Analyse de grands jeux de données : classification et analyses multivariées

M. Bailly-Bechet

Université Nice Sophia Antipolis – France

1 Introduction

On a parfois affaire à de grands jeux de données. Ce n'est pas grave [1], mais cela peut le devenir si on ne sait pas quelles questions poser. En particulier un grand nombre de variables peut devenir problématique, si on n'a pas d'idée sur quelles relations chercher. Il existe des techniques pour structurer et ordonner des données dites de grande dimension, dont nous allons voir quelques exemples simples dans cette fiche : classification et analyse multivariée.

Le jeu de données que l'on va étudier ici est issu d'une publication (Martin et al. 2007, Hepatology 45 :767-777) [2] Pour 40 souris, nous disposons :

- des données d'expression de 120 gènes recueillies sur membrane nylon avec marquage radioactif,
- des mesures de 21 acides gras hépatiques.

Par ailleurs, les 40 souris sont réparties selon deux facteurs :

- Génotype (2 modalités) : les souris sont soit de type sauvage (wt) soitgénétiquement modifiées (PPAR) ; 20 souris dans chaque cas.
- Régime (5 modalités) : les 5 régimes alimentaires ; 4 souris de chaque génotype sont soumises à chaque régime alimentaire.

Téléchargez le jeu de données à l'adresse `http://www.unice.fr/mbailly/docs/TP/nutrimouse.rda`, et chargez le sous R ainsi :

```
load("nutrimouse.rda")
ls()
```

```
[1] "nutrimouse"
```

```
names(nutrimouse)
```

```
[1] "gene"     "lipid"    "diet"     "genotype"
```

Regardez ce que contient chaque élément de `nutrimouse` et faites le lien avec la description ci-dessus.

1. Le statisticien demande en permanence plus de données, il ne va pas (trop) se plaindre quand il y en a beaucoup

2. Et en pratique du package mixOmics, avec une aide énorme fournie par le TP rédigé sur Wikistat `http://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-scenar-omic-nutri-mouse.pdf`

2 Classification

L'objectif de la classification (dans notre cas "non supervisée", la classification supervisée impliquant l'existence d'un jeu de données d'apprentissage) est de diviser les observations en k groupes, supposant l'existence de k sous-populations homogènes dans les données. Les deux éléments clefs de toute démarche de classification sont :

— le choix de la distance utilisée pour calculer la dissimilarité entre deux observations,

— le choix du nombre de groupes k.

De plus, pour une distance et une nombre de groupes donnés, la démarche de classification peut prendre plusieurs formes et donc donner des résultats différents. Deux méthodes parmi les plus classiques sont les k-means et la classification hiérarchique ascendante que nous allons voir ici.

2.1 Réallocation dynamique : les k-means

Dans la méthode des k-means, le nombre de groupes k est fixé au départ. Généralement, si l'utilisateur n'a pas d'idée précise du nombre de groupes à rechercher, il essaiera plusieurs classifications et choisira *a posteriori* celle qui fait le plus de sens (il existe des méthodes pour choisir le nombre de groupes *a posteriori*, hors du cadre de cette initiation).

L'idée des k-means est de regrouper les données x autour de k centroïdes, des "points moyens" qui représentent les k groupes. Si on suppose que l'on a commencé la procédure et que l'on a déjà k groupes de tailles et leurs k centroïdes, on peut calculer le ratio entre la somme des distances entre centroïdes et la somme des distances de chaque observation au centroïde du groupe auquel elle est affectée (ce qui vous rappelera peut-être la décomposition de variances inter et intra et l'ANOVA). Une bonne classification donne une valeur élevée de ce ratio ; une mauvaise, une valeur faible. La méthodes des k-means consiste à choisir aléatoirement une classification initiale, puis à vérifier si la réallocation d'une observation à un autre groupe que le sien améliore ce critère : si oui on effectue le changement, si non on garde la classification existante. On peut montrer que, à condition d'essayer de nombreuses configurations de départ, on va finalement trouver la classification qui maximise ce critère.

Sur nos données :

```
km<-kmeans(nutrimouse$gene,centers=5,nstart=10)
names(km)
### Quelle est la taille des groupes?
km$size
### Qui va dans quel groupe?
km$cluster
### Les 5 groupes correspondent-ils aux 5 régimes?
table(km$cluster,nutrimouse$diet)
```

2.2 Classification hiérarchique ascendante

En classification hérarchique ascendante, le nombre de groupes n'est pas fixé au départ. L'idée est de voir les n observations comme n groupes. On peut alors regrouper les 2 observations les plus

proches, obtenant ainsi $n-1$ groupes. Si on est capable de généraliser la distance employée pour calculer l'écart entre deux observations au cas où on veut comparer l'écart entre deux groupes (et ce n'est pas forcément trivial, il existe de nombreuses méthodes), on peut itérer la procédure : on regroupe ensuite les 2 observations (ou groupes) les plus proches, passant à $n-2$ groupes, etc... jusqu'au dernier regroupement qui rassemble toute les données en un seul groupe. On va alors pouvoir visualiser les regroupements sous la forme d'un dendrogramme (un "arbre"), et choisir le nombre de groupes ici aussi en fonction des résultats obtenus.

Sur nos données :

```
### Une première classification
### avec la distance euclidienne
d<-dist(nutrimouse$gene)
arbre<-hclust(d,method="ward.D2")
plot(arbre)
### On choisit par exemple 2 groupes
groups<-cutree(arbre,2)
### Les 2 groupes ont-ils des génotypes différents?
table(groups, nutrimouse$genotype)
### Une deuxième méthode
### avec la corrélation comme distance
d2<-1-(cor(t(nutrimouse$gene)))**2
arbre2<-hclust(as.dist(d2),method="ward.D2")
plot(arbre2)
groups2<-cutree(arbre2,k=2)
table(groups2, nutrimouse$genotype)
### Une comparaison avec les k-means
groups3<-cutree(arbre,5)
table(groups3,km$cluster)
```

3 Analyse multivariée

Ces techniques sont nombreuses, et ont comme objectif d'extraire le maximum d'information d'une jeu de données complexes. L'analyse multivariée permet de visualiser les données de manière simplifiée. De plus, elle permet de vérifier, dans les données, quelles dimensions (donc quelles variables) donnent des informations redondantes, et donc de trier parmi les variables lesquelles sont intéressantes pour poursuivre l'analyse.

Le choix de la technique à employer dépend du type de données étudiées et de la question à poser. Ici, on voit les deux techniques les plus courantes, l'analyse en composantes principales (pour les données quantitatives) et l'analyse factorielle des correspondances (pour les tableaux de contingence issus du comptage de variables qualitatives).

3.1 Analyse en composantes principales (ACP)

L'analyse en composantes principales est une des principales techniques d'analyse multivariée. Prenons par exemple les expressions de gènes dans notre jeu de données d'exemple. Chacun des 120 gènes est caractérisé par 40 valeurs, qui sont ses 40 expressions dans 40 conditions différentes. Si on voulait représenter l'ensemble de nos données, on devrait donc tracer un nuage de points à 40 dimensions, ce qui est visuellement et conceptuellement difficile à manipuler. L'objectif de l'analyse en composantes principales est de représenter ces données sur beaucoup moins de dimensions – traditionnellement 2 ou 3, pour que ce soit visuellement pratique – en perdant le moins d'information possible. Pour cela, la technique recherche les *composantes principales*, qui sont justement les dimensions le long desquelles on a le maximum de variabilité dans les données. Il faut noter que ces composantes principales ne sont pas forcément des dimensions présentes au départ, mais peuvent être n'importe quelle combinaison linéaire de celles-ci !

Sur nos données :

```
library(ade4)
### On choisit pour l'exemple
### de projeter sur 2 dimensions
pca<-dudi.pca(nutrimouse$gene,scann=F,nf=2)
### Combien de dimensions aurait-il fallu choisir?
barplot(pca$eig)
### Le graphe combinant gènes et souris
scatter(pca)
### Le graphe des individus, sur le même plan
plot(pca$li)
colors<-as.numeric(nutrimouse$genotype)
plot(pca$li, pch=19, col=colors)
legend("bottomright",legend=levels(nutrimouse$genotype),pch=19,col=1:2)
### Le graphe des variables, sur le même plan
s.corcircle(pca$co)
```

3.2 Analyse factorielle des correspondances (AFC)

L'AFC est une technique similaire à l'ACP, mais qui se pratique sur des tables de contingence. Très utilisée en sociologie et en écologie, elle permet notamment de visualiser les liens entre variables qualitatives.

L'exemple sur lequel on va travailler ici est différent, le premier ne se prêtant pas à ce type d'analyse. Cet exemple porte sur la couleur des yeux et la couleur des cheveux de 592 étudiants. Les données ont été collectées dans le cadre d'un projet de classe par les étudiants d'un cours de statistique élémentaire à l'Université de Delaware [3] .

3. Exemple intégralement issu de la fiche `https://pbil.univ-lyon1.fr/R/pdf/tdr620.pdf` dont je remercie vivement les auteurs !

```
snee74 <- read.table("http://pbil.univ-lyon1.fr/R/donnees/snee74.txt", header=TRUE)
names(snee74)
head(snee74)
attach(snee74)
```

La couleur des cheveux est définie par 4 modalités : blond, marron, noir et roux. La couleur des yeux est définie par 4 modalités : bleu, marron, noisette et vert.

```
summary(cheveux)
summary(yeux)
```

Le lien entre les deux couleurs s'obtient à l'aide d'un tableau croisé qui ventile la population entre les modalités de ces deux variables qualitatives. C'est une table de contingence.

```
table(yeux,cheveux) -> couleurs
couleurs
plot(couleurs)
```

L'AFC s'effectue de manière très similaire à l'ACP :

```
coa<-dudi.coa(unclass(couleurs),scann=F,nf=2)
scatter(coa)
```