le cnam

Sujet UE RCP216
Ingénierie de la fouille et de la visualisation des données

Anné universitaire 2015–2016

Examen 1e session : février 2016

Responsable : Michel CRUCIANU

Durée : 3h00

Tous documents autorisés.

Les téléphones mobiles et autres équipements communicants (PC, tablette, etc.) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve.

Sujet de 7 pages, celle-ci comprise.

Vérifiez que vous disposez bien de la totalité des pages du sujet en début d'épreuve et signalez tout problème de reprographie le cas échéant.

---

1 Fouille de données (14 points)

1. Supposons qu'un grand ensemble de n données est distribué sur m nœuds de calcul et nous souhaitons faire la classification automatique de ces données en k groupes disjoints. Pouvons-nous obtenir une classification de bonne qualité en exécutant *K-means* sur chacun des nœuds sur les données locales au nœud, afin d'obtenir env. k/m groupes par nœud, et ensuite en retournant la totalité de ces groupes ? Expliquez brièvement. Voyez-vous un cas favorable à cette approche ? Voyez-vous une variante de cette approche qui soit plus performante tout en restant « simple » ? Expliquez brièvement. (**3 points**)

   En considérant que les données arrivent en flux, proposez une méthode de répartition des données sur les nœuds de calcul favorisant la méthode de classification simple (ou simpliste ?) proposée ci-dessus. (**1 point**)

Correction :

Cela revient à superposer m classifications indépendantes concernant chacune n/m données différentes. Si la distribution des données sur les nœuds de calcul ne tient pas compte de la similarité entre données alors le résultat sera médiocre : un même groupe « naturel » de données sera réparti entre plusieurs centres (issus de nœuds de calcul différents) et chaque centre pourra regrouper des parties de plusieurs groupes « naturels » (trop de centres par nœud). Un cas favorable est celui où chaque nœud de calcul reçoit dès le départ toutes les données correspondant à plusieurs groupes « naturels » ; la probabilité pour que cela arrive par hasard est très faible, en revanche il est possible de se diriger vers cette distribution progressivement, en traitant par ex. les données d'un flux.

Lorsque les données arrivent en flux, il est possible de les répartir de façon à renforcer leur « localité » par rapport aux groupes « naturels » et ainsi favoriser cette solution simple de classification. Pour cela, on génère au départ k centres, on les regroupe en k/m groupes et on affecte chaque groupe de centre à un nœud de calcul. Lorsqu'une nouvelle donnée arrive, elle est dirigée vers le nœud qui contient le centre dont la donnée est la plus proche.

---

2. Une des représentations vectorielles populaires pour des documents textuels est le résultat de l'analyse sémantique latente, qui fait appel à une décomposition en valeurs singulières (SVD) de la matrice termes-documents (ou de sa transposée, la matrice documents-termes). Voyez-vous un intérêt à appliquer une *factorisation régularisée* plutôt que la SVD pour obtenir les représentations des documents et des termes ? Voyez-vous un désavantage de la factorisation régularisée dans ce cas ? Expliquez brièvement. (**3 points**)

Correction :

La SVD considère que les mots absente d'un texte ne doivent pas y être présents car non pertinents par rapport au sujet traité dans ce texte, alors qu'une factorisation régularisée considérerait plutôt que certains des mots absents du texte peuvent être pertinents par rapport au sujet. La factorisation régularisée introduirait donc moins de contraintes et pourrait, par exemple, gérer mieux la synonymie. Un désavantage de la factorisation régularisée dans ce cas serait l'augmentation de la complexité.

---

3. Le propriétaire d'un site web souhaite afficher des nouvelles provenant d'un grand nombre de flux d'informations. Chaque nouvelle comporte un titre, un texte et éventuellement une image. Le propriétaire du site souhaite pouvoir filtrer à faible coût, à partir des titres, certaines nouvelles qui abordent des sujets ou thématiques répertoriées. Expliquez comment procéder à l'aide d'un filtre de Bloom. Quelles propriétés doivent avoir les fonctions de hachage et les représentations employées pour les titres ? (**3 points**)

Correction :

L'utilisation directe d'un filtre de Bloom (voir le cours) avec des fonctions de hachage quelconques permettrait de filtrer les titres *signalés au départ* comme abordant des sujets ou thématiques répertoriées, les nouvelles ultérieures auraient une faible probabilité d'être filtrées car elles n'auraient pas les mêmes titres que celles signalées au départ. Pour permettre un filtrage de ces nouvelles ultérieures, les représentations employées pour les titres et les fonctions de hachage associée doivent être *sensibles à la similarité thématique* ; ainsi, la probabilité de collision entre des titres correspondant à une thématique répertoriée serait bien plus élevée qu'avec des fonctions de hachage quelconques.

---

4. Réseaux sociaux.

   Quels sont les biais de mesure auxquels on doit s'attendre quand on étudie un graphe réel (comme celui des routeurs d'Internet) ? Expliquez. (**1 point**)

   En raisonnant sur un graphe (dont les liens modélisent le fait que deux personnes soient amies), montrez que dans un groupe de n personnes, il y a toujours deux personnes qui ont le même nombre d'amis. (**1 point**)

   En pseudo-code ou dans un langage de programmation courant de votre choix (parmi Python, Scala, Java, C, C++, Ruby), présentez un algorithme de parcours en profondeur d'un graphe (algorithme itératif ou récursif). (**2 points**)

Correction :

Une opération de "mesure" d'un graphe consiste à déterminer expérimentalement sa structure : quels sont les noeuds et les liens qui le composent. Compte tenu de leur taille et de leur évolution rapide (en général), la capture s'avère délicate. Le processus de mesure peut par exemple être égocentré et conduire à une mauvaise estimation de la distribution réelle des degrés (exemple des noeuds de degré 1), ignorer des liens existants (*cf.* traceroute et loadbalancing). Il est également fréquent de ne pouvoir observer qu'une zone restreinte du réseau global, ou la totalité du réseau mais sur une durée courte.

Construisons un graphe dont les sommets représentent les personnes et plaçons une arête entre deux sommets lorsque les personnes correspondantes sont amies. Dire que deux personnes ont le même nombre d'amis revient à dire que deux sommets dans le graphe ont même degré... Nous allons montrer qu'il n'existe aucun graphe dont tous les sommets ont des degrés distincts. Supposons qu'un tel graphe existe et qu'il possède n sommets. Le degré maximal d'un sommet est donc $n-1$. Si tous les degrés des sommets sont distincts, on a donc nécessairement un sommet de degré 0, un sommet de degré 1,..., un sommet de degré $n-1$. Du fait de la présence d'un sommet de degré 0, disons x_0 , il est impossible d'avoir un sommet de degré $n-1$! (en effet, celui-ci devrait être relié à tous les autres, y compris x_0). On obtient ainsi une contradiction.

En Python :

```
graph = {0: [1,5],
         1: [0,2,3],
         2: [1,4],
         3: [1,4,5],
         4: [2,3,5],
         5: [0,3,4]
         }

def dfsIter(graph, root):
    visited = []
    stack = [root, ]

    while stack:
        node = stack.pop()

        if node not in visited:
            visited.append(node)
            # on va stocker la liste des voisins du noeud dans une
```

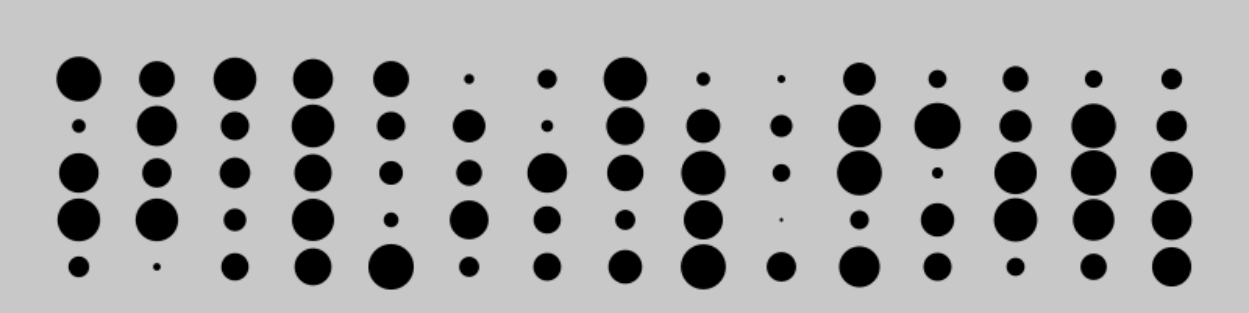
```
            # pile "à visiter", excepté ceux qui l'ont déjà été
            stack.extend([x for x in graph[node] if x not in visited])

    return visited
```

---

2 Visualisation et interaction (6 points)

On souhaite visualiser l'activité d'élèves dans un cours en ligne du Cnam. Le diagramme ci-dessous représente les notes obtenues par 5 élèves pour 15 exercices. Les notes, valeur réelles, varient de 0 à 5. Les cercles noirs ont une surface proportionnelle à la note. Quand la note est 0, un petit point noir est dessiné quand même (il y en a un dans la dixième colonne).



1. Quelle est la variable rétinienne qui est utilisée ici ? Donner un autre attribut possible plus efficace (au sens de Mackinlay) et un autre moins efficace. (**1 point**)

Correction :
Selon la classification de Bertin (cours 2, diapositive 43), c'est la variable de taille qui est utilisée ici. Mackinlay a affiné cette classification : il distingue la longueur, la surface et le volume. En utilisant les expériences de Cleveland et McGill, il a montré que, pour des données d'ordre quantitatif, le codage par surface était moins efficace que par exemple la longueur ou l'angle et plus efficace que la couleur ou la forme (cours 2, diapositive 46).

---

2. (**3 points**) Écrire de la manière la plus détaillée possible un programme Processing en mode immédiat produisant l'image ci-dessus. Si vous ignorez les instructions exactes (nom, syntaxe), décrivez-en le rôle clairement. On supposera que les notes

sont stockées dans une matrice appelée `Note` de réels de dimension `NEL*NEX`, où `NEL` est le nombre d'élèves et `NEX`, le nombre d'exercices. Pour simplifier, on initialisera ce tableau de nombres aléatoires avec la fonction `random(0,5)`. L'image a une largeur de 800 pixels et une hauteur de 400 pixels. Le dessin commence au pixel (25,25). Le fond est gris, de valeur 220. Attention : pour calculer la dimension des cercles, il faut faire une conversion entre la note et la surface du cercle, pas son diamètre. A la fin du programme, l'image est enregistrée dans un fichier appelé `examRCP216.png`.

Correction **:**

Un code possible, inspiré de ceux vus au TP2 de visualisation :

```
int NEL = 5;
int NEX = 15;
size(800,400);
float note[][] = new float[NEL][NEX];
for (int i=0;i<NEL;i++)
  for (int j=0;j<NEX;j++)
    note[i][j] = random(0,5);
float dx = float(width-50)/NEX;
float dy = float(height-50)/NEL;
smooth();background(220);
fill(0);noStroke();
float x=25;
float y=25;
for (int i=0;i<NEL;i++){
  for (int j=0;j<NEX;j++){
    float surf = map(note[i][j],0,5,1,dx*dx);
    ellipse(x,y,sqrt(surf),sqrt(surf));
    x+=dx;
  }
  x=25;y+=dy;
}
save("exo2visu.png");
```

---

3. Pour mieux analyser ces données, il faut rendre le programme interactif. Une invention de J. Bertin est bien adaptée ici : laquelle ? Donner un exemple (parmi ceux vus en cours) d'utilisation de ce procédé. (**1 point**)

Correction :
Pour analyser ce type de données matricielles de manière interactive, on peut utiliser la technique des matrices ordonnables de Bertin (cours 3, diapositive 5 et suiv.) ? Par des permutation manuelle des lignes ou des colonnes de la matrice représentée graphiquement, on arrive à faire émerger des groupes. Deux exemples cité en cours : classification de marques de "corn flakes" selon leur caractéristiques nutritives, classification de photos personnelles selon leur typologie (photo de groupe, panoramique, etc).

---

4. Ces données peuvent devenir très nombreuses dans le cas de cours en ligne de type MOOC (Massive Online Open Course). Par exemple, un MOOC du Cnam a compté jusqu'à 30000 élèves. Pour visualiser correctement notre diagramme dans ce cas, il faut un outil interactif permettant la visualisation conjointe du contexte global et d'une zone plus détaillée. Proposer une technique parmi celles vues en cours. (**1 point**)

Correction :
Le cours 4 est consacré pour l'essentiel à cette question. Parmi les technique d'interaction permettant la visualisation conjointe du contexte global et d'une zone plus détaillée, on a par exemple la distorsion (ex. du Perspective Wall) et les lentilles déformantes (diapositive 24 et suiv.). Les interfaces zoomables multi point de vue (diapositive 16 et suiv.) constituent une alternative.
