UVSQ université PARIS-SACLAY | UFR des Sciences CAMPUS DE VERSAILLES

2024 – 2025

Ranking et recommandation – Liste des Projets

M1 Informatique

Version du 16 avril 2025

Franck Quessette – franck.quessette@uvsq.fr
Sandrine Vial – sandrine.vial@uvsq.fr

Chaque projet met en avant un mécanisme nouveau lié à Pagerank. Il faut donc généralement comparer les résultat du Pagerank de base avec la modification proposée dans le sujet du projet.

Un ensemble de matrices est mis à votre disposition sur e-campus, utilisez celles que vous trouverez pertinentes pour montrer l'effet de la modification proposée dans le sujet. Vous pouvez aussi utiliser d'autres matrices du web comme `https://sparse.tamu.edu`.

À vous de choisir quelles expériences faire avec quelles valeurs de paramètres et comment vous présentez les résultats, généralement une courbe permet de bien faire passer l'intuition.

Quelques notations communes :

— x la distribution stationnaire issue de Pagerank.

— $x^{(k)}$ la distribution à la fin de l'itération k.

— $H^{[t]}$ le graphe du web à la date t, généralement le mois.

— $x^{[t]}$ la distribution stationnaire issue de Pagerank à la date t (attention ce n'est pas l'itération t).

— G La matrice d'itération de Pagerank.

Ce qui est attendu

1. Vous devez programmer les algos demandés en reprenant/modifiant l'algo de Pagerank que vous avez fait en TD.
2. Dans chaque sujet, il y a des paramètres à faire varier. Vous devez trouver dans quels intervalles il est pertinent de faire varier ces paramètres.
3. Vous devez faire un rapport en LaTeX de quelques pages expliquant ce que vous avez fait et donnant les résultats.

Modalités d'évaluation

1. Un numéro de projet est affecté à chaque étudiant (voir section ci après). Le projet est à faire en binôme, si vous et votre binôme avez des numéros de projet différents choisissez un des deux. Si vous et votre binôme avez le même numéro n, choisissez entre n et $9 - n$.
2. Vous devrez remettre le code et le rapport sur e-campus pour le **vendredi 16 mai 2025, 23h59**.
3. Il y aura une présentation orale, le **lundi 19 mai 2025, 23h59** durant laquelle on vous interrogera sur les détails de votre code. Toute suspicion d'utilisation d'une IA pour faire le code sera prise en compte dans la note finale.

Table des matières

1 Utilisation du Pagerank précédent pour initialiser un nouveau calcul-graphe Erdos 3

2 Utilisation du Pagerank précédent pour initialiser un nouveau calcul-graphe attachement préférentiel 3

3 Calcul du Pagerank par Gauss-Seidel ascendant 3

4 Calcul du Pagerank par Gauss-Seidel descendant 4

5 Calcul du Pagerank par l'algorithme utilisant les vecteurs ∇ et Δ 4

6 Matrice NCD 5

7 Simulation d'un Google Bombing 5

8 Le backspace pour les pages sans lien de sortie 5

Affectation des projets

Nom	Prénom	Projet
[redacted]	[redacted]	5
[redacted]	[redacted]	6
[redacted]	[redacted]	7
[redacted]	[redacted]	6
[redacted]	[redacted]	6
[redacted]	[redacted]	4
[redacted]	[redacted]	1
[redacted]	[redacted]	5
[redacted]	[redacted]	7
[redacted]	[redacted]	2
[redacted]	[redacted]	1
[redacted]	[redacted]	7
[redacted]	[redacted]	3
[redacted]	[redacted]	5
[redacted]	[redacted]	3
[redacted]	[redacted]	3
[redacted]	[redacted]	8
[redacted]	[redacted]	8
[redacted]	[redacted]	1
[redacted]	[redacted]	4
[redacted]	[redacted]	2
[redacted]	[redacted]	4
[redacted]	[redacted]	8
[redacted]	[redacted]	2

1 Utilisation du Pagerank précédent pour initialiser un nouveau calcul-graphe Erdos

Comme le Pagerank est calculé chaque mois et que le graphe du web évolue lentement, on peut essayer d'initialiser le vecteur x à la valeur obtenue lors du mois précédent. Il faut simplement faire attention aux nouveaux sommets.

L'initialisation est la suivante :

— si i est dans le graphe $H^{[t+1]}$ et $H^{[t]}$, alors $x^{[t+1]}[i] = x^{[t]}[i]$;

— si i est dans le graphe $H^{[t+1]}$ et pas dans $H^{[t]}$, alors $x^{[t+1]}[i] = 0$.

Etudier la convergence de l'algorithme des puissances lorsqu'il est initialisé comme proposé par rapport à la version du cours où il est initialisé avec e/n.

Pour cela, vous proposez des algorithmes de modification de graphes : ajout de sommets, et ajout d'arcs depuis les nouvelles pages vers les anciennes en utilisant un ajout d'un graphe de Erdos.

Un graphe de Erdos est un graphe dont le arcs sont générés aléatoirement avec une probabilité p. Voir https://perso.ens-lyon.fr/eric.thierry/Graphes2010/vincent-picard.pdf.

Paramètres à étudier :

— le nombre de nouveau sommets ;

— p ;

— α.

2 Utilisation du Pagerank précédent pour initialiser un nouveau calcul-graphe attachement préférentiel

Comme le Pagerank est calculé chaque mois et que le graphe du web évolue lentement, on peut essayer d'initialiser le vecteur x à la valeur obtenue lors du mois précédent. Il faut simplement faire attention aux nouveaux sommets.

L'initialisation est la suivante :

— si i est dans le graphe $H^{[t+1]}$ et $H^{[t]}$, alors $x^{[t+1]}[i] = x^{[t]}[i]$;

— si i est dans le graphe $H^{[t+1]}$ et pas dans $H^{[t]}$, alors $x^{[t+1]}[i] = 0$.

Etudier la convergence de l'algorithme des puissances lorsqu'il est initialisé comme proposé par rapport à la version du cours où il est initialisé avec e/n.

Pour cela, vous proposez des algorithmes de modification de graphes : ajout de sommets, et ajout d'arcs depuis les nouvelles pages vers les anciennes en utilisant un ajout d'un graphe par attachement préférentiel.

L'attachement préférentiel consiste à ajouter des arcs depuis les nouveaux sommets vers les sommets de degré élevé dans le graphe initial. Voir https://fr.wikipedia.org/wiki/Modèle_de_Barabási-Albert

Paramètres à étudier :

— le nombre de nouveau sommets ;

— le nombre de nouveau arcs ;

— α.

3 Calcul du Pagerank par Gauss-Seidel ascendant

L'algorithme de Gauss-Seidel consiste à mélanger les solutions à l'itération $k+1$ déjà calculées et les solutions à l'itération k pour les autres indices quand on calcule la solution à l'itération $k+1$.

Au lieu de :

$$x^{(k+1)}[i] = \sum_{j=1}^{n} x^{(k)}[j]G[j,i]$$

on a :

$$x^{(k+1)}[i](1 - G[i,i]) = \sum_{j=1}^{i-1} x^{(k+1)}[j]G[j,i] + \sum_{j=i+1}^{n} x^{(k)}[j]G[j,i]$$

Et dans l'algorithme les $x^{(k+1)}[i]$ sont calculés en i ascendants (du plus petit indice au plus grand). Il faut ensuite renormaliser le vecteur x car il n'est plus de norme égale à 1.

Proposez une version de Gauss-Seidel ascendant efficace en mémoire et en calcul et comparez expérimentalement avec la méthode des puissances.

Paramètre à étudier :

— α.

4 Calcul du Pagerank par Gauss-Seidel descendant

L'algorithme de Gauss-Seidel consiste à mélanger les solutions à l'itération $k+1$ déjà calculées et les solutions à l'itération k pour les autres indices quand on calcule la solution à l'itération $k+1$.

Au lieu de :

$$x^{(k+1)}[i] = \sum_{j=1}^{n} x^{(k)}[j]G[j,i]$$

on a :

$$x^{(k+1)}[i](1-G[i,i]) = \sum_{j=1}^{i-1} x^{(k)}[j]G[j,i] + \sum_{j=i+1}^{n} x^{(k+1)}[j]G[j,i]$$

Et dans l'algorithme les $x^{(k+1)}[i]$ sont calculés en i descendants (du plus grand indice au plus petit). Il faut ensuite renormaliser le vecteur x car il n'est plus de norme égale à 1.

Proposez une version de Gauss-Seidel descendant efficace en mémoire et en calcul et comparez expérimentalement avec la méthode des puissances.

Paramètre à étudier :

— α.

5 Calcul du Pagerank par l'algorithme utilisant les vecteurs ∇ et Δ

On pose $\nabla[j] = \min_i G[i,j]$ et $\Delta[j] = \max_i G[i,j]$. Ce sont deux vecteurs lignes.

On construit par itération deux vecteurs lignes $X^{(k)}$ et $Y^{(k)}$.

L'algorithme suivant converge vers la distribution stationnaire π de la chaine de Markov de matrice G et on a :

$$\forall k,\ \forall i,\quad X^{(k)}[i] \leq \pi[i] \leq Y^{(k)}[i]$$

Algorithme :

1. $X^{(0)} \leftarrow \nabla$.
2. $Y^{(0)} \leftarrow \Delta$.
3. Faire une boucle sur k :
   (a) $X^{(k+1)} = \max(X^{(k)}, X^{(k)}G + \nabla(1 - ||X^{(k)}||_1))$
   (b) $Y^{(k+1)} = \min(Y^{(k)}, Y^{(k)}G + \Delta(1 - ||Y^{(k)}||_1))$
   (c) jusqu'à ce que $||X^{(k)} - Y^{(k)}||_1 < \varepsilon$.

Proposer une version efficace en mémoire et en calcul de l'algorithme et comparez expérimentalement avec la méthode des puissances.

Paramètre à étudier :

— α.

6 Matrice NCD

Le projet a pour but de tester l'algorithme de Pagerank lorsque la matrice du WEB est NCD (voir https://en.wikipedia.org/wiki/Nearly_completely_decomposable_Markov_chain) et que α s'approche de 1. Comme on est pas sûr que les graphes du WEB fournis sont NCD, on va les modifier de la manière suivante :

1. Essayer de rendre les matrices du WEB décomposables en enlevant des arcs. On propose de faire deux expériences : enlever 10% puis 20% des arcs des graphes du web. Vous êtes libre de le faire de façon aléatoire (avec un tirage aléatoire) ou déterministe (on enlève un arc sur 10 ou sur 5). Si vous choisissez un tirage aléatoire, la méthode est la suivante :
   - Vous parcourez le fichier du graphe.
   - Pour chaque arc, vous faites un tirage aléatoire u.
   - Si $u < 0.1$ (ou 0.2), vous enlevez l'arc sinon vous le gardez.
   - N'oubliez pas de mettre à jour le degré sortant du sommet.

Comparez le temps de convergence par rapport à la matrice de base.

Paramètre à étudier :

- α.

7 Simulation d'un Google Bombing

A partir d'un graphe du web, ajouter différentes structures de graphes composés d'attaquants et d'une cible (déjà dans le graphe) et faites varier les nombres d'attaquants et la structure du graphe du web entre les attaquants avant de calculer dans chaque cas le Pagerank de la cible.

Plus précisément, vous allez étudier l'impact sur 3 cibles potentielles que vous déterminerez grâce aux pertinences initiales calculée par Pagerank : une cible de pertinence forte, une de pertinence moyenne, et une de pertinence faible.

Les structures de graphes que vous ajouterez sont :

- graphe complet à n sommets, tous les sommets pointant vers la cible.
- anneau à $n+1$ sommets, la cible faisant partie de l'anneau.
- n sommets isolés pointant vers la cible.

Essayer de déduire des règles empiriques pour avoir une attaque efficace. On supposera que l'attaque est efficace si la probabilité calculée par Pagerank devient significativement plus forte.

Paramètres à étudier :

- La taille du graphe attaquant.
- Le type du graphe attaquant.
- Le type de cible.

8 Le backspace pour les pages sans lien de sortie

On suppose que l'on n'utilise pas le modèle du surfer aléatoire pour donner des successeurs aux pages sans liens de sortie. On va supposer que l'utilisateur qui arrive sur une page sans url de sortie utilise la touche backspace pour revenir en arrière dans sa navigation.

Mais comme il y a peut-être plusieurs pages qui pointent sur une page sans lien de sortie, il faut se souvenir par où on est arrivé sur cette page. Pour cela, on modifie comme suit le graphe du web.

Chaque page P qui est de degré sortant nul et de degré entrant $D_{in} > 0$ est recopié en D_{in} exemplaires. Son nom devient (P, X) (où X a D_{in} valeurs) et la Xème page pointant sur P pointe maintenant sur la page (P, X) et est la seule à pointer sur cette page. Il est donc facile de savoir comment revenir de (P, X) quand on a visité cette page.

Le but du projet est de comparer les ranking obtenus par l'approche de Google et par cette autre approche. Attention, on garde la seconde modification de la matrice (c'est à dire le mélange avec le coefficient de 0.85 entre le graphe du web modifié et la matrice du surfer aléatoire).

Si les graphes du web que vous utiliserez n'ont pas de pages de degré sortant nul (on ne sait pas), vous transformerez aléatoirement ces graphes en détruisant des liens de sortie.

Paramètre à étudier :

— α.