SY09 Printemps 2015
TP 2
Analyse factorielle d'un tableau de distances, classification automatique

Exercice 1. Analyse factorielle d'un tableau de distances

On considère le tableau de données suivant :

$$X = \begin{pmatrix} 8.5 & 1.5 \\ 3.5 & 5.0 \\ 2.0 & 6.5 \\ 9.5 & 1.5 \\ 8.5 & 2.5 \\ 3.0 & 6.5 \\ 9.0 & 2.5 \\ 2.0 & 5.5 \end{pmatrix}.$$

1. Calculer le tableau D^2 des distances euclidiennes associé à ces données.
2. Calculer la matrice W des produits scalaires : d'une part directement à partir de X, d'autre part à partir de D^2.
3. Vérifier que W (ou $\frac{1}{n}W$) est semi définie positive.
4. Déterminer la matrice Λ des vecteurs propres (normés au sens de D_p) de $\frac{1}{n}W$ et L la matrice diagonale des valeurs propres.
5. En déduire la représentation multidimensionnelle fournie par l'AFTD.
6. Tracer le nuage associé au tableau initial X et le nuage associé à la représentation fournie par l'AFTD ; comparer ces deux représentations.
7. En utilisant les résultats précédents, écrire en R la fonction `aftd` ayant comme argument d'entrée la distance D et comme argument de sortie la représentation multidimensionnelle calculée par l'AFTD ainsi que la qualité de cette représentation.

Données de mutations. Les principaux outils permettant d'appliquer l'AFTD sont les suivants :
- la classe `dist` permet de manipuler les tableaux de proximités (distance, dissimilarités, similarités). Les fonctions `as.dist` et `as.matrix` permettent de passer de la représentions matricielle classique à la représentation en tableau de distance.
- La fonction `dist` permet d'effectuer le calcul des principales distances (euclidienne, ...) à partir d'un tableau de données.
- La fonction `cmdscale` réalise l'AFTD pour trouver une représentation associée à un tableau de distances dans un espace de dimension d à spécifier (paramètre `k` de la fonction),
- enfin, le module `Shepard` (bibliothèque MASS) permet de tracer un diagramme de Shepard.

On se propose d'utiliser ces outils sur les données de mutations (fichier mutations2.txt). Charger les données et les stocker dans un tableau de données, puis convertir la variable en tableau de distances :

```
mutations <- read.table("mutations2.txt", header=F, row.names=1)
mutations <- as.dist(mutations)
```

1. Effectuer l'AFTD en utilisant la fonction `AFTD` écrite dans l'exercice 1 et en utilisant la fonction `cmdscale`. Comparer les résultats alors obtenus.
2. Effectuer l'AFTD avec un nombre de variables d de représentation allant de 2 à 5. À chaque fois, calculer la qualité de la représentation, et tracer les diagrammes de Shepard. Interpréter les résultats.

```
+--------------------------------------------------------------------+
|     MUTATION DISTANCES AMONG 20 SPECIES (FITCH AND MARGOLIASH)     |
|                                                                    |
|     The source of this data is a paper by Fitch and Margoliash     |
| in Science(1967).   For a more recent reference see Scientific     |
| American (1972?).                                                  |
| Every species has a protein  molecule, Cytochrome c, which varies  |
| from species to species but has a similar function for all. It     |
| consists of a long chain of  amino acids. There are only a few     |
| acids,  but different  molecules are  obtained  by varying  the    |
| acids in each  position in the chain. The  number of positions     |
| with different  acids measures  distance between  two species.     |
| these distances are given in the data below.                       |
| For example, the amino acids in Cytochrome c for two species look  |
| like this:                                                         |
|  Moth            XXYVPLY .........SEXI                             |
|  Screwworm fly   XXYVPLY .........LSEI                             |
| where the whole chain is 110 in length, and the letters represent  |
| particular amino acids. Each difference contributes to mutation    |
| distance according to the minimum number of nucleotides that would|
| need to be changed to convert one into the other.                  |
| Fitch & Margoliash used these data to construct a phylogenetic     |
| tree.                                                              |
| Ref: Science, v. 155, 279-284.                                     |
+--------------------------------------------------------------------+

Man                 0
Monkey             01  0
Dog                13 12  0
Horse              17 16 10  0
Donkey             16 15 08 01  0
Pig                13 12 04 05 04  0
Rabbit             12 11 06 11 10 06  0
Kangaroo           12 13 07 11 12 07 07  0
Pekin Duck         17 16 12 16 15 13 10 14  0
Pigeon             16 15 12 16 15 13 08 14 03  0
Chicken            18 17 14 16 15 13 11 15 03 04  0
King Penguin       18 17 14 17 16 14 11 13 03 04 02  0
Snapping Turtle    19 18 13 16 15 13 11 14 07 08 08 08  0
Rattlesnake        20 21 30 32 31 30 25 30 24 24 28 28 30  0
Tuna               31 32 29 27 26 25 26 27 27 27 26 27 27 38  0
Screwworm Fly      33 32 24 24 25 26 23 26 26 26 26 28 30 40 34  0
Moth               36 35 28 33 32 31 29 31 30 30 31 30 33 41 41 16  0
Bakers Mould       63 62 64 64 64 64 62 66 59 59 61 62 65 61 72 58 59  0
Bread Yeast        56 57 61 60 59 59 59 58 62 62 62 61 64 61 66 63 60 57  0
Skin Fungus        66 65 66 68 67 67 67 68 66 66 66 65 67 69 69 65 61 61 41 0
```

Exercice 2. Méthode des centres mobiles

Le but de cet exercice est de tester les performances de l'algorithme des centres mobiles sur trois jeux de données réelles : Iris, Crabs et mutations.

Données Iris

1. Tenter une partition en $K \in \{2, 3, 4\}$ classes avec la fonction `kmeans` ; visualiser et commenter.
2. On cherche à présent à étudier la stabilité du résultat de la partition. Effectuer plusieurs classifications en $K = 3$ classes du jeu de données. Observer les résultats, en termes de classification obtenue et d'inertie intra-classes. Ces résultats sont-ils toujours les mêmes ? Commenter et interpréter.
3. On cherche à déterminer le nombre de classes optimal.
   (a) Effectuer $n = 100$ classifications en prenant $K = 2$ classes, puis $K = 3$ classes, $K = 4$ classes, ..., jusqu'à $K = 10$ classes. On constitue ainsi neuf échantillons iid $\{I_{K\,1}, \ldots, I_{K\,100}\}$ contenant 100 valeurs d'inertie intra-classe chacun.
   (b) Pour chaque valeur de K, calculer l'inertie intra-classe minimale $\widehat{I_K} = \min_{i=1,\ldots,100} I_{K\,i}$. Représenter la variation d'inertie minimale en fonction de K (on inclura l'inertie totale, correspondant à l'inertie intra-classe lorsque $K = 1$).

   Proposer un nombre de classes en se basant sur ces informations.
4. Comparer les résultats de la partition obtenue par les centres mobiles avec la partition réelle des iris en trois groupes.

Données Crabs

Charger les données Crabs et effectuer le pré-traitement suivant :

```
library(MASS)
data(crabs)
crabsquant <- crabs[,4:8]
crabsquant <- crabsquant/matrix(rep(crabsquant[,4],dim(crabsquant)[2]),
    nrow=dim(crabsquant)[1],byrow=F)
crabsquant <- crabsquant[,-4]
```

1. Effectuer la classification des données Crabs au moyen de l'algorithme des centres mobiles.
2. Comparer à la partition réelle des crabes suivant l'espèce et le sexe.

Données mutations

On calculera tout d'abord une représentation des données mutations dans un espace de dimension $d = 5$, sur laquelle on pourra utiliser la fonction `kmeans`.

1. Effectuer plusieurs classifications de cette représentation en $K = 3$ classes au moyen de l'algorithme des centres mobiles. On pourra représenter les résultats obtenus dans le premier plan factoriel de l'AFTD.
2. Étudier la stabilité du résultat de la partition. Commenter et interpréter.