Pagerank

Nicolas Chenciner & David Bühler

3 mars 2013

1 Utilisation

Compilation

Un makefile est fourni :

```
$ make
```

Exécution :

```
$ ./pagerank file [zap count epsilon z]
```

Options :

- file : nom du fichier contenant la description du graphe.
- zap : facteur zap (flottant) ; 0 par défaut.
- count : nombre maximum d'itérations de l'algorithme ; 0 par défaut.
- epsilon : distance maximale entre deux vecteurs successifs calculés par l'algorithme ; 0 par défaut.
- z : facteur influant sur le vecteur initial utilisé par l'algorithme :
  - si z est négatif, le vecteur initial est le vecteur « uniforme » dont chaque élément vaut 1/`length` (comportement par défaut).
  - si z est positif, le vecteur initial est le vecteur v tel que $v[z] = 1$ et $\forall i \neq z, v[i] = 0$.

Format de fichier

Les graphes sont décrits par des fichiers placés dans le répertoire `examples/` tels que :

- les lignes vides ou commencant par le caractère # sont ignorées ;
- les autres lignes doivent être de la forme « i j », symbolisant un arc du sommet i vers le sommet j.

2 Explication de code

2.1 Découpage des classes

Vect<T> Classe paramétrée représentant un vecteur de type T ; plusieurs constructeurs permettent d'initialiser facilement un vecteur, à partir d'une liste, d'un tableau ou d'une valeur par défaut.

FVect Vecteur de type float ; dispose des opérations d'addition et de multiplication par une constante, ainsi que du calcul de la norme entre deux vecteurs.

Matrix<T> Matrice de type T, dans le format décrit par l'énoncé, avec les méthodes permettant d'accéder aux valeurs ou de les modifier.

FMatrix Matrice de type float ; dispose des opérations de multiplication et de multiplication « transposée » par un `FVect`.

Graph Représentation d'un graphe, sous forme de hashmap. La méthode `stoch` calcule la matrice stochastique associée de manière optimale. Contient également la méthode statique `zapPagerank`.

GraphParser Lecture d'un fichier et construction du graphe associé.

Test Exécution d'un test : construction du graphe à partir d'un fichier, calcul de la matrice stochastique, exécution de l'algorithme *pagerank* avec les paramètres donnés, affichage des résultats.

MainTest Main, parsing des arguments, exécution du test.

3 Complexité

3.1 Multiplication d'une matrice par un vecteur

La méthode mult_naive effectue un appel à get pour chaque élément du tableau selon un parcours des lignes et des colonnes ; get étant de complexité $O(n)$ (potentiellement une opération par colonne), elle opère donc en $O(n^3)$, où n est la longueur de la matrice carrée.

La méthode mult est nettement plus efficace, puisqu'elle ne considère que les éléments non nuls de la matrice en cherchant directement dans les trois tableaux de la structure : elle effectue

– deux accès au tableau L pour chaque ligne ;

– pour chaque élément non nul, un accès à I, un accès à C, une multiplication et une addition.

Soit un nombre fixe d'opérations pour chaque ligne et chaque élément non nul.

Ce calcul est donc en $O(n + m)$ où n est le nombre de lignes de la matrice et m le nombre d'éléments non nul.