Rapport de TP sur les PageRank

Modéliser l'aléa

Jean-Christophe CORVISIER
Mohammed Amine KHELDOUNI

4 janvier 2019

Table des matières

1 Introduction

On se propose dans ce TP d'étudier les PageRank à l'aide de la théorie markovienne. Nous allons nous intéressé plus précisément, étant donné un graphe de pages web, à la maximisation de son PageRank. Celui-ci peut en effet être écrit en fonction d'un certain vecteur propre (mesure invariante) d'une matrice stochastique.
Enfin, nous verrons cette maximisation dans le cadre du théorème érgodique et la modélisation de ce problème en fonction d'un contrôle ν donné.

2 Réponses aux questions théoriques

2.1 Question 1

X_n est une chaîne de Markov de matrice de transition P.
Le comportement de ce processus est donc régi par la loi de probabilité conditionnelle :

$$\mathbb{P}(X_{n+1} = j | X_n = i) = P(i, j)$$

Le vecteur z est appelé vecteur de téléportation car pour une matrice creuse P_{ss}, on remplace les lignes nulles par le vecteur z pour former la matrice P_1. Cela revient à téléporter la chaîne de Markov à un nouvel état, retrouvé par les probabilités coefficients de z.

Pour la suite, nous avons considéré dans le code SCILAB un nombre de pages (noeuds du graphe) de $n = 10$ et un $\alpha = 0.8$.
Le graphe affiché est comme suit

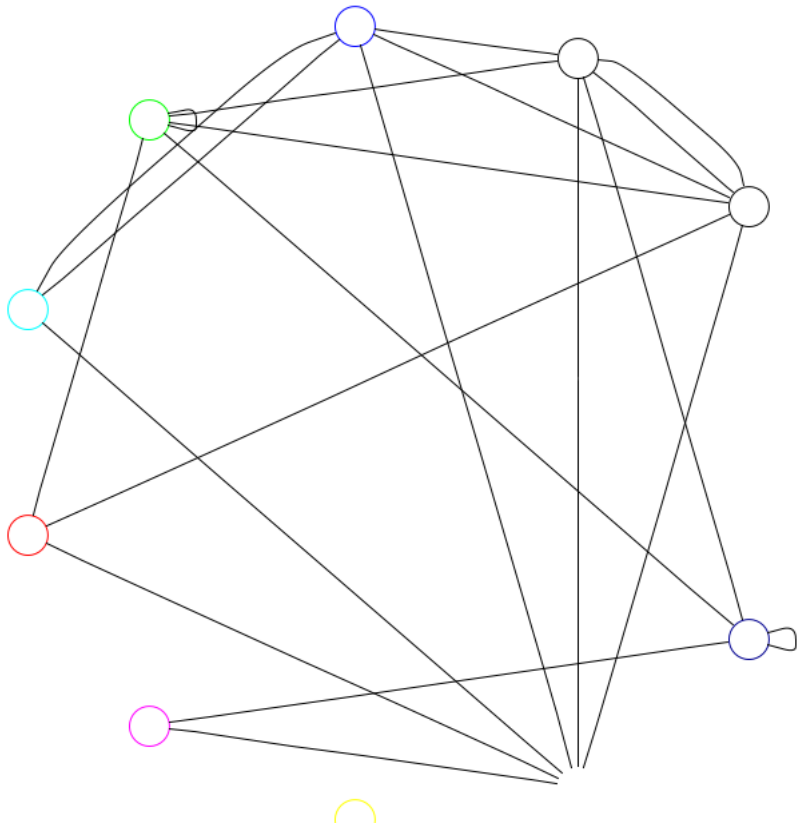

FIGURE 1 – Graphe décrivant les transitions possible d'une page à l'autre

2.2 Question 2

Le graphe décrivant les états possibles de la chaîne de Markov X_n est connexe (avec une seule composante). Ce qui permet donc d'affirmer que X_n est récurrente positive.
Par suite, la mesure invariante π existe et est unique.

Une fois la fonction *google* implémentée, nous avons préparé notre matrice de transition P pour les calculs et les simulations qui suivent.
On vérifie bien que $\forall i, \quad \sum_j P_{i,j} = 1$

2.3 Question 3

Pour privilégier les calculs sur les matrices creuses, on effectue le calcul suivant :

$$P_1 x = \alpha * P_{ss}^T x + (\alpha d \bullet z)^T x + ((1-\alpha)e \bullet z)^T x$$

Et on vérifie bien que pour x un vecteur aléatoire de taille n, on a l'égalité ci-dessus. Numériquement, cette égalité est vérifiée et la différence entre les deux termes est bien négligeable (de l'ordre de 10^{-15}).

2.4 Question 4

En diagonalisant la matrice P^T, on cherche le vecteur propre associé à la valeur propre 1. Ce vecteur propre n'est autre que la probabilité invariante transposée π^T car

$$\pi P = \pi \Leftrightarrow P^T \pi^T = \pi^T$$

Après calcul, nous obtenons le vecteur π suivant :

$$\pi = \begin{pmatrix} 0.102 \\ 0.107 \\ 0.059 \\ 0.06 \\ 0.061 \\ 0.068 \\ 0.385 \\ 0.048 \\ 0.049 \\ 0.06 \end{pmatrix}$$

de PageRank $\sum_i \pi_i = 1$.

2.5 Question 5

On propose pour cette question une seconde méthode de calcul de la mesure invariante π. On utilise pour cette méthode, un calcul itératif d'une suite p_k telle que $P_1 p_k = p_{k+1}$. Cette suite converge vers π.

On obtient donc la bonne valeur du vecteur π à 10^{-17} d'erreur.

2.6 Question 6

On redéroule maintenant le même algorithme itératif pour le calcul de π, en utilisant la matrice creuse P_{ss}.
Et on s'assure également que la différence entre $P_1\pi$ et π est suffisamment faible. Cette valeur est en effet de l'ordre de 10^{-17} également.

2.7 Question 7

Pour cette question, on veut maximiser le PageRank des m premières pages. Ceci revient à maximiser $\sum_{i=0}^{m} \pi_i$.

On pense alors à l'algorithme naïf qui consisterait à générer toutes les sous matrices de la matrice d'adjacence, composées des p premières lignes et des $n - m$ dernières colonnes. Et on compare les PageRank calculés à partir de ces sous matrices, dont on aura remplacer pour chacune un des coefficients de la matrice d'adjacence qui est nul par un 1.

Pour cela, on effectue la décomposition en nombre binaire d'une chiffre allant de 0 jusqu'au nombre de sous matrices possibles, donc jusqu'à $2^{p(n-m)} - 1$. Ensuite, on construit les différentes sous matrices à partir de cette décomposition binaire.

Résultats Pour $p = 2$ et $m = \frac{n}{2}$, le programme tourne bien et fournit la matrice de configuration suivante :

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

associée à un PageRank maximal de $\sum_{i=0}^{m} \pi_i = 0.538$

2.8 Question 8

Dans cette sous section, on s'assure de la convergence par le théorème érgodique, à savoir que :

$$\lim_{T\to+\infty} \frac{1}{T}\mathbb{E}[\sum_{t=0}^{T-1} r(X_t)] = \sum_x \pi(x)r(x)$$

Pour $T = 100000$, la différence entre les deux termes nous donne

$$|\frac{1}{T}\mathbb{E}[\sum_{t=0}^{T-1} r(X_t)] - \sum_x \pi(x)r(x)| \approx 9.4\ \ 10^{-6}$$

Il y a donc bien convergence du résultat de la fonction SCILAB *"ergodique_markov_T"* vers celui de *"ergodique_markov"* pour T suffisamment grand.

Par ailleurs, cette limite $c = \sum_x \pi(x)r(x)$, peut être retrouvée en résolvant le système linéaire suivant :

$$\begin{cases} (P-I)w + R - c &= 0 \\ (P-I)c &= 0 \end{cases}$$

On vérifie que le noyau de $P - I$ est engendré par le vecteur composé par des 1. Cela se fait en exécutant la fonction *linsolve* qui résout le système linéaire $(P - I)x = 0$. Cela fournit un noyau $Ker(P - I) = vect(e)$ où $e = (1, 1, ..., 1) \in \mathcal{M}_{n,1}(\mathbb{R})$.

2.9 Question 9

Notations

On note pour cette sous section P_r le projecteur spectral sur l'espace propre associé à la valeur propre 1 (et donc au vecteur propre π). Puis on construit les matrice $A = P - I$, $S = P_r - (P_r - A)^{-1}$ et R la matrice formée des coefficients $(r(x))_x$.

Résultats

En poursuivant le code SCILAB fournit dans l'énoncé, on vérifie bien que les coefficients de $S\ P_r$ et $P_r\ S$ sont tous nuls à 10^{-17} et 10^{-16} près. Et enfin, on vérifie que $w = -S\ R$ et $c = P_r\ R$ sont solutions du système linéaire ci-dessus :

$$Aw + R - c = 10^{-15} \begin{pmatrix} -0.88 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Puis

$$Ac = 10^{-15} \begin{pmatrix} 0.336 \\ -0.177 \\ -0.128 \\ -0.594 \end{pmatrix}$$

w et c sont donc bien solutions du système linéaire.

Réciproquement, on vérifie que c doit être égal à $P_r R$. On obtient déjà une matrice $P_r\ A$ de coefficients presque nuls (de l'ordre de 10^{-16}). Puis comme $P_r R - P_r c = 0$ et $Ac = 0$, on a par sommation des deux équations :

$$P_r R = (P_r - A)c$$

Il s'en suit que

$$c = (P_r - A)^{-1} PrR = (P_r - S)P_r R = P_r R$$

car P_r est un projecteur et $SP_r = 0$.
En conclusion, c doit valoir $P_r R$, ce qu'on peut vérifier avec la résolution de système linéaire de SCILAB.

2.10 Question 10

Si w est un point fixe de la fonctionnelle $w \to \alpha P_1 w + b$, alors, le calcul de $Pw + R$ se fait, en notant $c = (1 - \alpha)zw$, comme suit

$$\begin{array}{rcll} Pw + R & = & \alpha P_1 w + (1 - \alpha)zw + R & \\ & = & w - b + R + (1 - \alpha)zw & \text{car } w \text{ est un point fixe} \\ & = & w + c & R \text{ à les même coefficients que } b \end{array}$$

On vérifie ce résultat numériquement par les étapes suivantes :

— w est un point fixe :

$$w - P_1 w + b = 10^{-15} \begin{pmatrix} 0.61 & -0.55 & -0.66 & 0.66 \end{pmatrix}$$

— L'égalité est bien vérifiée :

$$w + c - Pw + R = 10^{-15} \begin{pmatrix} 0.888 & -0.888 & 0 & 0.888 \end{pmatrix}$$

2.11 Question 11

L'opérateur $w \to \alpha P_1 w + b$ est un opérateur contractant et admet donc un point fixe w. On en déduit alors une méthode itérative pour le calcul de ce dernier, programmée dans la fonction *"iterative_ c"*.

On trouve le vecteur w suivant :

$$w = \begin{pmatrix} 2.52 \\ 2.6 \\ 2.23 \\ 2.42 \end{pmatrix}$$

On vérifie bien évidemment que w est bien un point fixe trouvé pour l'opérateur introduit en début de sous section avec une erreur de $\epsilon \approx 10^{-8}$.

2.12 Question 12

Nous avons tenté d'implémenter l'algorithme donné, en réalisant des maximisations successives des w_x en testant toutes les combinaisons possibles de modification de la matrice d'adjacence. A chaque étape k de l'algorithme, nous effectuons une boucle sur x variant de 1 à n, puis dans cette boucle nous testons toutes les combinaisons de modifications de la matrice d'adjacence qui permettent de maximiser à w^k fixé la valeur en position x du vecteur $alphaP^{\nu}prim * w^k + P^{\nu}.R_m$. Puis une fois la meilleure redirection ν^k trouvée, on utilise la fonction de recherche de point fixe de la question 11 pour trouver w^{k+1}. On s'arrête lorsqu'on a dépassé un nombre maximal d'itérations ou bien que deux valeurs successives w^k et w^{k+1} sont très proches en normes. Notre algorithme ne fonctionne pas très bien car retourne un pageRanking inférieur à celui trouvé en question 7 , 0.12 avec une matrice d'adjacence modifiée comme décrite ci-desous :

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

3 Conclusion

Le *PageRank*, qui permet de quantifier la popularité d'un site web, est un facteur clé pour un moteur de recherche comme Google. Néanmoins, la difficulté du calcul du *PageRank*, due a la taille gigantesque du réseau, rend sa maximisation un point sensible et situe ce problème au coeur du domaine d'optimisation stochastique.

Nous avons dans ce projet, esquissé quelques méthodes et algorithmes s'appuyant sur des théorèmes mathématiques probabilistes et la théorie de la mesure tels que le théorème érgodique, pour optimiser le coefficient du *PageRank* et donné donc plus d'intérêt à une page internet selon les liens qu'elle possède dans le web.