Agrégation externe de mathématiques, session 2014
Épreuve de modélisation, option B : Calcul Scientifique

(*public2015-B2*)

Résumé : On s'intéresse à certains modèles et algorithmes utilisés par les moteurs de recherche sur internet pour évaluer la pertinence des résultats d'une recherche et permettre ainsi d'afficher les résultats par ordre d'importance. Les méthodes employées sont issues de l'algèbre linéaire et peuvent présenter des interprétations en terme de théorie des graphes.

Mots clefs : Algèbre linéaire. Éléments propres de matrices.

► *Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Des suggestions de développement, largement indépendantes les unes des autres, vous sont proposées en fin de texte. Vous n'êtes pas tenu(e) de les suivre. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury appréciera que la discussion soit accompagnée d'exemples traités sur ordinateur.*

1. Introduction

1.1. *Contexte*

La plupart des moteurs de recherche sur internet fonctionnent à peu près de la façon suivante :

(1) L'utilisateur donne une liste de mots-clés.

(2) Le moteur détermine de façon assez automatique et peu précise les pages qui contiennent un des mots-clés (ou tous les mots-clés selon les options de recherche).

(3) Parmi toutes les pages ainsi obtenues, le moteur détermine un degré de pertinence afin d'afficher en premier les résultats les plus significatifs pour l'utilisateur. En général, seuls les 10 ou 20 premiers résultats sont affichés sur la première page. Ceci est crucial car dans 90% des cas, l'utilisateur se contente de la première page de résultats.

La qualité d'un moteur de recherche réside donc en particulier dans sa capacité à estimer de façon automatique et efficace la pertinence des pages obtenues. Le but de ce texte est de présenter des algorithmes utilisés en pratique pour répondre à cette question à partir de l'analyse des liens entre les pages.

1.2. *Notations*

On note $(P_i)_{i\in\{1,\dots,N\}}$ les différentes pages sélectionnées par le moteur à partir de la demande de l'utilisateur par les étapes (1) et (2) ci-dessus. Bien entendu, pour l'illustration des algorithmes décrits dans ce texte, on prendra N relativement petit (de l'ordre de 10), mais on gardera à l'esprit qu'en pratique, le nombre N peut avoisiner 10^6.

Comme chacun sait, le WEB est constitué d'un ensemble de liens (dit *hypertextes*) entre les différentes pages. Ceci incite naturellement à représenter celui-ci comme un graphe orienté G dont les sommets sont les pages P_i et les arêtes construites selon les liens. Autrement dit, le graphe G admet une arête de P_i vers P_j si et seulement s'il existe un lien de la page P_i vers la page P_j. Chaque arête a un poids égal à 1.

On dit qu'une page P_i est une racine du graphe si aucune autre page P_j ne pointe vers P_i. On dit qu'une page P_i est une feuille du graphe si elle ne pointe vers aucune autre page P_j.

On associe à ce graphe G, la matrice d'adjacence $A = (a_{ij})_{i,j}$ où

$$(1)\qquad \begin{cases} a_{ij} = 0 & \text{s'il n'y a pas de lien de } P_i \text{ vers } P_j, \\ a_{ij} = 1 & \text{s'il y a un lien de } P_i \text{ vers } P_j. \end{cases}$$

De façon réciproque, étant donnée une matrice carrée à coefficients positifs B, on peut lui associer un graphe orienté G_B dont les arêtes sont repérées par les coefficients non nuls de la matrice, qu'on appelle alors *poids* de l'arête.

2. Le modèle liens/contenu

2.1. *Principe*

Ce modèle est basé sur la constatation qu'on peut *grosso modo* distinguer deux types de pages WEB : les pages de *contenu* dans lesquelles se trouvent les informations proprement dites sur des sujets donnés, et les pages de *liens* qui contiennent essentiellement des liens vers d'autres pages (qui, elles, contiennent l'information). En réalité, la situation est plus complexe et il y a bien entendu des pages qui sont à la fois dans les deux catégories.

Pour tenir compte de ces spécificités, on cherche à associer à chaque page P_i, un **score de contenu**, noté c_i, et un **score de liens**, noté l_i. L'idée est que plus c_i est grand, plus les informations contenues dans la page P_i sont supposées pertinentes et plus l_i est grand, plus les liens contenus dans la page P_i sont pertinents.

Afin de calculer les valeurs de ces *scores*, la philosophie du modèle proposé est la suivante :

- Les pages ayant un score de liens élevé doivent pointer vers des pages ayant un score de contenu élevé.
- Réciproquement, les pages ayant un score de contenu élevé doivent être référencées dans les pages ayant un score de liens élevé.

Cette philosophie va être mise en œuvre à travers une méthode itérative.

2.2. *Description de l'algorithme*

On note $c^n = (c_i^n)_i$ et $l^n = (l_i^n)_i$ les vecteurs colonnes contenant les deux types de scores de chaque page à l'itération n. Partant de la philosophie décrite ci-dessus, l'algorithme est le suivant :

– On choisit une valeur initiale des scores de lien $l^0 \in (\mathbb{R}_+)^N$

– On calcule par récurrence

$$\tilde{c}_j^{n+1} = \sum_{i|a_{ij}=1} l_i^n, \qquad \forall j \in \{1,\dots,N\}, \tag{2}$$

$$\tilde{l}_i^{n+1} = \sum_{j|a_{ij}=1} \tilde{c}_j^{n+1}, \qquad \forall i \in \{1,\dots,N\}. \tag{3}$$

$$c^{n+1} = \frac{\tilde{c}^{n+1}}{\|\tilde{c}^{n+1}\|_2}, \qquad l^{n+1} = \frac{\tilde{l}^{n+1}}{\|\tilde{l}^{n+1}\|_2}. \tag{4}$$

Remarquons que pour calculer $\tilde{l}^{n+1}$, on utilise la nouvelle valeur $\tilde{c}^{n+1}$ du vecteur des scores de contenu. Par ailleurs, on se convainc aisément que l'étape de normalisation est essentielle pour espérer la convergence de l'algorithme. De plus, celle-ci ne nuit pas à l'interprétation des résultats obtenus pour classer les pages selon leur pertinence.

Remarque 1. *Si lors de la première itération, on a $\tilde{l}^1 = 0$, alors on ne peut pas effectuer la renormalisation* (4)*. On pose alors $c^1 = 0$, $l^1 = 0$ et on dit que l'algorithme a convergé en une itération. Si en revanche $\tilde{l}^1 \neq 0$, alors $\tilde{l}^n$ et $\tilde{c}^n$ ne s'annuleront jamais et l'étape* (4) *est donc bien définie pour tout n.*

2.3. *Analyse*

La première question naturelle que l'on se pose est celle de la convergence de la méthode.

Théorème 1. *L'algorithme* (2)–(4) *converge pour tout choix de la donnée initiale l^0 vers deux vecteurs à coefficients positifs, notés c et l. De plus, $c_i = 0$ si P_i est une racine du graphe et $l_i = 0$ si P_i est une feuille du graphe. Enfin, les scores obtenus à la limite sont compatibles au sens où l'on a*

$$l \text{ est colinéaire à } Ac \text{ et } c \text{ est colinéaire à } {}^tAl.$$

Proposition 1. *Si tous les coefficients de l^0 sont strictement positifs, alors c et l sont des vecteurs propres respectifs des matrices ${}^tAA$ et $A{}^tA$, associés à la plus grande de leurs valeurs propres notée Λ.*

Malheureusement plusieurs situations pathologiques peuvent se produire. Tout d'abord, les sous-espaces propres des deux matrices pour la valeur propre Λ peuvent être de dimension d_Λ strictement plus grande que 1, auquel cas le résultat de l'algorithme à convergence dépendra du choix de la donnée initiale $l^0 > 0$. D'autre part, il peut se produire que l'on obtienne $c_i = 0$ même pour des pages P_i qui ne sont pas des racines du graphe, ou bien $l_i = 0$ même pour des pages P_i qui ne sont pas des feuilles du graphe. Dans ce cas, tout se passe comme si l'algorithme ignorait les pages en question, ce qui n'est pas souhaitable.

Définition 1. *On dit que l'algorithme* (2)–(4) *est* **bien posé** *si on a* $d_\Lambda = 1$ *et si les scores* c *et* l *obtenus à la limite vérifient :*

- *Si* P_i *n'est pas une racine, alors* $c_i > 0$.
- *Si* P_i *n'est pas une feuille, alors* $l_i > 0$.

Pour étudier le caractère bien posé de l'algorithme nous aurons besoin de deux définitions.

Définition 2. *On dit que le graphe orienté* G *est* **fortement connexe** *si pour toutes pages distinctes* P_i *et* P_j*, il existe un chemin* **orienté** *dans* G *de* P_i *vers* P_j.

On dit que le graphe orienté G *est* **faiblement connexe** *si pour toutes pages distinctes* P_i *et* P_j*, il existe un chemin* **non orienté** *(c'est-à-dire où l'on ignore le sens de parcours des arêtes) qui relie* P_i *à* P_j.

Définition 3 (Graphe de contenu)**.** *On associe au graphe orienté* G*, un graphe non orienté pondéré* G' *défini de la façon suivante :*

- *Les sommets de* G' *sont les pages* P_i *qui ne sont pas des racines du graphe* G.
- G' *contient une arête (non orientée) entre* P_i *et* P_j *si et seulement s'il existe une page* P_k *(éventuellement une racine de* G*) qui pointe à la fois vers* P_i *et* P_j*. On affecte à cette arête un poids égal au nombre de telles pages* P_k.
- *Conventionnellement, on rajoute au graphe* G' *une arête « en boucle » qui joint* P_i *à elle-même et dont le poids est le nombre d'autres pages* P_k *qui pointent vers* P_i *dans* G.

Intuitivement, le rôle du graphe G' dans l'étude de l'algorithme (2)–(4) se comprend en regardant deux itérations successives de celui-ci. Soit $(P_i \leftrightarrow P_j)$ une arête non orientée dans G' et P_k une page qui pointe vers P_i et P_j. À l'itération n, le score de contenu de la page P_i influe sur le score de liens de P_k, puis à l'itération suivante, le score de liens de P_k influe sur le score de contenu de P_j.

Par ailleurs, la matrice d'adjacence M du graphe G' n'est autre que la matrice ${}^tAA$ dans laquelle on a supprimé les colonnes et les lignes nulles. Remarquons enfin que le graphe G' n'étant pas orienté, les notions de forte et faible connexité coïncident.

Théorème 2. *Si* G' *est connexe, alors le modèle* (2)–(4) *est bien posé.*

Ce théorème s'obtient en montrant que l'hypothèse de connexité de G' implique que tout vecteur propre v de M à coefficients positifs ou nuls a en fait tous ses coefficients strictement positifs.

On peut par ailleurs établir que la condition de connexité de G' est en réalité une condition nécessaire.

2.4. *Améliorations*

La condition de connexité sur le graphe G' obtenue du théorème 2 est beaucoup trop forte et on voit bien que celle-ci n'est pas vérifiée même dans des cas simples. On s'intéresse donc ici à une adaptation du modèle qui fournisse un algorithme bien posé sous des conditions plus faibles et donc plus réalistes.

Pour cela, on constate que le précédent modèle ne tient compte que des liens directs entre les pages. L'idée du nouvel algorithme est de prendre en compte les liens d'ordre plus élevé entre les pages. Concrètement cela signifie que si une page P_i est accessible en deux clics depuis une page P_j dont le score de liens est élevé, alors son score de contenu doit être valorisé et réciproquement.

L'idée est donc de remplacer la matrice d'adjacence A dans l'algorithme par une nouvelle matrice $\tilde{A}$ construite à partir de A. On propose de prendre par exemple

$$\tilde{A} = e^A - I = A + \frac{A^2}{2} + \cdots + \frac{A^n}{n!} + \cdots . \tag{5}$$

Ceci revient à prendre en compte tous les chemins dans le graphe G en affectant un poids $\frac{1}{n!}$ aux chemins de longueur n. L'algorithme devient

$$\tilde{c}_j^{n+1} = \sum_i \tilde{a}_{ij} l_i^n, \qquad \forall j \in \{1,\dots,N\}, \tag{6}$$

$$\tilde{l}_i^{n+1} = \sum_j \tilde{a}_{ij} \tilde{c}_j^{n+1}, \qquad \forall i \in \{1,\dots,N\}. \tag{7}$$

$$c^{n+1} = \frac{\tilde{c}^{n+1}}{\|\tilde{c}^{n+1}\|_2}, \qquad l^{n+1} = \frac{\tilde{l}^{n+1}}{\|\tilde{l}^{n+1}\|_2}. \tag{8}$$

Théorème 3. *Si le graphe G est faiblement connexe, alors le modèle* (6)–(8)*, où $\tilde{A}$ est définie en* (5)*, est bien posé.*

Là encore la condition sur le graphe G est en réalité une condition nécessaire.

Pour démontrer ce théorème, on définit un nouveau graphe $\tilde{G} = G_{\tilde{A}}$ comme celui associé à la nouvelle matrice $\tilde{A}$ (voir section 1.2) et le nouveau graphe de contenu $\tilde{G}'$ comme celui associé à la matrice ${}^t\tilde{A}\tilde{A}$ dont on a supprimé les lignes et colonnes nulles. Notons que $\tilde{G}$ et G ont les mêmes feuilles et les mêmes racines.

D'après ce qu'on a vu précédemment, il suffit de démontrer que le graphe non orienté $\tilde{G}'$ est connexe. Pour cela, on suppose que ce n'est pas le cas et qu'on peut séparer $\tilde{G}'$ en deux sous ensembles non vides disjoints $\tilde{G}' = H \sqcup L$ tels qu'il n'existe aucune arête dans $\tilde{G}'$ entre un sommet dans H et un sommet dans L.

Soient $h \in H$ et $l \in L$ quelconques. Comme G est faiblement connexe, il existe un chemin **non orienté** dans G qui relie h à l :

$$h = g_1 \leftrightarrow g_2 \leftrightarrow \cdots \leftrightarrow g_{k-1} \leftrightarrow g_k = l. \tag{9}$$

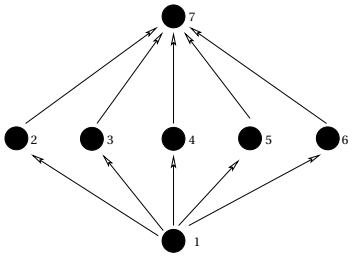
Soit i le plus petit indice tel que $g_i \notin H$. On a $i \geq 2$ et $g_i \notin L$. En effet, dans le cas contraire on aurait $g_{i-1} \in H$ et $g_i \in L$ avec une arête orientée dans G de g_{i-1} vers g_i ou bien de g_i vers g_{i-1} (ou éventuellement les deux). Dans le premier cas par exemple, comme g_{i-1} n'est pas une racine (car il est dans $\tilde{G}'$), il existe $g \in G$ tel que $g \to g_{i-1}$ et on a donc $g \to g_{i-1} \to g_i$. Les sommets g_i et g_{i-1} sont donc accessibles depuis un sommet commun dans G, ce qui signifie qu'il y a une arête entre g_i et g_{i-1} dans $\tilde{G}'$, ce qui contredit l'hypothèse de départ.

On déduit donc que $g_{i-1} \in H$ et que g_i n'est ni dans H, ni dans L, c'est donc nécessairement une racine dans G. On peut alors supposer que $g_{i+1} \in L$ quitte à refaire le raisonnement en changeant h en g_{i+1}. Comme g_i est une racine, on a nécessairement les arêtes orientées

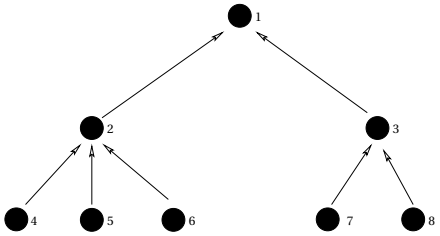
suivantes : $g_i \to g_{i-1}$ et $g_i \to g_{i+1}$ ce qui montre que $g_{i-1} \in H$ et $g_{i+1} \in L$ sont accessibles depuis un sommet commun g_i et qu'il y a donc une arête entre ces deux sommets dans le graphe $\tilde{G}'$, ce qui contredit aussi l'hypothèse.

3. Exemples

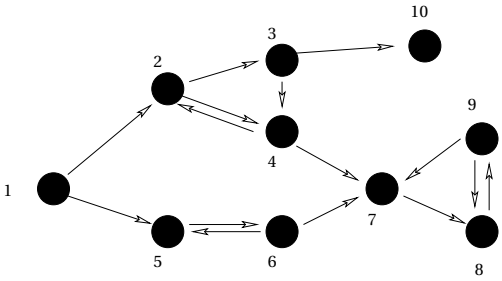
Voici quelques exemples très simples sur lesquels on peut illustrer les divers concepts et résultats qui précédent :



Graphe 1



Graphe 2



Graphe 3

Suggestions pour le développement

► *Soulignons qu'il s'agit d'un menu à la carte et que vous pouvez choisir d'étudier certains points, pas tous, pas nécessairement dans l'ordre, et de façon plus ou moins fouillée. Vous pouvez aussi vous poser d'autres questions que celles indiquées plus bas. Il est très vivement souhaité que vos investigations comportent une partie traitée sur ordinateur et, si possible, des représentations graphiques de vos résultats.*

- Proposer un algorithme similaire à celui proposé dans la section 2.2 mais qui soit initialisé par une valeur des scores de contenu $c_0 \in \mathbb{R}^n$. Comparez les deux algorithmes.
- Pourquoi utilise-t-on les valeurs de $\tilde{c}^{n+1}$ pour calculer l^{n+1} et non pas celles de $\tilde{c}^n$?
- Démontrer que la matrice d'adjacence de G' est bien ${}^tAA$ privée de ses lignes et colonnes nulles. Comment interpréter la matrice $A{}^tA$ privée de ces lignes et colonnes nulles ?
- Illustrer sur les exemples donnés dans le texte, ou sur des exemples que vous introduirez, les propriétés et les défauts de l'algorithme proposé et les améliorations éventuellement obtenues grâce aux variantes données dans le texte.
- Démontrer le théorème 1 et/ou le théorème 2.
- Que peut-on dire de la vitesse de convergence de l'algorithme ?
- Construire les graphes de contenu des exemples proposés.
- Donner une interprétation du graphe $\tilde{G}$ construit lors de la démonstration du théorème 3. Le construire dans les exemples proposés.
- Vérifier numériquement, ou démontrer, le fait que si on définit $\tilde{A} = A + \alpha A^2$, pour $\alpha > 0$ quelconque au lieu de (5), alors le résultat du théorème 3 est encore vrai. Que pensez-vous de cette nouvelle méthode d'un point de vue pratique ?