

Classification

Mathieu Pigeon

UQAM

1 Introduction

2 Concepts de base

3 Algorithme

- Classification par ré-allocation dynamique
- Algorithmes hiérarchiques ascendants

4 Applications

- Application 1 : Eurojobs
- Application 2 : Uber
- Application 3 : Données télématisques

Classification non supervisée

- **Objectif principal** : on cherche une répartition des observations en classes homogènes (ou catégories) en optimisant un critère assurant de regrouper des observations dans des classes, chacune la plus **homogène** possible et, entre elles, les plus **distinctes** possibles.
- Attention aux termes utilisés : la classification (en anglais on parlera souvent alors de *clustering*) est une technique d'apprentissage non supervisée et doit être distinguée du classement, ou discrimination (en anglais on dira alors *classification*) qui est une technique d'apprentissage supervisée.

Base de données

Comment regrouper des pays dont la structure économique est similaire ?

| | Agr | Min | Man | PS | Con | SI | Fin | SPS | TC |
|------------|------|-----|------|-----|-----|------|-----|------|-----|
| Belgium | 3.3 | 0.9 | 27.6 | 0.9 | 8.2 | 19.1 | 6.2 | 26.6 | 7.2 |
| Denmark | 9.2 | 0.1 | 21.8 | 0.6 | 8.3 | 14.6 | 6.5 | 32.2 | 7.1 |
| France | 10.8 | 0.8 | 27.5 | 0.9 | 8.9 | 16.8 | 6.0 | 22.6 | 5.7 |
| W. Germany | 6.7 | 1.3 | 35.8 | 0.9 | 7.3 | 14.4 | 5.0 | 22.3 | 6.1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

TABLE: Composantes de l'économie (%) pour certains pays.

Objectif

- On a n observations dans un espace de dimension p :

$$\mathbf{x}_i = [x_{i1} \quad x_{i2} \quad \cdots \quad x_{ip}]^T, \quad i = 1, \dots, n$$

et **aucune** variable réponse.

- Chacune des observations est caractérisée par un poids p_i , $i = 1, \dots, n$ tels que $\sum_{i=1}^n p_i = 1$.
- Généralement, on a $p_i = 1/n$, $i = 1, \dots, n$.
- On veut déterminer K groupes C_1, \dots, C_K .

Indice de similarité

Si on note N l'ensemble des observations $\{1, \dots, n\}$, alors l'indice de similarité est une application $s : N \times N \rightarrow \mathbb{R}^+$ telle que

$$\begin{aligned}s(i, j) &= s(j, i), & \forall (i, j) \in N \times N \\s(i, i) &= S > 0, & \forall i \in N \\s(i, j) &\leq S, & \forall (i, j) \in N \times N.\end{aligned}$$

On peut également définir une version normée de cet indice :

$$s^*(i, j) = \frac{1}{S} s(i, j), \quad \forall (i, j) \in N \times N.$$

Indice de dissimilarité

Si on note N l'ensemble des observations, alors l'indice de dissimilarité est une application $d : N \times N \rightarrow \mathbb{R}^+$ telle que

$$\begin{aligned} d(i, j) &= d(j, i), & \forall (i, j) \in N \times N \\ d(i, j) &= 0 \Leftrightarrow i = j. \end{aligned}$$

On peut également définir une version normée de cet indice :

$$d^*(i, j) = \frac{1}{D} d(i, j), \quad \forall (i, j) \in N \times N.$$

On a alors

$$d^*(i, j) = 1 - s^*(i, j) \text{ et } s^*(i, j) = 1 - d^*(i, j).$$

Distance

Une distance est un indice de dissimilarité qui vérifie, en plus, la propriété suivante (inégalité triangulaire) :

$$d(i, j) \leq d(i, k) + d(k, j), \quad \forall (i, j, k) \in N \times N \times N.$$

Pour $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, on peut définir plusieurs mesures de distance (présentées dans \mathbb{R}^2)

- la mesure **euclidienne** :

$$d^2(x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2;$$

- la mesure **maximale** :

$$d^2(x, y) = (\max(|x_1 - y_1|, |x_2 - y_2|))^2;$$

Distance

- la mesure **Manhattan** :

$$d^2(x, y) = (|x_1 - y_1| + |x_2 - y_2|)^2;$$

- la mesure **Minkowski d'ordre p** :

$$d^2(x, y) = ((x_1 - y_1)^p + (x_2 - y_2)^p)^{2/p};$$

- la mesure **normalisée** (ou mesure **ACP**) :

$Q = D_{s^2}^{-1} = \text{diag}(s_1^{-1}, \dots, s_p^{-1})$, où s_j^2 est la variance de la j^{e} composante de x ; et

- la mesure euclidienne généralisée avec mesure¹ $\mathbf{Q} > 0$ par

$$d^2(x, y) = (\mathbf{x} - \mathbf{y})^T \mathbf{Q} (\mathbf{x} - \mathbf{y}).$$

1. La matrice \mathbf{Q} doit être définie positive.

Distance

- On utilise généralement la distance **euclidienne** si toutes les variables sont mesurées sur la même échelle.
- Si toutes les variables ne sont pas mesurées sur la même échelle, on utilise plutôt la mesure normalisée.
- De façon équivalente, on peut également centrer et réduire les données avant d'utiliser la distance euclidienne.

Partition

- Une *partition* \mathcal{P}_K de N en K groupes est un ensemble $\{C_1, \dots, C_K\}$ de groupes non-vides, d'intersections nulles et dont l'union permet de retrouver N , c'est-à-dire
 - $C_k \neq \emptyset, \forall k \in \{1, \dots, K\};$
 - $C_k \cap C_j = \emptyset, \forall k \neq j;$ et
 - $C_1 \cup \dots \cup C_K = N.$

Exemple 1

On considère $N = \{1, 2, 3, 4, 5, 6, 7\}$ et

$$\mathcal{P}_3 = \{\{7\}, \{5, 4, 6\}, \{1, 2, 3\}\}.$$

Vérifier qu'il s'agit d'une partition.

Hiérarchie

- Une *hiérarchie* \mathcal{H} de N est un ensemble $\{C_1, \dots, C_K\}$ de groupes non-vides tels que
 - $N \in \mathcal{H}$;
 - $\forall i \in N, \{i\} \in \mathcal{H}$; et
 - $\forall A, B \in N, A \cap B \in \{A, B, \emptyset\}$.

Exemple 2

On considère $N = \{1, 2, 3, 4, 5, 6, 7\}$ et

$$\mathcal{H} = \{\{1\}, \dots, \{7\}, \{4, 5\}, \{2, 3\}, \{4, 5, 6\}, \{1, 2, 3\}, \{4, 5, 6, 7\}, N\}.$$

Vérifier qu'il s'agit d'une hiérarchie.

Inertie

- L'inertie permet de décrire à l'aide d'un seul nombre l'hétérogénéité d'un groupe de points.
- Soit $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_p] \in \mathbb{R}^p$ un nuage dont chacun des points est caractérisé par un poids p_i , $i = 1, \dots, n$ tels que $\sum_{i=1}^n p_i = 1$.
- Généralement, on a $p_i = 1/n$, $i = 1, \dots, n$.

Inertie

- Le centre de gravité d'un nuage de points est défini par

$$g = \sum_{i=1}^n p_i \mathbf{x}_i.$$

- L'inertie totale d'un nuage de point est donnée par

$$I_T = \sum_{i=1}^n p_i d^2(\mathbf{x}_i, g),$$

où $d()$ est la mesure de distance euclidienne.

Inertie

- Dans une procédure de classification, le nuage de points sera partitionné en K groupes C_1, \dots, C_K dont les poids respectifs sont donnés par P_j tel que

$$P_j = \sum_{i:\mathbf{x}_i \in C_j} p_i.$$

- Le centre de gravité de chacun des groupes est donné par

$$\mathbf{g}_j = \frac{1}{P_j} \sum_{i:\mathbf{x}_i \in C_j} p_i \mathbf{x}_i.$$

Inertie

- L'inertie entre les groupes (*between clusters inertia*) est donnée par

$$I_B = \sum_{j=1}^K P_j d^2(g_j, g).$$

- L'inertie interne des groupes (*within clusters inertia*) est donnée par

$$I_W = \sum_{j=1}^K P_j I_{C_j}, \quad I_{C_j} = \frac{1}{P_j} \sum_{i: \mathbf{x}_i \in C_j} p_i d^2(\mathbf{x}_i, g_j).$$

- Le théorème de König-Huygens indique que

$$I_T = I_B + I_W,$$

c'est-à-dire que l'inertie total d'un jeu de données est constante.

Inertie

- Une « bonne » classification aura ainsi une inertie inter-groupe élevée et une inertie intra-groupe faible.
- Minimiser l'intertie intra-groupe est équivalent à maximiser l'inertie inter-groupe (puisque l'inertie totale est constante).
- La proportion de l'inertie totale expliquée par la partition \mathcal{P}_K est

$$\left(1 - \frac{I_W}{I_T}\right) (100).$$

- Cette valeur augmente lorsque le nombre de groupe(s) K augmente : elle ne peut être utilisée que pour comparer des partitions avec le même nombre de groupes.

Classification

- On cherche à diviser la base de données de façon à ce que la variabilité intra-groupe soit la plus petite possible.
- Dans un algorithme de classification par ré-allocation dynamique, on mesure cette variabilité intra-groupe par

$$W(C_j) = \frac{1}{\text{card}(C_j)} \sum_{i, i' \in C_j} \|\mathbf{x}_i - \mathbf{x}_{i'}\|^2,$$

où $\text{card}(C_j)$ est le nombre d'éléments dans le groupe C_j et

$$\|\mathbf{x}_i - \mathbf{x}_{i'}\| = \sqrt{\sum_{k=1}^p (x_{ik} - x_{i'k})^2}.$$

Classification

- On doit donc résoudre le problème d'optimisation

$$\min_{C_1, \dots, C_K} \sum_{k=1}^K W(C_k).$$

- Potentiellement très long à résoudre : K^n possibilités de séparer une base de données contenant n observations en K groupes.
- On va plutôt utiliser un algorithme itératif qui convergera vers une solution qui sera un optimum local.

Classification par ré-allocation dynamique

Algorithmes :

1. On assigne au hasard un groupe (1 à K) à chacune des observations.
2. Pour chacun des groupes, on calcule le centroïde, c'est-à-dire le vecteur moyenne.
3. On réassigne chaque observation au groupe dont le centroïde est le plus près (en utilisant la distance euclidienne généralement).
4. On répète les étapes 2. et 3. jusqu'à stabilisation.

Classification par ré-allocation dynamique

- À chaque itération, la fonction que l'on cherche à minimiser

$$\sum_{k=1}^K W(C_k)$$

diminue.

- L'algorithme peut converger vers un minimum local plutôt que vers le minimum global.
- La solution dépend des groupes obtenus à l'étape 1. → il est important de répéter la procédure quelques fois en faisant varier les groupes initiaux.
- Il faut également utiliser l'algorithme avec différentes valeurs de K et utiliser son jugement pour déterminer la valeur la plus appropriée.

Exemple 3

- On considère une base de données contenant 4 observations dans \mathbb{R}^2

[,1] [,2]

| | | |
|------|---|----|
| [1,] | 5 | 4 |
| [2,] | 4 | 5 |
| [3,] | 1 | -2 |
| [4,] | 0 | -3 |

- Utiliser la classification par ré-allocation dynamique avec $K = 2$ et en utilisant les deux premières lignes de la base de données comme centroïdes de départ pour diviser la base de données.

Exemple 3 (suite)

```
kmeans(X, centers = X[1:2,])  
K-means clustering with 2 clusters of sizes 2, 2
```

Cluster means:

```
[,1] [,2]  
1 0.5 -2.5  
2 4.5  4.5
```

Clustering vector:

```
[1] 2 2 1 1
```

...

Exemple 3 (suite)

- Calculer la valeur minimale de la fonction à optimiser.
- Quelle proportion de la variabilité totale est expliquée par la classification obtenue ?

Algorithmes hiérarchiques ascendants

- La procédure hiérarchique ascendante implique de débuter l'analyse avec le plus grand nombre possible de groupes (*cluster*) (généralement en prenant les données individuellement).
- À chacune des étapes, on fusionne les deux groupes les plus semblables (les plus proches).
- On poursuite la procédure jusqu'à terminer l'analyse avec le moins de groupes possibles (généralement un seul groupe contenant toutes les données).
- On analyse l'arbre de classification (ou dendrogramme) ainsi obtenu et on choisit le nombre de classes K à conserver.

→ Demande de prendre une décision supplémentaire : comment définit-on la similitude entre deux groupes ?

Algorithmes hiérarchiques ascendants

La similitude entre deux groupes peut être déterminée de plusieurs façons.

- Lien simple (*simple linkage*) : la proximité entre deux groupes (A et B) est donnée par

$$\delta(A, B) = \min_{a_i \in A, b_j \in B} (d(a_i, b_j)),$$

c'est-à-dire qu'on utilise le minimum de la distance entre chacun des points du groupe A et chacun des points du groupe B .

Algorithmes hiérarchiques ascendants

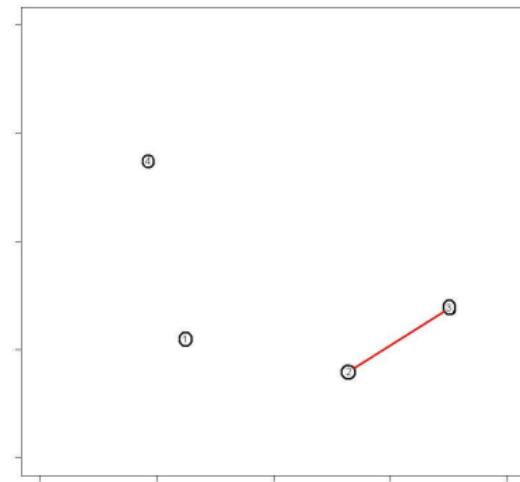
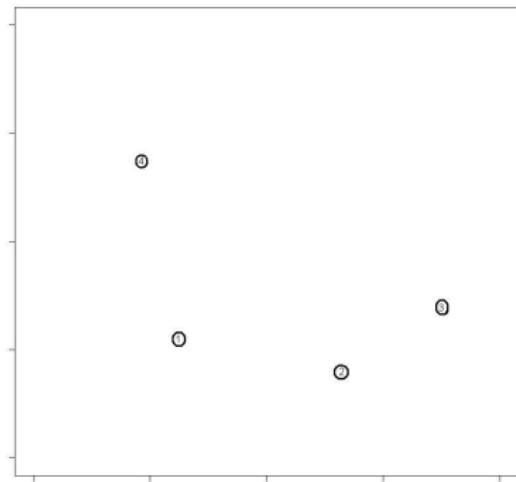


FIGURE: Lien simple.

Algorithmes hiérarchiques ascendants

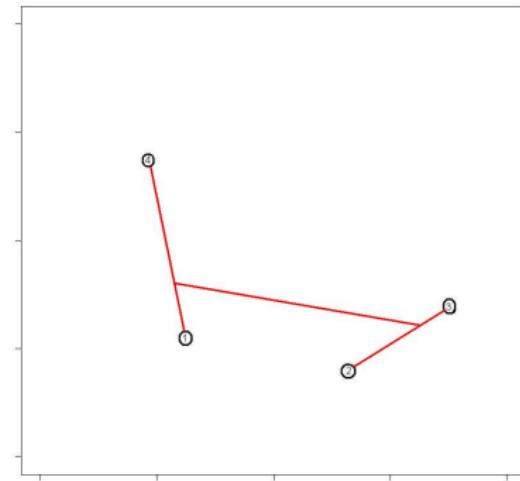
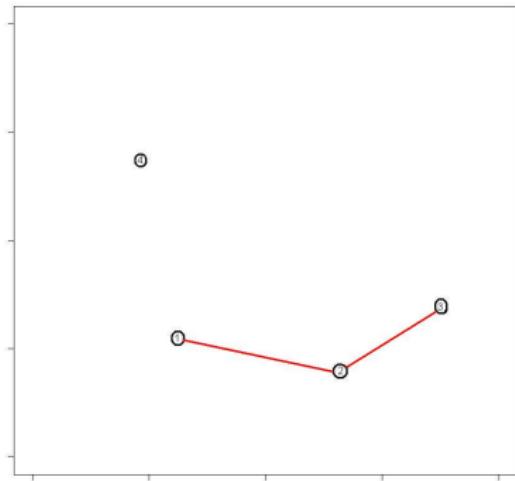


FIGURE: Lien simple.

Algorithmes hiérarchiques ascendants

- Lien complet (*complete linkage*) : la proximité entre deux groupes (A et B) est donnée par

$$\delta(A, B) = \max_{a_i \in A, b_j \in B} (d(a_i, b_j)),$$

c'est-à-dire qu'on utilise le maximum de la distance entre chacun des points du groupe A et chacun des points du groupe B .

Algorithmes hiérarchiques ascendants

- Lien moyen (*average linkage*) : la proximité entre deux groupes (A et B) est donnée par

$$\delta(A, B) = \frac{1}{\text{card}(A)\text{card}(B)} \sum_{a_i \in A} \sum_{b_j \in B} (d(a_i, b_j)).$$

Cette fois, on utilise la moyenne des distances entre chacun des points du groupe A et chacun des points du groupe B .

Algorithmes hiérarchiques ascendants

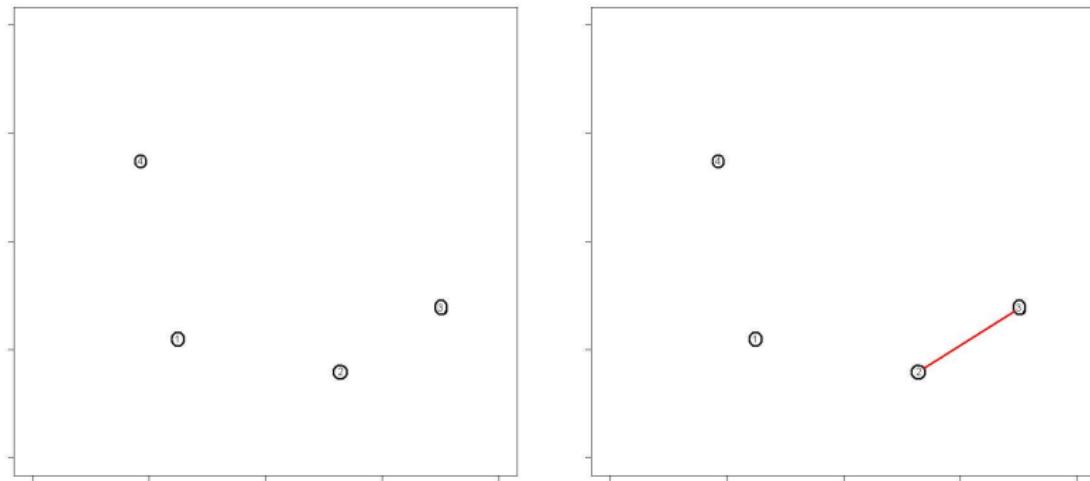


FIGURE: Lien moyen.

Algorithmes hiérarchiques ascendants

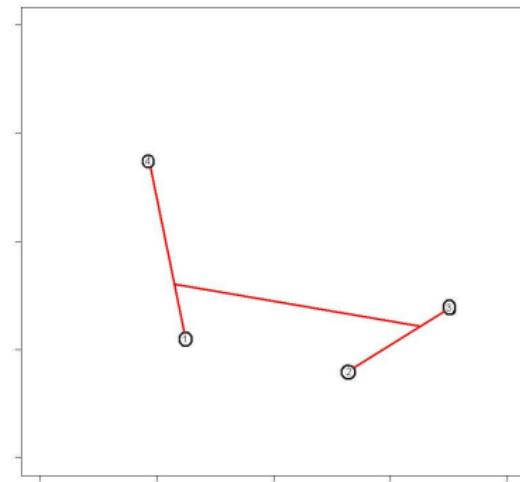
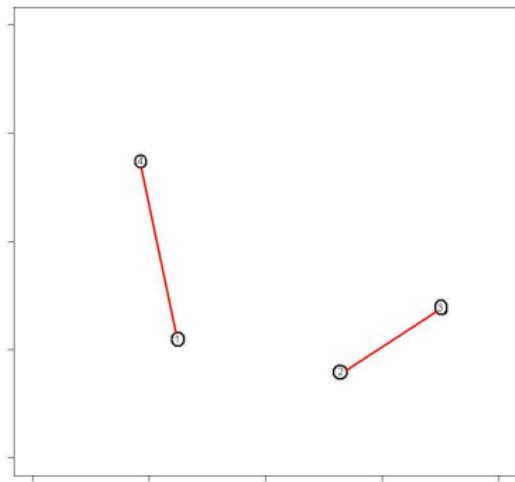


FIGURE: Lien moyen.

Algorithmes hiérarchiques ascendants

- Algorithme de Ward : on décompose l'inertie du nuage de points et minimise la perte d'information (perte d'inertie entre les groupes I_B) à chaque étape : on optimise donc le même critère que pour un algorithme de classification par ré-allocation dynamique.
- À chacune des étapes, la perte d'information due au regroupement des groupes A et B est donnée par

$$\delta(A, B) = \frac{P_A P_B}{P_A + P_B} d^2(g_A, g_B).$$

Également, on définit la perte d'information due au regroupement des groupes $C = A \cup B$ et D par

$$\delta(C, D) = \frac{(P_A + P_D)\delta(A, D) + (P_B + P_D)\delta(B, D) - P_D\delta(A, B)}{P_A + P_B + P_D}.$$

Algorithmes hiérarchiques ascendants

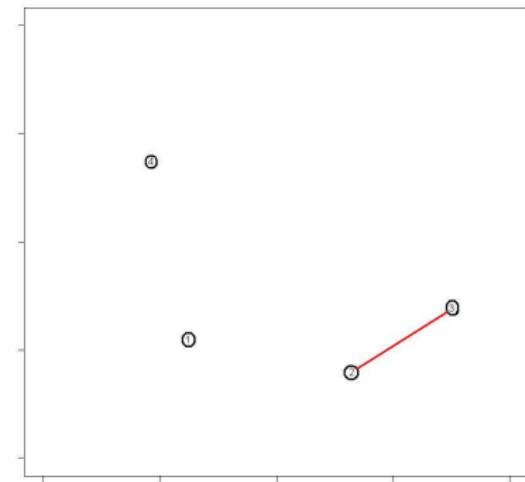
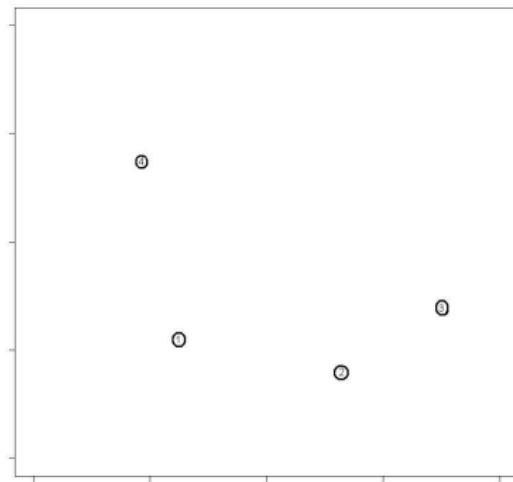


FIGURE: Algorithme de Ward. On obtient une perte d'information de 1.04 lors du premier lien.

Algorithmes hiérarchiques ascendants

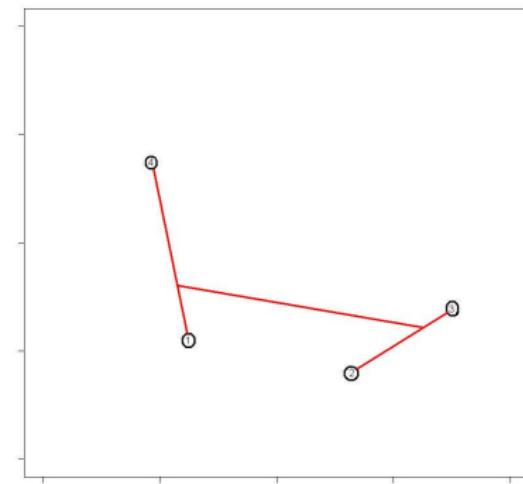
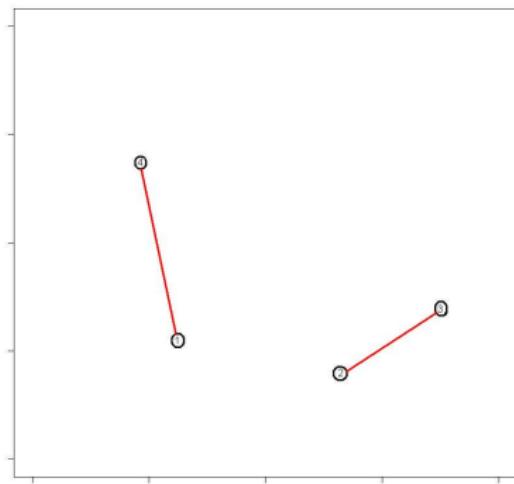


FIGURE: Algorithme de Ward. On obtient une perte d'information de 1.673 lors du second lien et de 3.248 lors du dernier.

Algorithmes hiérarchiques ascendants

Ainsi, avant de pouvoir utiliser un algorithme hiérarchique ascendant, il faut répondre aux deux questions suivantes :

- Quelle règle de ressemblance, de proximité, entre les observations choisir ? → **Choix de la mesure de distance**
- Comment définit-on la similitude entre deux groupes ? → **Choix du lien.**

Exemple 4

- On considère la base de données utilisée à l'Exemple 3

| | [,1] | [,2] |
|------|------|------|
| [1,] | 5 | 4 |
| [2,] | 4 | 5 |
| [3,] | 1 | -2 |
| [4,] | 0 | -3 |

- Utiliser un algorithme hiérarchique ascendant avec une fonction de lien complet et en utilisant la distance euclidienne pour diviser la base de données.

Exemple 4 (suite)

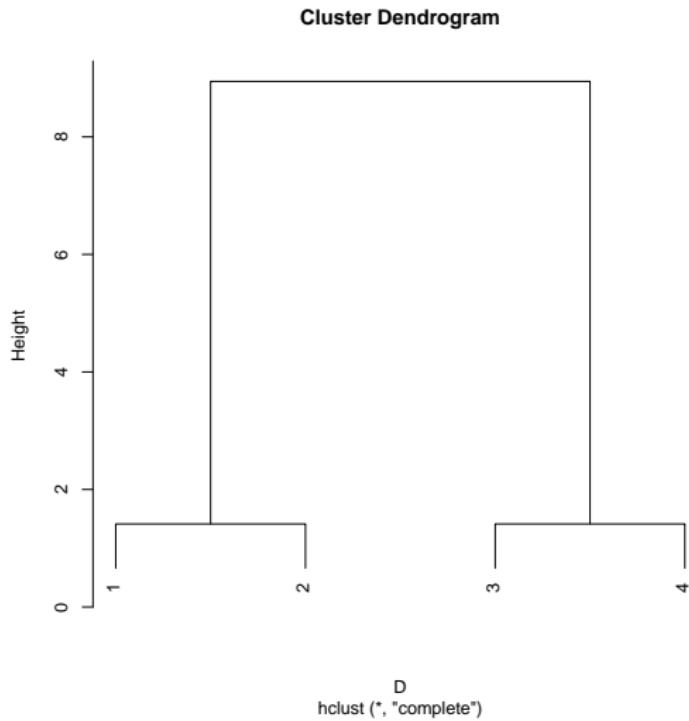
```
X <- matrix(c(5,4,4,5,1,-2,0,-3),4,2,byrow=TRUE)
D <- dist(X, method = "euclidean")
D
      1           2           3
2  1.414214
3  7.211103 7.615773
4  8.602325 8.944272 1.414214
```

Exemple 4 (suite)

```
modele <- hclust(D, method = "complete")
plot(modele)

cutree(modele, k = 2)
[1] 1 1 2 2
```

Exemple 4 (suite)



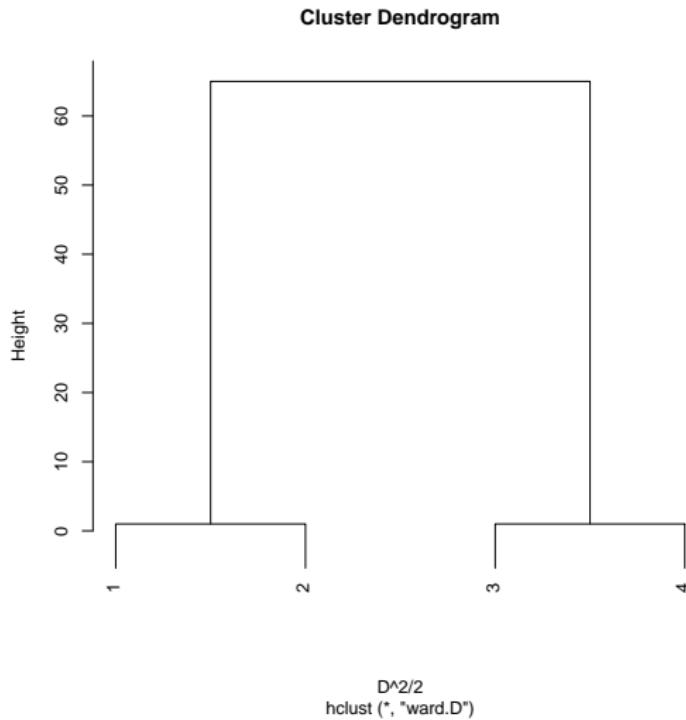
Exemple 4 (suite)

- Refaire l'exemple en utilisant la distance Manhattan.
- Refaire l'exemple en utilisant la distance euclidienne et l'algorithme de Ward (poids égaux à 1).

Exemple 4 (suite)

```
D <- dist(X, method = "euclidean")
modeleW <- hclust(D^2/2, method="ward.D")
plot(modeleW)
```

Exemple 4 (suite)



Application 1 : Eurojobs

```
set.seed(20)
clusters <- kmeans(data, 3)

# Ajouter les groupes à la base de données
data$groupe <- as.factor(clusters$cluster)

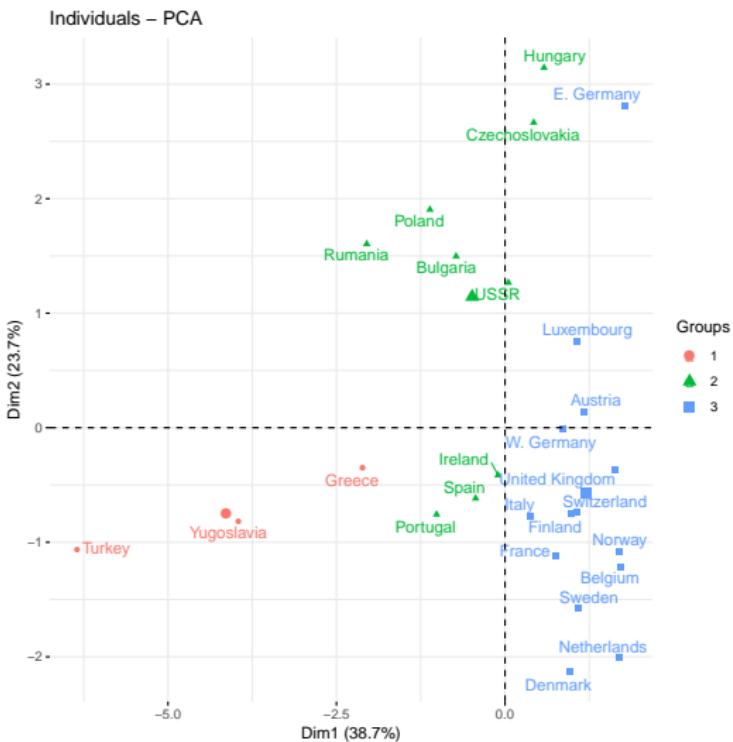
# Propriétés des groupes
str(clusters)
```

Application 1 : Eurojobs

```
# ACP (à faire avant d'ajouter les groupes à la base
res.pca <- PCA(data, scale.unit = TRUE, ncp = 5,
                 graph = FALSE)

fviz_pca_ind (res.pca, habillage = data$groupe,
               repel = TRUE)
```

Application 1 : Eurojobs



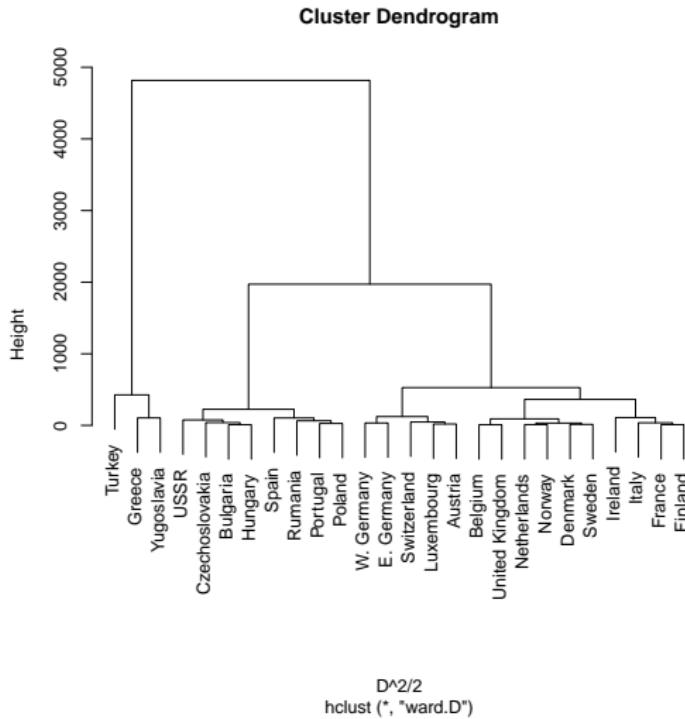
Application 1 : Eurojobs

```
# Matrice des distances
D <- dist(data[,1:9], method = "euclidean")

# Arbre avec l'algo. de Ward et des poids égaux à 1
tree <- hclust(D^2/2,method="ward.D")

# Ajout des groupes à la base initiale
data$groupe2 <- as.factor(cutree(tree, k = 3))
```

Application 1 : Eurojobs



Application 1 : Eurojobs

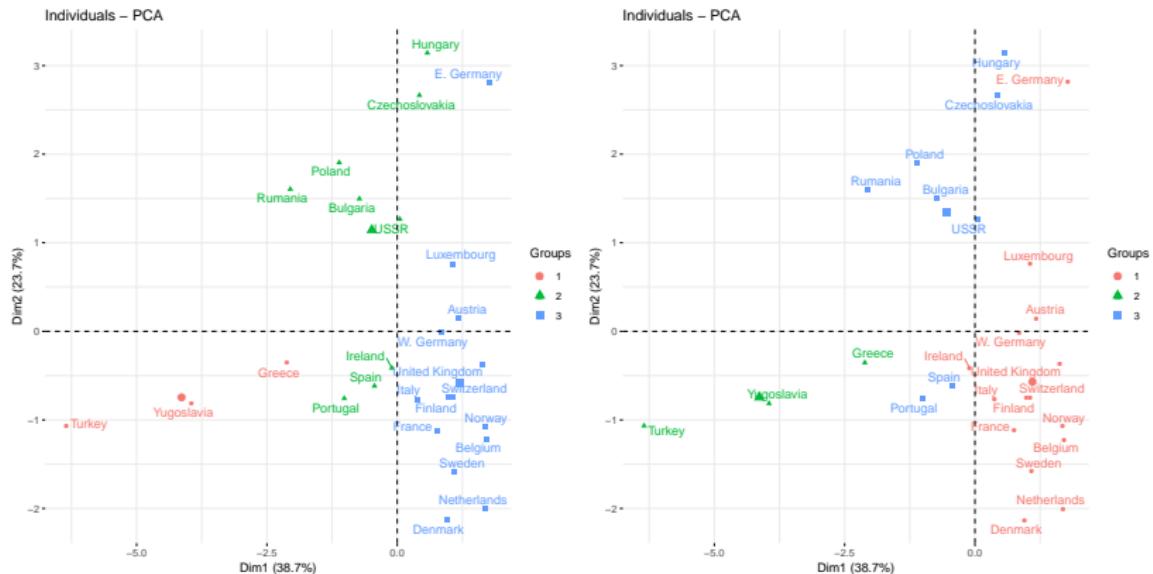


FIGURE: Classification par ré-allocation dynamique ($K = 3$) à gauche et Algorithme de Ward à droite.

Application 1 : Eurojobs

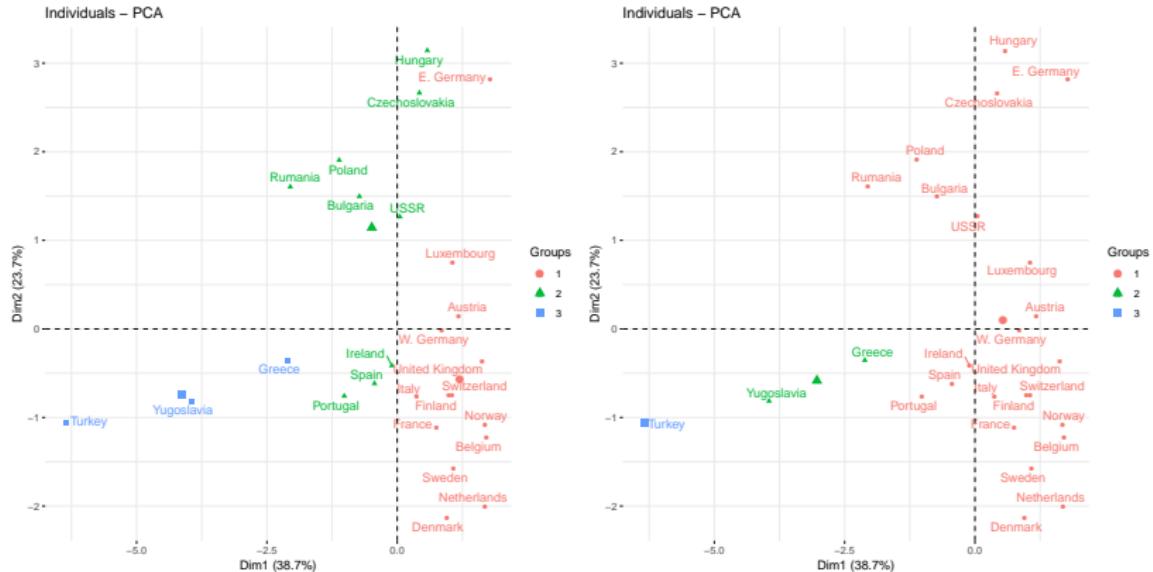


FIGURE: Utilisation du lien complet à gauche et du lien moyen à droite.

Application 2 : Uber

```
data14[1:3,]
```

| Date.Time | Lat | Lon | Base | Year | Month | ... |
|---------------------|---------|----------|--------|------|-------|-----|
| 2014-04-01 00:11:00 | 40.7690 | -73.9549 | B02512 | 2014 | 4 | ... |
| 2014-04-01 00:17:00 | 40.7267 | -74.0345 | B02512 | 2014 | 4 | ... |
| 2014-04-01 00:21:00 | 40.7316 | -73.9873 | B02512 | 2014 | 4 | ... |
| ... | | | | | | |

Application 2 : Uber

```
clusters <- kmeans(data14[,2:3], 5)

data14$groupe <- as.factor(clusters$cluster)

mescouleurs <- rainbow(length(levels(data14$groupe)))
plot(data14[,2], data14[,3], pch = 19, col =
      mescouleurs[data14$groupe])
```

Application 3 : Données télématiques

- Cet exemple est extrait de l'article *Covariate Selection from Telematics Car Driving Data*, écrit par M. Wüthrich et publié en 2017 dans le *European Actuarial Journal*, volume 7, No 1, pages 89 à 108.
- Les compagnies récoltent de plus en plus de données télématiques dont elles peuvent extraire des informations sur les habitudes de conduites des assurés. Cette information peut être utilisée pour compléter les informations habituellement utilisées en tarification : âge, type de voiture, distance parcourue, etc.
- Les données télématiques comprennent généralement de l'information sur : la vitesse, la localisation, les accélérations, les virages, etc.

Application 3 : Données télématiques

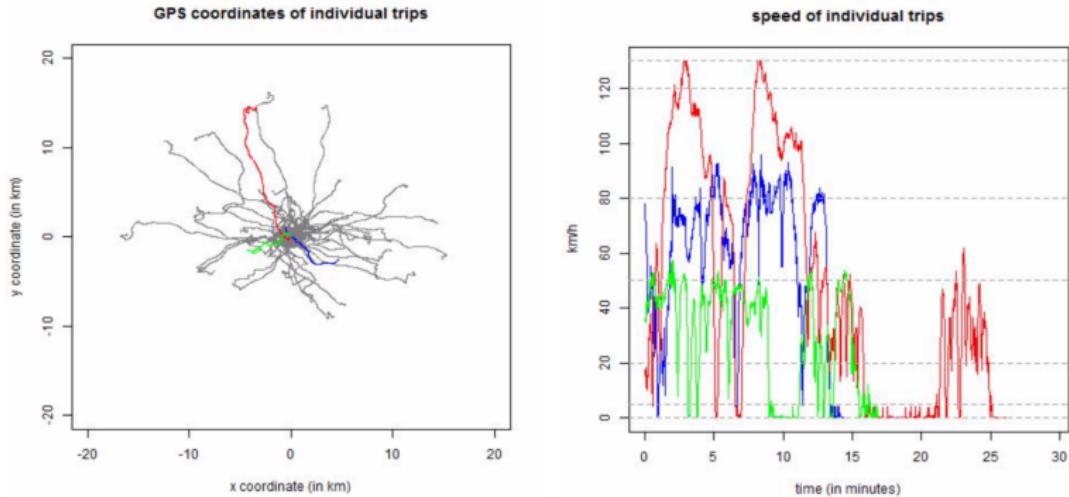
- L'étude porte sur 1 753 conducteurs pour lesquels on dispose de 200 trajets.
- Chaque trajet consiste en une série de coordonnées (x_t, y_t) , $t = 0, \dots, T$ dont on peut extraire la vitesse

$$v_t = \frac{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}{t - (t-1)}, \quad t = 1, \dots$$

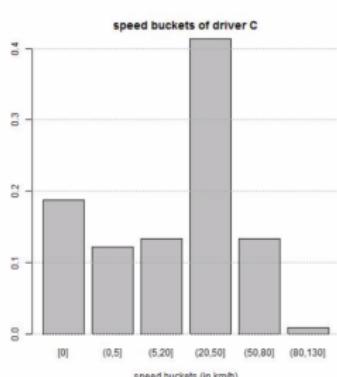
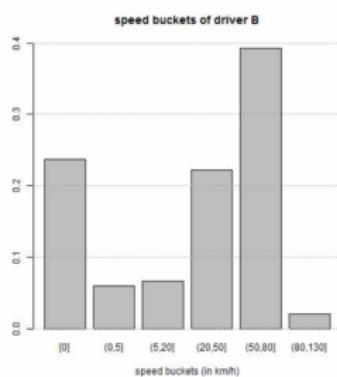
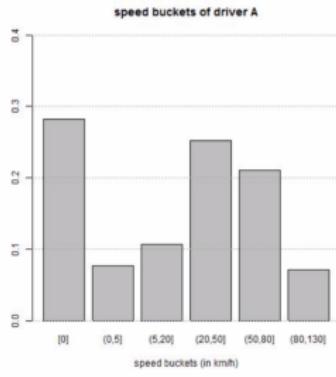
et une approximation de l'accélération donnée par

$$a_t = \frac{v_t - v_{t-1}}{t - (t-1)}, \quad t = 2, \dots$$

Application 3 : Données télématiques



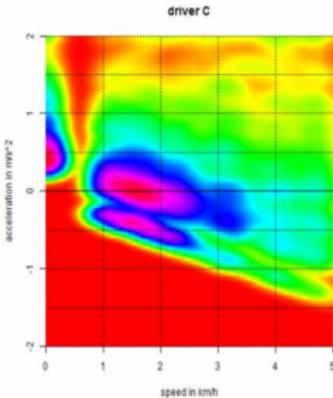
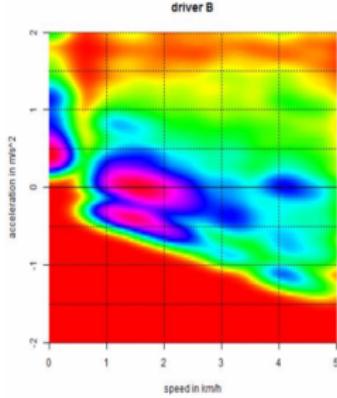
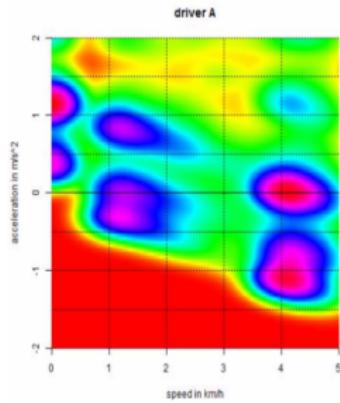
Application 3 : Données télématiques



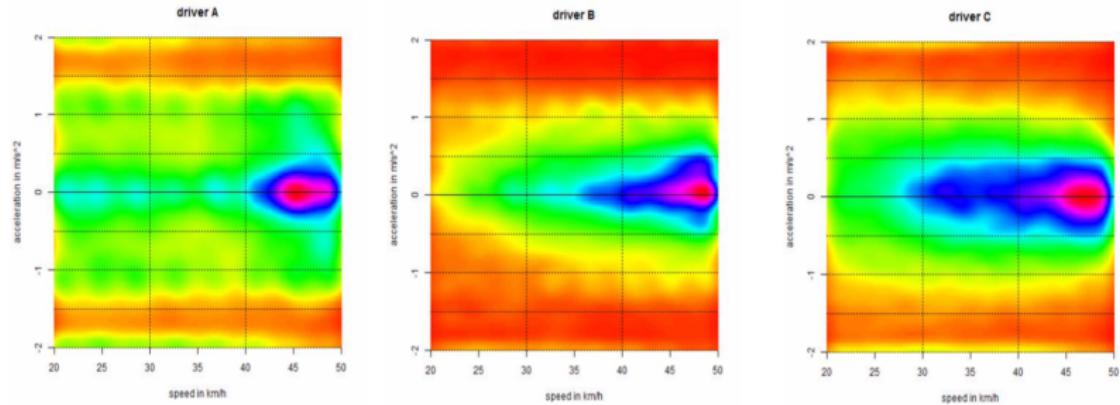
Application 3 : Données télématiques

- Pour chacun des intervalles de vitesse, on peut créer une carte vitesse-accélération moyenne où la couleur est fonction de la distribution empirique.
- On chercher à regrouper les conducteurs ayant des cartes similaires : analyse de classification.

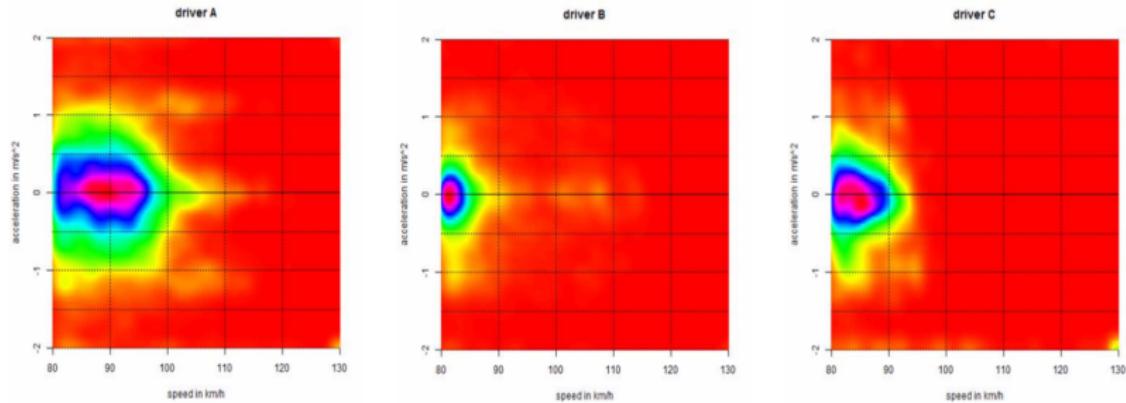
Application 3 : Données télématiques



Application 3 : Données télématiques



Application 3 : Données télématiques



Application 3 : Données télématisques

- Chaque graphique \mathcal{R} est divisé en J rectangles R_1, \dots, R_J tels que

$$\bigcup_{j=1}^J R_j = \mathcal{R} \text{ et } R_j \cap R_{j'} = \emptyset, \quad \forall j \neq j'.$$

- Si F_i est la distribution empirique sur le graphique \mathcal{R} d'un conducteur i , $i = 1, \dots, 1753$.
- On définit

$$x_j = \int_{R_j} dF(y) \geq 0, \quad j = 1, \dots, J$$

et telle que $\sum_{j=1}^J x_j = 1$. x_j mesure la masse de probabilité d'un conducteur sur le rectangle R_j .

Application 3 : Données télématiques

- Ainsi, un conducteur i est représenté par un vecteur de probabilités

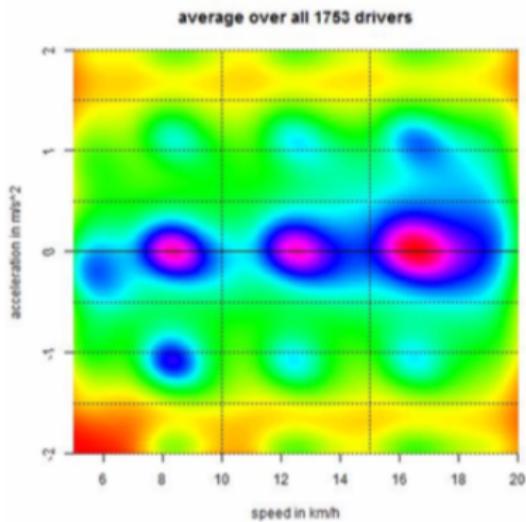
$$\mathbf{x}_i = [x_{i1} \quad \cdots \quad x_{iJ}] .$$

- La dissimilarité entre deux conducteurs est donnée par (on pourrait ajouter des poids w_j)

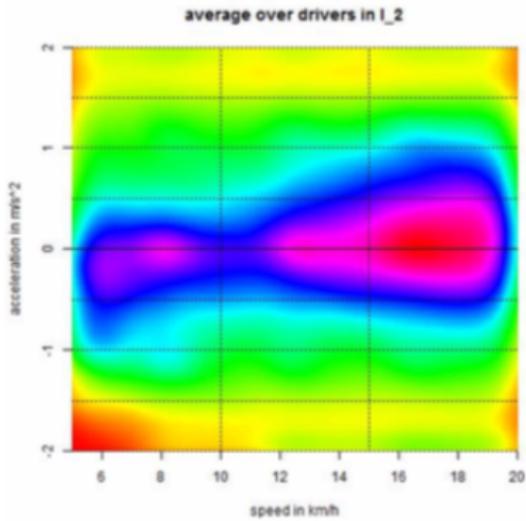
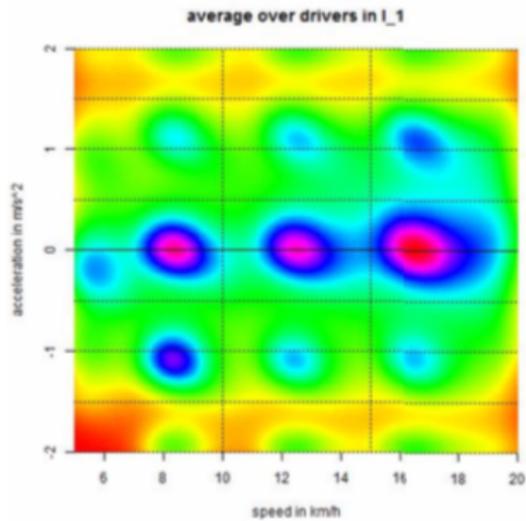
$$d(\mathbf{x}_i, \mathbf{x}_{i'}) = 0.5 \sum_{j=1}^J (x_{ij} - x_{i'j})^2 .$$

- On utilise un algorithme de classification par ré-allocation dynamique pour obtenir les groupes.

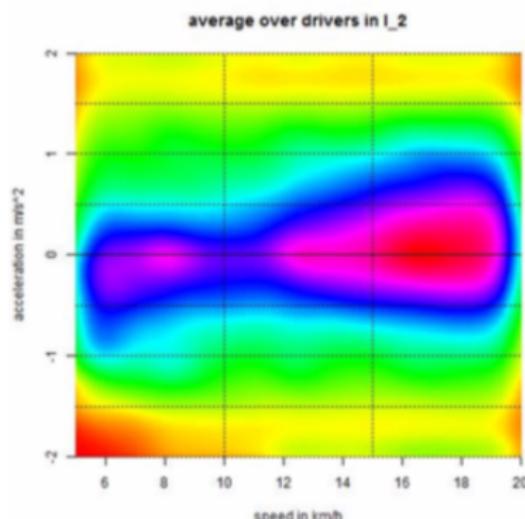
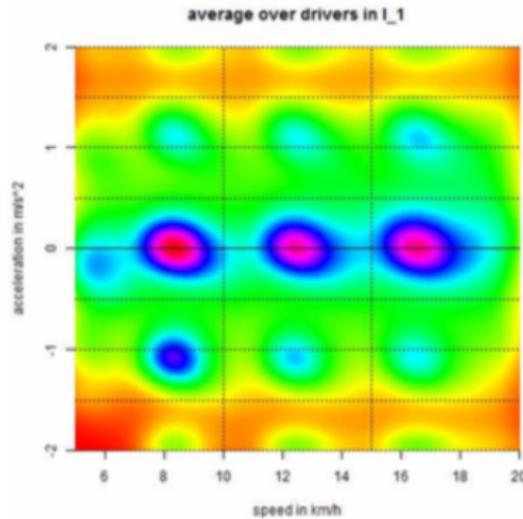
Application 3 : Données télématiques



Application 3 : Données télématiques



Application 3 : Données télématiques



Application 3 : Données télématiques

