Algorithmes pour le Traitement de Données (ATD)
Examen 2016-2017, M1 Informatique

E. Gaussier

4 mai 2017

Il vous est demandé d'apporter le plus grand soin dans la rédaction de vos réponses, en particulier dans la justification des solutions.

Durée : 2h.
Documents autorisés.

1 Classification hiérarchique ascendante

On dispose d'un ensemble de 4 individus dont on donne le tableau de distances entre individus.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Question 1 Appliquer l'algorithme de classification hiérarchique avec la méthode du lien simple aux individus précédents en précisant à chaque étape la nouvelle table de distances. Dessiner le dendrogramme associé à cette classification ;

Question 2 Appliquer l'algorithme de classification hiérarchique avec la méthode du lien complet aux individus précédents en précisant à chaque étape la nouvelle table de distances. Dessiner le dendrogramme associé à cette classification ;

Question 3 Appliquer l'algorithme de classification hiérarchique avec la méthode du lien moyen aux individus précédents en précisant à chaque étape la nouvelle table de distances. Dessiner le dendrogramme associé à cette classification.

Remarque
Dans le cas où, dans une question, il y aurait plusieurs alternatives, donnez les toutes.

2 Calcul de similarité en grande dimension

On considère une collection de N documents textuels, avec un vocabulaire de M termes (le vocabulaire de la collection est défini comme l'ensemble des mots distincts apparaissant dans un ou plusieurs documents de la collection). Les documents sont indicés de 1 à N, et les mots de 1 à M (on ne s'occupe pas ici du passage d'une chaîne de caractères à un indice).

On suppose que la collection est représentée par un tableau de dimension N, tableau noté $Coll$. Chaque élément $Coll[d_i]$ $(1 \leq d_i \leq N)$ est une structure complexe, notée *Document*, qui contient divers champs représentant le document d_i :

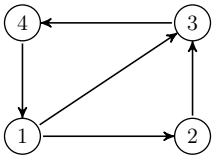
```
struct Document {
   int longueur;              /* longueur du document */
   int TabMots[longueur];     /* tableau contenant les indices des mots présents */
                              /* dans le document, par ordre croissant d'indices */
   int TabOcc[longueur];      /* tableau contenant les nombres d'occurrences de ces mots */
};
```

Question 4 Ecrire une fonction qui calcule le produit scalaire entre deux documents d'indices d_i et d_j. On utilisera la pondération présence/absence, qui prend la valeur 1 si le mot est présent dans le document, et 0 sinon.

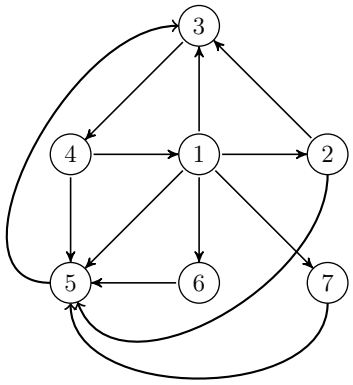
Question 5 Quelle structure doit être utilisée si l'on veut calculer efficacement le produit scalaire entre un document et tous les autres ?

3 Calcul du PageRank

On considère les deux graphes ci-dessous.



(a) Graphe A



(b) Graphe B

Question 6 Fixez λ à 0.6. Calculez la matrice de probabilité pour le graphe A.

Question 7 Fixez λ à 0. Calculez la matrice de probabilité pour le graphe B.

Question 8 Toujours pour le graphe B, calculez le PageRank de chaque page avec la méthode des puissances pour $\lambda = 0$.

La matrice de transition P est définie par :

$$P_{ij} = \begin{cases} \lambda \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}} + (1-\lambda)\frac{1}{N} & \text{si } \sum_{j=1}^{N} A_{ij} \neq 0 \\ \frac{1}{N} & \text{sinon} \end{cases}$$

où $A_{ij} = 1$ s'il existe un lien de la page i vers la page j, et 0 sinon.