Ingénierie de la fouille et de la visualisation des données
RCP 216 HTT, CNAM Paris
23 juin 2015
Examen de 3 heures
(Documents papier autorisés)

Consignes :
- Pensez à reporter votre numéro d'anonymat sur toutes les feuilles.
- Merci de soigner votre écriture et de ne pas rédiger au crayon papier sur votre copie.
- Eteignez vos téléphones portables. Ceux-ci ne sont pas autorisés, même pas pour servir d'horloge.

Ce sujet comporte 8 pages d'énoncé.

1 Fouille de données (13 points)

1. Peut-on faire un parallèle entre la matrice utilisateurs-articles (clients-articles) d'un système de recommandation et la matrice termes-documents (ou sa transposée) représentant une collection de textes ? Justifiez brièvement. (**1 point**)

   Peut-on envisager d'appliquer à la matrice utilisateurs-articles une décomposition inspirée de l'analyse sémantique latente ? Quelle serait son utilité pour un système de recommandation ? Expliquez. (**2 points**)

Correction :

Dans la matrice termes-documents, chaque document est décrit à travers l'ensemble des termes qu'il contient (représentés souvent par leur valeur TF-IDF). Les documents étant en général courts par rapport à l'ensemble de documents, leur représentation est très creuse. Dans la matrice utilisateurs-articles, chaque utilisateur est décrit à travers l'ensemble des articles qu'il a acquis (ou notés) ; l'information acquis / non acquis est binaire, alors que la note est une valeur numérique. Le nombre d'articles étant en général très élevé par rapport aux articles acquis (ou notés) par un utilisateur, la matrice est très creuse. Dans les deux cas, on peut considérer la matrice *incomplète* : un utilisateur pourrait acquérir (ou noter) à l'avenir des articles qu'il n'a pas encore (notés), répondant à un usage similaire à des articles déjà acquis (notés) ; un document pourrait être réécrit en utilisant d'autres mots tout en restant dans une même *thématique* et en gardant le même sens. Il est possible de faire un parallèle aussi entre la matrice termes-documents et la matrice articles-utilisateurs, avec l'avantage supplémentaire qu'un article fait partie d'un « groupe d'usage » (comme

un document aborde, en général, une thématique), alors qu'un utilisateur peut avoir un intérêt marqué pour plusieurs (et même nombreux) « groupes d'usages ».

L'idée de départ de l'analyse sémantique latente (et d'autres méthodes de même nature) est que chaque document aborde une ou plusieurs thématiques et à chaque thématique correspond un ensemble (ou plutôt une distribution) de mots ; identifier la (ou les) thématique(s) d'un document et les mots utilisables pour aborder une thématique permet d'obtenir des similarités entre documents de façon plus robuste qu'en considérant directement les occurrences des mots dans des documents précis. On peut ainsi réduire l'impact de l'emploi accidentel de mots peu usuels, on réduit l'impact des homonymes (qui peuvent contribuer à une confusion entre thématiques), on améliore la prise en compte de mots différents (par ex. « voiture », « automobile ») mais associés à une même thématique (« transports »). Une telle analyse peut être appliquée à la matrice utilisateurs-articles pour identifier de façon implicite des « groupes d'usages » et améliorer ainsi la recommandation.

---

2. Pour la classification automatique avec *streaming K-means* des données issues d'un flux de données nous avons le choix entre les deux approches suivantes :
   (a) Initialiser les centres avec *K-means*‖ au démarrage (peu après le démarrage du flux) et les faire évoluer seulement sous l'influence de l'algorithme de classification (re-calcul à chaque itération du centre de chaque groupe à partir des données affectées au groupe).
   (b) Initialiser les centres avec *K-means*‖ au démarrage et les ré-initialiser ensuite périodiquement, également avec *K-means*‖.

   Indiquez deux avantages de chacune de ces méthodes (par rapport à l'autre). (**3 points**)

   Proposez une autre approche d'initialisation combinant des avantages des deux approches mentionnées. (**2 points**)

Correction :

La première méthode proposée initialise les centres au démarrage (avec *K-means*‖) et les fait évoluer seulement à travers des itérations de *K-means*. La seconde méthode ré-initialise périodiquement les centres (avec *K-means*‖) et, entre deux ré-initialisations successives, garde les centres inchangés et affecte les nouvelles données aux centres.

La première méthode présente les avantages suivants (entre autres) par rapport à la seconde :

– Les centres évoluent à chaque itération et reflètent mieux la distribution des données lorsque celle-ci évolue peu (ou très lentement) et aucun nouveau groupe n'apparaît.

– Le coût des calculs reste le même entre itérations successives, alors que pour la seconde méthode les ré-initialisations sont bien plus coûteuses que les itérations ordinaires.

La seconde méthode présente les avantages suivants (entre autres) par rapport à la première :

– Lorsque la distribution des données n'est pas stationnaire (évolue fortement dans le temps, par exemple de nombreuses données arrivent dans des régions de l'espace où il n'y avait pas de données avant), une ré-initialisation périodique permet de mieux caractériser la distribution des données.
– Le coût des itérations ordinaires (entre deux re-initialisations successives) est faible car seuls sont réalisés les calculs nécessaires pour affecter les données aux groupes et on évite le recalcul de la position des centres.

(d'autres avantages respectifs, montrant une compréhension des processus, ont été bien notés)

Une approche d'initialisation combinant des avantages des deux méthodes mentionnées consisterait, par exemple, à

– effectuer périodiquement des ré-initialisations *partielles* : un centre existant est supprimé, un nouveau centre est généré par *K-means||* ; il est possible de choisir de façon plus informée le centre à supprimer ;
– lors de chaque itération ordinaire, la position d'un seul centre (ou de x% des centres) est mise à jour à partir des données nouvellement arrivées.

Cette approche permettrait de mieux suivre la distribution des données, tout en minimisant et lissant le coût des calculs. D'autres approches, bien motivées par l'analyse (réponses à la question précédente), ont été bien notées.

---

3. Une *startup* emploie le *crowdsourcing* pour obtenir des données à jour concernant les conditions de vie (coûts, sécurité, climat, etc.) dans le monde entier. Les (très nombreuses) personnes qui envoient des informations factuelles au site sont identifiées par une adresse de courriel ou un identifiant de réseau social. Certains contributeurs sont peu fiables ou manipulateurs. Peut-on employer un filtre de Bloom pour éviter de prendre en compte les informations provenant des contributeurs peu fiables déjà identifiés ? Expliquez avec vos propres mots ce qu'est un filtre de Bloom et comment il pourrait être utilisé pour résoudre ce problème. (**2 points**)

Correction :

Un filtre de Bloom est une structure de données permettant de vérifier (à faible coût mémoire et de calcul) si un élément est présent ou non dans un (très grand) ensemble. Le principe consiste à appliquer à la description d'une donnée une fonction de hachage qui prend un nombre élevé de valeurs différentes (pour minimiser

les collisions), à positionner à 1 la valeur de chaque case mémoire dont le numéro est le *hash* de chaque élément de l'ensemble, et ensuite à regarder si la case mémoire dont le numéro est le *hash* d'une donnée est à 1 ou à 0. Si la valeur est 1, la donnée est considérée comme faisant partie de l'ensemble (faux positifs possibles), si la valeur est 0 alors la donnée ne fait pas partie de l'ensemble (en revanche, pas de faux négatifs).

Dans le cas considéré, la fonction de hachage sera appliquée à l'identifiant de l'utilisateur (adresse de courriel ou identifiant de réseau social). L'ensemble visé (et connu) est celui des contributeurs peu fiables ou manipulateurs. Lorsqu'un message arrive, si la valeur de la case mémoire dont le numéro est le *hash* de l'identifiant de l'utilisateur est à 1, alors l'utilisateur est considéré membre de l'ensemble visé et son message n'est pas pris en compte.

---

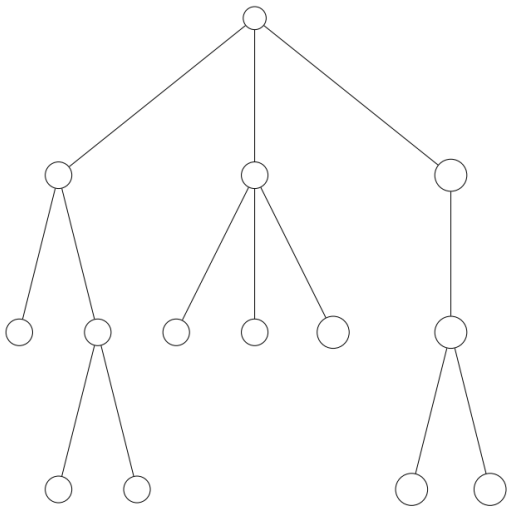
4. Quels sont les avantages et inconvénients majeurs de l'algorithme de clustering de Louvain, présenté dans le cours ? Expliquez ce qu'est la limite de résolution de cet algorithme. (**1 point**)

Correction :

- Avantages : vitesse, optimisation mémoire.
- Inconvénients : non déterminisme (stabilité du nombre de communautés), limite de résolution.
- Limite de résolution : l'optimisation de la modularité peut amener des communautés, même bien définies, à ne pas être repérées, car une autre partition a une modularité meilleure. Exemple de l'anneau des cliques ou des paires de cliques donné en cours.

---

5. Expliquez l'algorithme de parcours en largeur, en détaillant l'ordre de visite des noeuds, sur le graphe suivant. Puis, faites de même avec le parcours en profondeur. (**2 points**)



Correction **:**

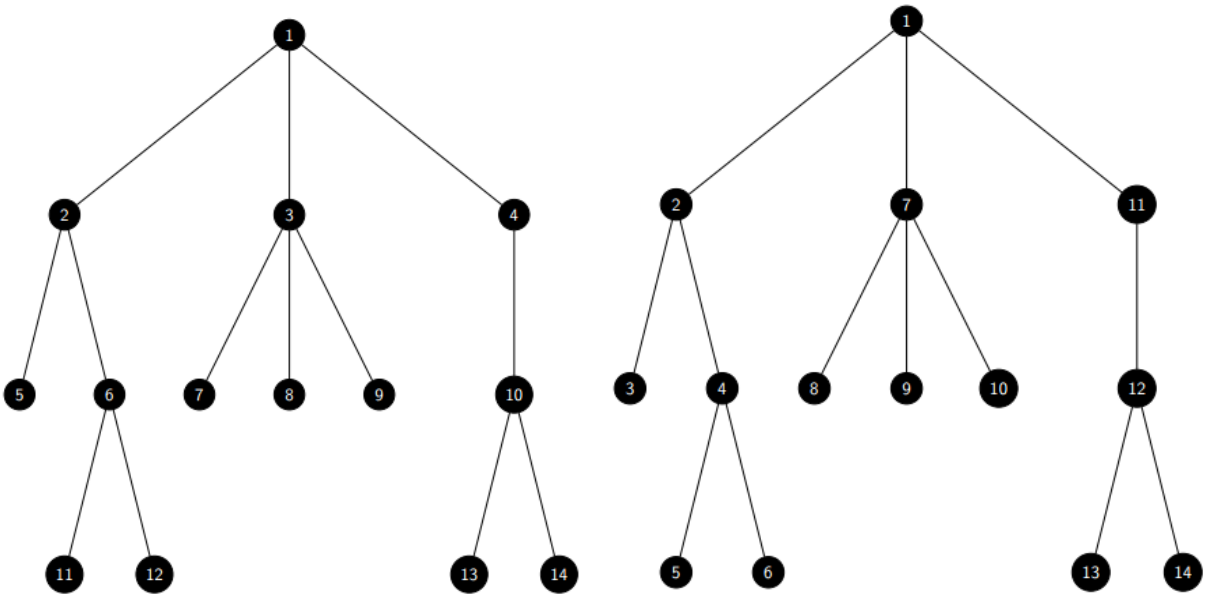


TABLE 1 – Parcours en largeur (à gauche) et en profondeur (à droite)

2 Visualisation et interaction (7 points)

1. Expliquer en quelques phrases pourquoi, dans le classement de Mackinlay, la couleur n'est pas associée à des tâches de manipulations de données quantitatives. (**1 point**)

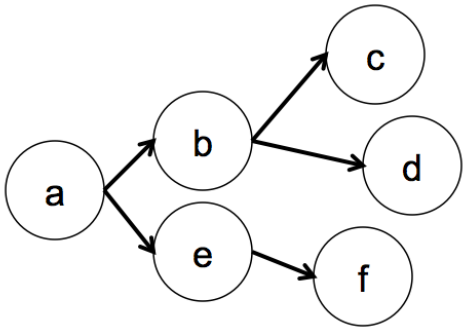
Correction :
Mackinlay place en effet la couleur en bas de son échelle de performance des attributs graphiques pour les données quantitatives (ex. : des températures, cours d'actions - cf cours 2).

La couleur est perçue par les humains selon une modalité tri-dimensionnelle : rouge, vert, bleu pour les cônes de l'oeil qui sont les premiers éléments de la chaîne de vision. La traduction de ces informations en une seule dimension comme la teinte n'obéit pas à une formule connue et invariante. On doit tenir compte des particularités physiologiques (le daltonisme par exemple). De plus, certaines gammes de teintes sont beaucoup mieux discriminées que d'autres (on peut aussi évoquer les facteurs culturels dans le classement des teintes).

Donc concrètement, on ne peut pas faire une correspondance universelle entre teinte et par exemple cours d'une action. On pourrait le faire entre la longueur d'onde d'une lumière et le cours d'une action, mais notre système visuel ne fonctionne pas comme un spectroscope. Pour comparer graphiquement des données quantitatives, il faut donc avoir recours à d'autres attributs. La longueur (utilisée dans les graphes en barres) est l'attribut le plus adapté.

---

2. Proposez quatre représentations graphiques de type différent pour l'arbre ci-dessous. Donnez le nom de chaque représentation et faites le dessin correspondant, en indiquant bien à chaque fois le nom des sommets. (**2 points**)



Correction :

Les représentations graphiques pour les arbres ont été vues au cours 3. Bertin résume les variantes connues de son temps (1965) dans un schéma.

La représentation Treemap a été inventée plus tard (1991) et les arbres hyperbolique en 1995. Pour l'arbre demandé, on peut retenir par exemple l'imposition rectiligne, l'imposition circulaire, le semis ordonné et le treemap - tous faciles à dessiner.

---

3. Un géographe du Cnam s'intéresse à la production viticole française. Il dispose d'une table (sous la forme d'un fichier CSV) contenant pour chaque zone de production les informations suivantes :
   – longitude et latitude de la zone (en fait, de son centre géographique),
   – production 2014 de vin, en hectolitres.

   On suppose que la table contient une centaine d'entrées. Le géographe veut produire une carte schématique où la production de chaque zone est représentée par un cercle plein dont la surface est proportionnelle à la production.

   Donner le code Processing pour produire une telle carte sauvegardée dans un fichier image. On ne demande pas forcément un code fonctionnel, mais que les étapes importantes des traitements soient bien décrites. (**2 points**)

Correction :

L'exercice s'inspire directement des exemples vu dans le TP 2 Cartographie. Voilà un pseudo-code :

1) créer une zone de dessin avec size(800,800) puis lire le fichier csv dans une table en mémoire (il existe des fonctions java qui font ça directement, la taille de la table n'est pas un enjeux ici)

2) rechercher les valeurs extrèmes (min et max) pour les latitudes, longitude et production (avec une boucle sur toute la table)

3) pour chaque entrée de table faire :

3.1) calculer une correspondance linéaire pour l'ordonnée X du cercle avec la longitude. On peut utiliser la fonction map() de Processing pour ça : X = map(longitude, minLong, maxLong, 0, width)

3.2) calculer une correspondance linéaire pour l'abscisse Y du cercle avec la latitude

Y = map(latitude,minlat,maxlat, 0, height)

3.3) calculer une correspondance linéaire pour le rayon R du cercle avec la racine carrée de la production (car la surface d'un cercle est elle-même proportionnelle au carré du rayon !)

R = map(sqrt(production),sqrt(minProd),sqrt(maxProd),5,50) // 5 et 50 obtenus par essai/erreur pour une meilleure lisibilité de la carte

3.4) dessiner un cercle de rayon R aux coordonnées (X,Y) avec ellipse(X,Y,2*R,2*R)
4) enregistrer le dessin dans un fichier : save("resultat.png") ;

---

4. Un spécialiste renommé de la visualisation d'information a proposé un canevas que doivent suivre les logiciels de visualisation interactive. Quel est ce chercheur et que propose-t-il ? Donner pour chacune des étapes qu'il propose un exemple dans le contexte de données de production viticoles de l'exercice précédent, mais cette fois à l'échelle mondiale. (**2 points**)

Correction :
Il est fait référence ici à Schneiderman et sa « mantra » : *"overview first, zoom and filter, then detail on demand"*. Pour une application de visualisation de données viticoles à l'échelle mondiale, on pourrait avoir :
- en vue globale de démarrage, une planisphère de type GoogleEarth (inconvénient : ce n'est pas une vue 100% globale, il faut tourner le globe) ou, mieux, une interface abstraire type Treemap
- le zoom peut être actionné sur des zones géographiques plus détaillées : le continent devient pays puis région etc ou sur des typologies de production (vin blanc vs vin rouge par ex.). Le filtrage se fait sur les paramètres inutiles dans le zoom (année de production par ex.)
- le « détail à la demande » serait ici une carte d'identité pour une appellation particulière d'un pays donné, avec des graphiques de chronologie de production par type de vins.

Schneiderman précise aussi les autres fonctions utiles d'un logiciel interactif de visualisation : mise en relation, historique de la navigation, fonction d'extraction (cf cours 4).
