Rapport

Frederick Coupvent Des Graviers - Sonny Klotz - Florian Lienhart - Thomas Momenzadeh

Projet M1 Informatique
Accélération de Aitken quadratique

21/05/2018

Module *Méthodes de ranking et recommandations*

Table des matières

1 Structures de données **1**
1.1 Le graphe du web 1
1.2 Implémentation 1

2 Méthode des puissances **2**
2.1 Algorithme 2
2.2 Principe 3
2.3 Complexité 3

3 Accélération de Aitken quadratique **3**

4 Résultats expérimentaux **4**
4.1 Description des données 4
4.2 Mesures de performances 4

Introduction

Le projet s'inscrit dans le cadre de l'UE **Méthodes de ranking et recommandations**, et plus particulièrement, nous étudierons l'algorithme *Pagerank* de Google.

Cet algorithme a le rôle de classifier les pages web en attribuant une note de **pertinence** à chacune des pages. Pour entrer un peu plus dans les détails, Google établit un graphe du web à l'aide des pages et des liens hypertextes contenues dans celles-ci.

La majeure problématique consiste à gérer efficacement la masse de données que représente le web. Ainsi, l'enjeu de *Pagerank* va être de traiter efficacement en mémoire le calcul des notes de pertinence.

Ce document va présenter dans une première partie le choix de l'implémentation pour les structures de données utilisées. Nous allons ensuite parler de la méthode des puissances pour le calcul des pertinences pour ensuite étudier une variante, l'accélération de Aitken quadratique. Enfin, nous présenterons les résultats expérimentaux sur six graphes du web.

1 Structures de données

1.1 Le graphe du web

Les graphes du web sont représentées dans notre programme sous la forme d'une matrice du web. Chaque ligne de la matrice représente une page web et les valeurs non nulles indique qu'il existe un lien hypertexte vers une autre page.

La taille du web étant immense, les graphes pourront aussi être représentés par des fichiers de données volumineux. À titre d'exemple, le fichier *wb-edu.txt* du sujet est un fichier d'une taille de 1 Go.

Il faut tout de même noter le caractère creux de notre matrice. En effet, la matrice de *wb-edu.txt* contient environ 10^7 lignes, ce qui nous donnerait, si on représente notre matrice entièrement, 10^{14} valeurs à stocker. Cette matrice serait constituée majoritairement de zéros puisque l'on sait le nombre de valeurs non nulles à environ $6 \cdot 10^7$, ce qui est beaucoup plus envisageable en terme de mémoire.

L'enjeu va donc être d'implémenter les algorithme du calcul de *Pagerank* en exploitant au mieux le caractère creux de la matrice.

1.2 Implémentation

Pour faire un choix d'implémentation convenable, il faut prendre de l'avance quant aux tâches réalisées par notre programme. L'essentiel de la complexité va reposer sur deux opérations, l'importation de la matrice à partir du fichier, et, le calcul des pertinences.

Le calcul des pertinences repose sur un produit vecteur-matrice. Soient n le nombre de noeuds du graphe du web, Π notre vecteur de pertinence et G notre matrice, le produit s'effectue de la manière suivante :

$$\forall i \in \{1, \cdots, n\}, (\Pi \cdot G)[i] = \sum_{j=1}^{n} \Pi[i] \cdot G[j,i]$$

On effectue n produits vecteur-colonne. Ainsi pour pouvoir ignorer les valeurs nulles de G (car elles ne modifient pas la valeur de la somme), il faut avoir un stockage de G sur les **colonnes**.

Maintenant, pour ce qui est des fichiers utilisés, ils utilisent une structure organisée selon les lignes. Ainsi, notre travail va être de lire linéairement notre fichier pour stocker les arcs, puis d'effectuer un tri rapide pour réorganiser nos arcs en fonction des colonnes. Un tel travail s'effectue en $O(n \cdot log(n))$ en moyenne.

2 Méthode des puissances

2.1 Algorithme

Algorithme 1 : Méthode des puissances

Données : Vecteur Π de pertinence de taille n, Matrice du web M, e vecteur de 1 de taille n

$k \leftarrow 0$ et $\Pi^{(k)} \leftarrow \frac{1}{n} \cdot e$;

répéter

  $k \leftarrow k+1$;

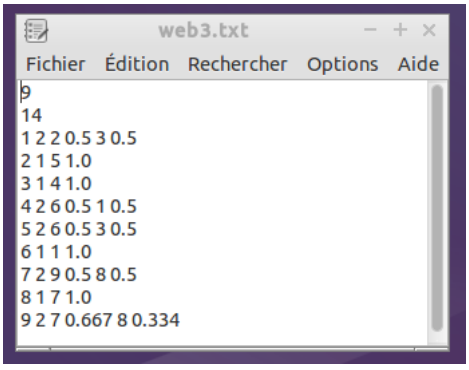
  pour $i \leftarrow 1$ *allant à* n **faire**

    $\Pi^{(k)}[i] \leftarrow \alpha \cdot (\sum_{j=1}^{n} \Pi^{(k-1)}[j] \cdot M[j,i]) + \frac{1-\alpha}{n} + \frac{\sigma \cdot \alpha}{n}$;

jusqu'à $\|\Pi^{(k+1)} - \Pi^{(k)}\|_1 > \varepsilon$;

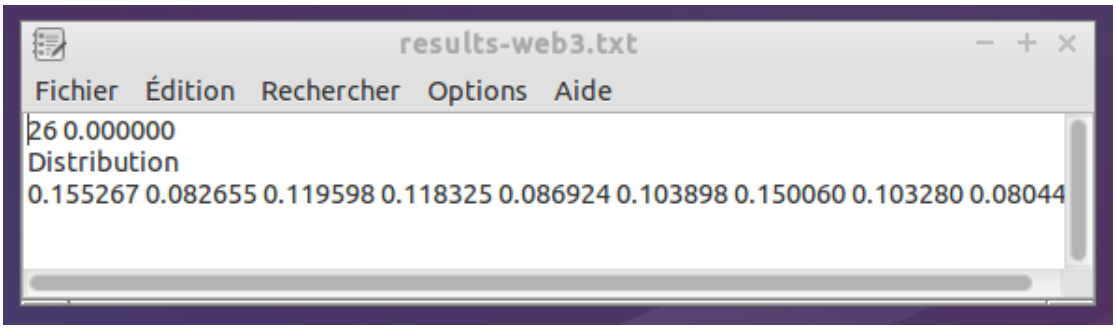
$\alpha = 0.85$, $\varepsilon = 10^{-6}$, $\sigma = \frac{f^T \cdot e}{n}$ et f le vecteur ligne tel que $f[i]$ vaut 1 si le degré sortant du noeur i de M est nul et 0 sinon.

Les distributions obtenues avec notre algorithme ont pu être vérifiées sur de petits graphes à l'aide du logiciel Scilab :




```
B  =

    0.    0.5   0.5   0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    1.    0.    0.    0.    0.
    0.    0.    0.    1.    0.    0.    0.    0.    0.
    0.5   0.    0.    0.    0.    0.5   0.    0.    0.
    0.    0.    0.5   0.    0.    0.5   0.    0.    0.
    1.    0.    0.    0.    0.    0.    0.    0.    0.
    0.    0.    0.    0.    0.    0.    0.    0.5   0.5
    0.    0.    0.    0.    0.    0.    1.    0.    0.
    0.    0.    0.    0.    0.    0.    0.667 0.334 0.
```




```
--> p = alpha * ( p * B) + ((1 - alpha)/9)
 p  =


         column 1 to 6

   0.1552675   0.0826553   0.1195979   0.1183249   0.0869237   0.1038973

         column 7 to 9

   0.1500637   0.1032817   0.0804437
```

2.2 Principe

L'algorithme effectue des multiplications vecteur-matrice à répétitions, cela dit, nous n'utilisons pas directement la matrice du web, mais on effectue deux transformations.

La première transformation, $M^{'} \leftarrow M + \frac{f^T \cdot e}{n}$, vise à rendre la matrice du web stochastique. C'est un élément critique pour justifier la convergence de l'algorithme.

La deuxième transformation, $G \leftarrow \alpha \cdot M^{'} + (1-\alpha) \cdot \frac{e^T \cdot e}{n}$, ajoute un bruit à la matrice pour accélérer la convergence de l'algorithme.

2.3 Complexité

En ce qui concerne la complexité, nous allons simplement énoncer ici que le nombre d'itérations de la boucle *do-while* de l'algorithme dépend de la convergence de la suite $\Pi^{(k+1)} = \Pi^{(k)} \cdot M$.

Soit $\{\lambda_1, \cdots, \lambda_n\}$ l'ensemble des valeurs propres de M trié par valeur décroissante. La suite des $(\Pi^{(k)})_{n \in \mathbb{N}}$ se comporte comme une suite géométrique de raison λ_2. Ainsi, après la transformation de M en la matrice G, on peut prouver que la convergence est accélérée en $\alpha \cdot \lambda_2$.

3 Accélération de Aitken quadratique

principe réduire la complexité en lambda3 de la suite pikplus1 = piK * M grâce à une estimation de la valeur de lambda 2

d'apres le sujet lambda estimé à l'aide de 3 termes successifs de Pi (on calcule lambda2 une fois et on l'utilise pour le reste de l'algo) estimer lambda 2 : Donner la formule pour l'estimation (pikplus2 - pikplus1) / (pikplus1 - pik) on pourra choisir pi0 pi1 et pi2 pour calculer lambda2 pb on sait pas comment implémenter ca

on doit estimer aussi u2 = beta2 * v2, v2 vecteur propre de M, => rappeler definition de vecteur propre pareil on sait pas comment calculer ces valeurs

Ensuite la fomule de recurrence devient pikplus1 = pik*G - u2 * lambda2 puissance(k) Et donc mtn suite devrait converger comme suite geometrique de raison lambda3

4 Résultats expérimentaux

4.1 Description des données

Pour étudier les performances des algorithmes, nous utilisons six graphes du web présentés dans le tableau ci-dessous :

Graphe du web	nb noeuds	nb arcs	taille
wb-cs-stanford	9 914	36 854	624 Ko
Stanford	281 903	2 312 497	37 Mo
Stanford-Berkeley	683 446	7 583 376	120 Mo
in-2004	1 382 908	16 917 053	277 Mo
wikipedia	1 634 989	19 753 078	309 Mo
wb-edu	9 845 725	57 156 537	1.06 Go

À la consultation de ce tableau, il est forcé de constater du volume que représente ces données, ce qui justifie le choix des implémentations qui a été fait.

Nous voudrions signaler que suite au téléchargement du fichier *Stanford-Berkeley* le nombre de lignes indiquées dans le fichier n'est pas correct, 68344 au lieu de 683446.

Le code source est en **langage C**, le contrôle de l'utilisation correcte de la mémoire a été effectué à l'aide de l'option **-g** du compilateur **gcc** et l'outil d'exécution **valgrind**. Cependant, l'utilisation de ces outils décuplent le temps de calcul, nous avons donc préparer une deuxième version sans debug pour l'utiliser arès avoir vérifier l'utilisation correcte de la mémoire.

4.2 Mesures de performances

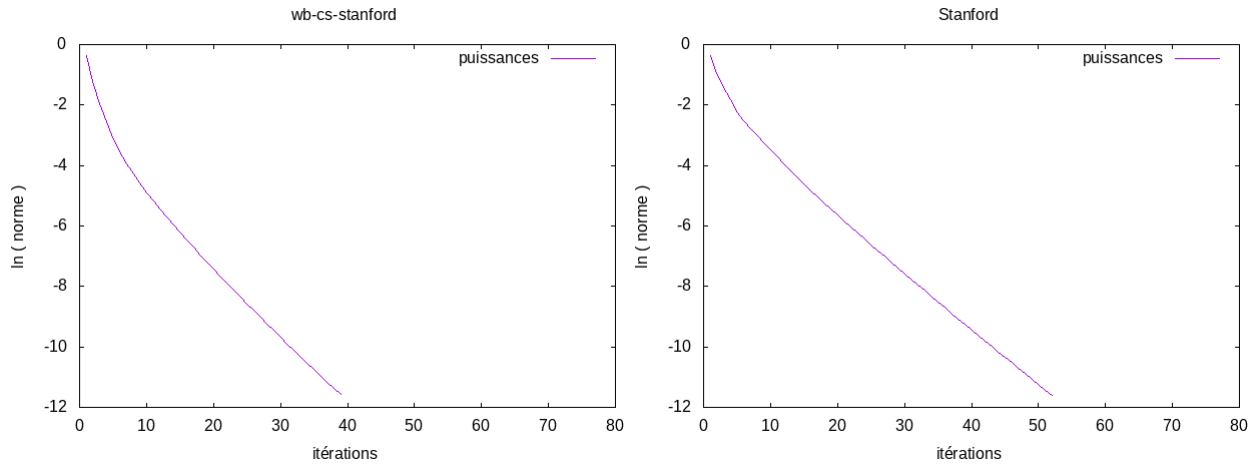
Nous vous présentons dans un premier temps les résultats en temps et en itérations de l'implémentation de la méthode des puissances et de l'accélération de aitken.

Graphe du web	nb noeuds	nb arcs	taille	Puis (ité)	Puis (sec)	Aitk (ité)	Aitk (sec)
wb-cs-stanford	9 914	36 854	624 Ko	x	x	x	x
Stanford	281 903	2 312 497	37 Mo	x	x	x	x
Stanford-Berkeley	683 446	7 583 376	120 Mo	x	x	x	x
in-2004	1 382 908	16 917 053	277 Mo	x	x	x	x
wikipedia	1 634 989	19 753 078	309 Mo	x	x	x	x
wb-edu	9 845 725	57 156 537	1.06 Go	x	x	x	x

commenter le tableau et décrire la machine utilisée RAM et Processeur

Il est aussi intéressant d'avoir un compte-rendu visuel pour comparer les méthodes. Les graphes ci-dessous représentent la valeur de $log(\|\Pi^{(k+1)} - \Pi^{(k)}\|_1)$ en fonction de l'itération k pour les deux méthodes implémentées pour les six graphes du web. Ces courbes ont été réalisées à l'aide du logiciel *GNUPLOT*.

On rappelle les constantes utilisées : $\alpha = 0.85$ et $\varepsilon = 10^{-6}$. Les courbes ont été ajustées à la même échelle pour pouvoir les comparer de manière visuelle.



Conclusion

Pour conclure, le bilan des fonctionnalités de l'application n'est pas satisfaisant. En effet, nous n'avons pas réussi à implémenter l'accélération de Aitken pour mesurer son comportement sur les exemples du sujet. Cela est dû à un manque de temps pour comprendre en détail les mathématiques sous-jacentes de l'accélération de Aitken, nous nous sommes donc retrouver dans l'impossibilité d'implémenter les calculs demandés.

Cela dit, le projet dans son ensemble a été une expérience enrichissante au niveau du travail, et de l'organisation. En effet, le point critique que représente la masse de données utilisées nous a imposé une rigueur au niveau de la structure de notre code (compilation et debug). Mais aussi, le projet nous

a permis de nous renforcer du point de vue de la démarche expérimentale et de la présentation des résultats.