soi qui demande une actualisation constante. On utilise notamment pour cela des "robots" appelés *web-crawler* qui parcourent de manière infatigable le réseau. Nous n'allons pas abordé ce problème et supposer que cette base de données existe. Lorsque vous allez taper un mot dans Google il va donc chercher dans sa base de données les pages dont l'indexation fait apparaître ce mot. Le problème qui se pose alors va tout de suite se voir avec des exemples :

Si vous tapez "Mathématiques" vous obtenez environ 9 270 000 résultats en 0,20 secondes.

Si vous tapez "Daft Punk" vous obtenez environ 82 600 000 résultats en 0,17 secondes !

Je ne connais pas le nombre maximal de pages qui peut être relié à un mot clef mais en 2013 la taille de la base de données était estimée à 30 000 milliards d'URL référencées !

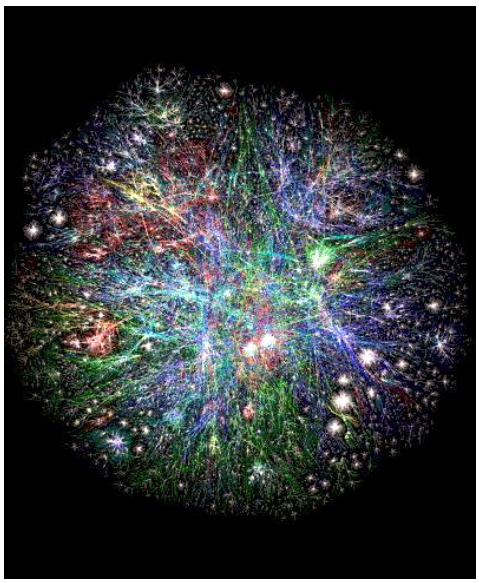
Juste pour s'amuser la lettre "a" renvoie à environ 15 600 000 000 résultats (0,23 secondes) !

On le voit tout de suite, le problème va être de classer ces quantités énormes de pages web et de le faire de sorte que ce classement réponde dans les premières réponses à ce qu'attend l'utilisateur. C'est ce problème déjà conséquent que nous allons aborder dans la suite.

3 Graphes orientés et structure du web

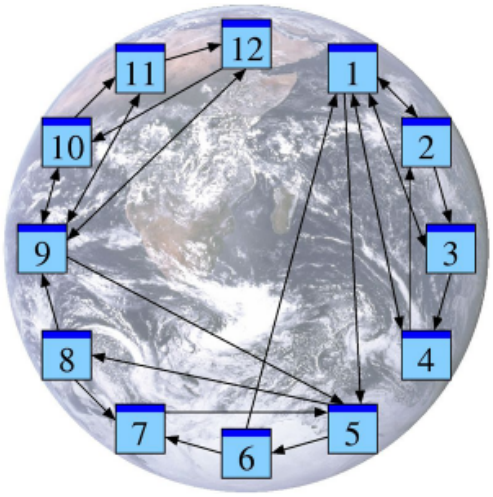
Une des difficultés du problème de classement est lié au web lui-même. Il ne semble pas vraiment structuré et ressemble plus à un organisme en évolution qu'à une structure figée. Des pages apparaissent et disparaissent chaque jour. Imaginer un algorithme de classement devient alors difficile. On peut néanmoins lui associer un objet mathématique : un graphe.

La raison tient au fait que le web est un immense hypertexte qui suivant la page de Wikipedia est "un système contenant des nœuds liés entre eux par des hyperliens permettant de passer automatiquement d'un nœud à un autre. Un document hypertexte est donc un document qui contient des hyperliens et des nœuds."



Si on néglige dans un premier temps le contenu des pages, le web est en terme mathématique un graphe dont chaque page web j est un sommet et chaque lien de la page j vers la page i est une arrête.

Dans la suite on numérote les pages web par 1, 2, ..., n et nous ferons une flèche entre le noeud i et le noeud j si la page i cite la page j. Un tel graphe est donné par l'exemple suivant :



Les arrêtes sont orientées : si on a $j \to i$ on n'a pas forcément $i \to j$.

De manière formelle un graphe orienté est la donnée d'un ensemble de sommets V et d'un ensemble d'arcs $E = \{e_1, \ldots, e_m\}$. Un arc est un couple ordonné de sommets. Lorsque $e = (v_i, v_j)$ on dit que l'arc e va du sommet v_i au sommet v_j.

Exercice 1.

Tracer le graphe dont les sommets sont les nombres entiers de 1 à 12 et dont les arcs représentent la relation "être diviseur de".

3.1 Autre représentation d'un graphe orienté : la Matrice d'adjacences

On peut représenter un graphe orienté par une matrice d'adjacences. Une matrice $(n \times m)$ est un tableau de n lignes et m colonnes. Un couple (i, j) désigne l'intersection de la ligne i et de la colonne j.

Dans une matrice d'adjacences, les lignes et les colonnes représentent les sommets du graphe. Un "1" à la position (i, j) signifie qu'un arc part de i pour rejoindre j.

Exercice 2.

Écrire la matrice d'adjacences du graphe construit à l'exercice 1.

Exercice 3.

Écrire la matrice d'adjacences du graphe illustrant le web.

La matrice d'adjacences contient en fait toute l'information du graphe. On peut donc reconstruire le graphe à partir de la donnée de sa matrice.

Exercice 4.

Construire le graphe orienté dont la matrice d'adjacences est

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

4 Graphes et méthodes de classement des pages web

Classer les pages web revient donc à structurer d'une certaine façon un graphe orienté en attachant un poids à chaque sommet du graphe. Le choix de ce poids doit évidemment refléter des propriétés du graphe mais aussi avoir une interprétation claire au niveau du domaine d'application, ici les pages web.

4.1 Premier modèle : comptage naïf et degré intérieur d'un graphe orienté

La première "intuition" pour organiser les pages web est la suivante : une page web est importante si beaucoup de pages web y font référence.

Cela paraît effectivement être un bon argument. La page recevant le plus de liens pour un sujet donné serait considérée comme la page de référence ou l'incontournable du sujet.

On associe donc à chaque page i le nombre comptant les pages qui y renvoient. Autrement dit, on vient de définir l'importance μ_i de la page web i par la somme

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

Exercice 5.

Calculer le poids de chaque sommet du graphe de l'exercice 3.

En fait, ce poids est une quantité bien connue en théorie des graphes. On l'appelle le **degré intérieur** d'un sommet i. Il représente le nombre d'arcs ayant i pour extrémité finale et se note $d_-(i)$.

4.2 Second modèle : comptage pondéré et degré extérieur d'un graphe orienté

Le coefficient précédent n'est pas très bon. Mis à part les problèmes de manipulations de ce coefficient qui sont ici très simples, il y a aussi le fait que toutes les pages web ont le même

poids, la même importance dans le comptage. Or, certaines pages émettent des quantités importantes de liens. Ceux-ci semblent donc moins spécifiques pour le mot clés considéré. On va donc pondérer son poids par le nombre de liens qui est émis par la page.

Pour cela nous définissons le degré extérieur $d_+(i)$ d'un sommet i en comptant le nombre d'arcs ayant pour extrémité initiale l'arc i.

Exercice 6.

Calculer le degré extérieurs des sommets du graphe de l'exercice 3.

Le nouveau coefficient s'écrit donc

$$\mu_i = \sum_{j \to i} \frac{1}{d_+(j)}.$$

Exercice 7.

Calculer le poids de chaque sommet du graphe de l'exercice 3.

4.3 Troisième modèle : comptage récursif

Ce qui n'est pas pris en compte finalement dans les comptages précédents est l'importance des pages qui citent la page considérée. Intuitivement, nous aurions envie de dire qu'une page est importante si des pages réputées importantes y font référence. C'est en gros le phénomène de reconnaissance du savoir par les pairs qui se traduit ainsi sur les pages web. Le problème est évidemment que le calcul de ce coefficient devient alors beaucoup plus compliqué. La formule est, elle, facile à écrire.

$$\mu_i = \sum_{j \to i} \frac{\mu_j}{d_+(j)}.$$

Exercice 8.

Calculer le poids de chaque sommet du graphe de l'exercice 1. Qu'observez-vous ?

Exercice 9.

Calculer le poids de chaque sommet du graphe de l'exercice 3.

Pour résoudre cet exercice vous avez sans doute éprouvé quelques difficultés. Je vous laisse imaginer le même problème avec les 30 000 milliards de pages référencées par Google. Même avec un ordinateur la situation devient vite impraticable.

5 Un peu d'algèbre linéaire

En fait l'équation précédente peut se réécrire à l'aide de la **matrice d'adjacences pondérée** par les degrés extérieurs. Elle se construit exactement de la même manière que la matrice d'adjacences :

Sur la ligne i et la colonne j de la matrice, le coefficient $m_{i,j}$ est donné par $1/d_+(j)$ si $j \to i$.

Exercice 10.

Construire la matrice d'adjacences du graphe de l'exercice 1 et de l'exercice 3.

La matrice ainsi obtenue a des propriétés intéressantes que nous interpréterons plus loin :
- Tous ces coefficients sont positifs.
- Si on additionne les coefficients d'une colonne non identiquement nulle on obtient toujours 1, autrement dit

$$\sum_{i=1}^{n} m_{i,j} = 1.$$

Exercice 11.

Le démontrer !

Lorsque ces propriétés sont vérifiées pour toutes les colonnes d'une matrice on dit que la matrice est **stochastique**.

5.1 Calcul matriciel

Il existe une manière utile de coder les systèmes linéaires. Nous allons seulement donner le cas de matrice de dimension 3 mais on peut généraliser ce procédé en dimension quelconque. Considérons donc une famille d'inconnues x, y, z et fabriquons le vecteur $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$. Alors les quantités suivantes

$$\begin{cases} ax + by + cz, \\ dx + ey + fz, \\ gx + hy + iz, \end{cases}$$

peuvent s'écrire sous la forme

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} ax + by + cz \\ dx + ey + fz \\ gx + hy + iz \end{pmatrix}$$

On multiplie chaque ligne de la matrice par la colonne du vecteur en respectant l'ordre des termes. On peut de cette manière réécrire le système de détermination du poids des pages sous la forme

$$M.\mu = \mu,$$

où μ représente le vecteur dont les composantes sont les μ_i et M la matrice d'adjacences pondérée du graphe.

Tout système linéaire admet une représentation sous cette forme dite **forme matricielle**.

Exercice 12.

Écrire le système d'équation ($\star$) sous forme matricielle.

Muni de cette notation, notre problème de recherche des μ_i revient à la recherche d'un vecteur μ solution de $M\mu = \mu$.

Plusieurs problèmes mathématiques importants se posent naturellement :
- Existe-t-il une solution ?
- Cette solution est-elle unique ?
- Peut-on la calculer explicitement ? algorithmiquement ?

Nous allons explorer tout d'abord le problème de l'existence.

5.2 Existence d'une solution au problème de classement ?

5.2.1 0 or not 0 ?

Remarquons tout d'abord que le vecteur nul est toujours solution de notre système. Évidemment, un vecteur nul ne permet pas de faire un classement des pages et il faut donc espérer que ce n'est pas la seule solution.

Malheureusement, cela arrive parfois ! Le graphe de l'exercice 1 en fournit un exemple. C'est ce que vous avez dû voir en essayant de calculer les poids dans l'exercice 7. On peut aussi voir que le problème provient essentiellement des pages qui n'émettent pas de liens et créaient une colonne nulle dans la matrice.

5.2.2 Multiplicité des solutions

Il peut aussi y avoir plusieurs solutions non nulles possibles. Du coup, comment choisir le bon classement ? Pour avoir un exemple, vous pouvez considérer le graphe suivant :

5.2.3 Un cas qui marche...

Le graphe de l'exercice 3 fonctionne. On peut résoudre le système et on obtient une unique solution non nulle

$$\mu = (5, 2, 2, 2, 2, 6, 2, 4, 2, 5, 2, 2, 2, 2).$$

Ici tout se passe bien.

Avant de savoir comment nous allons résoudre les problèmes d'existence et d'unicité, nous allons développer une **interprétation probabiliste** du vecteur μ.

6 Marche aléatoire sur la toile et mesures invariantes

Cette section va en fait justifier le terme de **stochastique** employé tout à l'heure au sujet de la matrice d'adjacences pondérée.

6.1 Marche aléatoire

Imaginons un "surfeur" du net qui clique sans regarder sur les liens disponibles à partir d'une page. Sur la page i, il a $d_+(i)$ possibilités de se déplacer vers une page j connectée à i. Si la page j est connectée à i, la probabilité d'aller de la page i à la page j est de $1/d_+(i)$ et 0 si la page j n'est pas reliée à i. Ces coefficients sont exactement ceux que l'on retrouve dans la matrice d'adjacences pondérée du graphe ! Autrement dit, ces coefficients peuvent s'interpréter comme le **déplacement aléatoire d'un surfeur sur le web**.

Il faut noter que ce n'est pas le contenu qui est important ici mais la structure du graphe.

6.2 Mesure invariante

Nous allons maintenant interpréter plus précisément le déplacement de notre surfeur du net.

On se donne tout d'abord un vecteur $x = (x_i)_{i=1,\dots,n}$ tel que $x_i \geqslant 0$ et $\sum_{i=1}^{n} x_i = 1$. Ce type de vecteur est dit stochastique ou une **mesure de probabilité** sur les pages $1, \dots, n$. On interprète x_i comme la probabilité de se trouver à la page p_i.

Nous allons d'abord faire un pas sur notre graphe à partir de la configuration des probabilités donnée par x. Supposons que notre surfeur parte de la page j avec la probabilité x_j. La probabilité d'aller à la page i, $j \to i$, est $m_{i,j}x_j$. On en déduit donc la probabilité d'arriver sur la page i par n'importe quel chemin est

$$y_i = \sum_{j=1}^{n} m_{i,j}x_j.$$

Autrement dit, lorsqu'on fait un pas de marche aléatoire sur le graphe, cela revient à calculer le vecteur $y = (y_i)$ à partir de x via la matrice d'adjacences pondérée. En écriture matricielle cela s'écrit :

$$y = M\,x.$$

On peut voir cette action comme le fait de prendre l'image de x par l'application linéaire $T(x) = M\,x$.

Lorsque le promeneur va continuer sa marche, nous allons appliquer T mais cette fois à y et ainsi de suite. Nous sommes donc conduits à étudier la suite de vecteurs suivant :

$$\nu_{n+1} = T(\nu_n),$$

avec $\nu_0 = x$.

C'est donc une suite récursive.

Supposons un instant que cette suite converge. Sa limite μ va satisfaire la relation

$$\mu = T(\mu).$$

Cette équation s'écrit encore $\mu = M\,\mu$ qui est exactement la forme matricielle de l'équation déterminant les poids μ_i des pages.

Le vecteur μ est appelé **mesure invariante** ou **mesure d'équilibre**. C'est par construction l'état vers lequel va converger notre marche aléatoire si on attend assez longtemps. Les pages ayant des μ_i élevés seront donc a priori celles sur lesquelles on a le plus de chance de tomber au bout d'un certain temps. Cela correspond bien à l'idée de popularité d'une page que nous cherchons à modéliser.

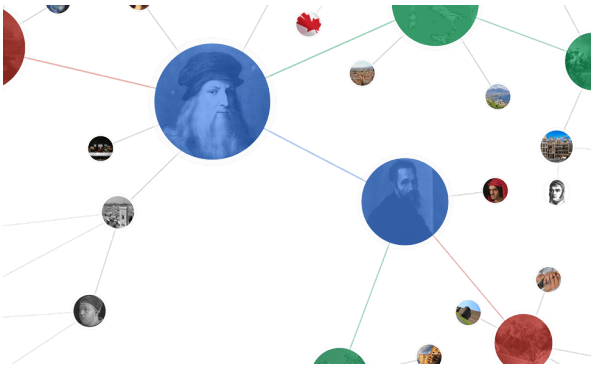
Exercice 13.

Faire les premiers pas de la marche aléatoire sur le graphe de l'exercice 3. Programmer la suite et tester la convergence de la suite. Comparer le résultat obtenu avec celui de l'exercice 8.

Nous venons donc de faire d'une pierre deux coups. Non seulement il est possible a priori de calculer la solution du système linéaire algorithmiquement sans résoudre le système linéaire mais en procédant par itération de T à partir d'une configuration arbitraire de poids x.

7 Développement futur : le "knowledge graph"

Une fois le graphe des pages web formé, on peut chercher à décrire certains chemins qui vont aider une personne dans ses recherches autour d'un sujet en lui proposant les liens pertinents par rapport à sa requête initiale. L'idée de construire de cette façon une vaste encyclopédie du savoir avec liens interactifs est maintenant possible en utilisant les ressources de Google et de Wikipedia. C'est le "Knowledge graph" actuellement en cours de développement.



8 Références

J'ai utilisé les articles de vulgarisation de Michael Eisermann sur l'algorithme PageRank de Google et qui sont tous disponibles sur sa page web :

M. Eisermann, L'algorithme Pagerank de Google : une promenade sur la toile, Journées APMEP à Rouen, Octobre 2009.
M. Eisermann, Comment fonctionne Google ? 15 p., 2009.

Il y a aussi l'article de Michel Rigo de l'Université de Liège qui donne une présentation très didactique de l'algorithme PageRank à l'aide des graphes et des matrices :

M. Rigo, Google, graphes et matrices...la matrice cachée de Google, Printemps des Sciences 2008.

Dans l'exercice 4, vous devriez obtenir le graphe suivant :