AgroParisTech

Classification non supervisée

E. Lebarbier, T. Mary-Huard

Table des matières

1 Introduction **4**

2 Méthodes de partitionnement **5**
2.1 Mesures de similarité et de dissimilarité, distances 5
2.1.1 Définitions 5
2.1.2 Quelle distance employer ? 6
2.2 Formalisation du problème 7
2.3 Méthode des K-means 9
2.3.1 Principe 9
2.3.2 Exemple 9
2.3.3 Propriétés de l'algorithme 10
2.4 Classification ascendante hiérarchique 11
2.4.1 Principe 11
2.4.2 La distance de Ward 11
2.4.3 Exemple 12
2.4.4 Le dendrogramme 13
2.4.5 Généralisation 14
2.5 Quelques considérations supplémentaires 16

3 Méthode probabiliste : le modèle de mélange **18**
3.1 Présentation générale 18
3.1.1 Modèle 18
3.1.2 Affectation 20
3.1.3 Exemple d'un modèle de mélange de distributions gaussiennes 21
3.2 Estimation des paramètres du modèle 22
3.2.1 Algorithme EM 23
3.2.2 Calculs pour un modèle de mélange de distributions gaussiennes 25
3.2.3 Propriétés de l'algorithme EM 25
3.2.4 En pratique 26
3.2.5 Variantes de l'algorithme EM 26
3.2.6 Démonstration de la propriété 1 27
3.3 Choix du nombre de populations 28
3.3.1 Critères de sélection 28
3.3.2 En pratique 29
3.4 Logiciels 30

4 Exemples d'application **31**
4.1 Caractérisation de races anciennes de poules 31
4.1.1 Présentation du problème 31
4.1.2 Formalisation 32
4.1.3 Classification 33
4.1.4 Interprétation des résultats 34
4.2 Caractérisation de la phyllotaxie d'Arabidopsis 37
4.2.1 Présentation du problème 37
4.2.2 Classification par modèle de mélange et interprétation des résultats 39

5 Annexes **44**
5.1 Etude des deux étapes de la procédure K-means 44
5.2 Programmes pour l'analyse des données Poules 45
5.2.1 Programme SAS 45
5.2.2 Programme R 47
5.3 Programme R pour l'analyse des données de Phyllotaxie 48
5.3.1 Programme 48
5.3.2 Détails sur la fonction Mclust 49
5.3.3 Installation du package *mclust* 50

Chapitre 1

Introduction

La classification non supervisée désigne un corpus de méthodes ayant pour objectif de dresser ou de retrouver une typologie existante caractérisant un ensemble de n observations, à partir de p caractéristiques mesurées sur chacune des observations. Par typologie, on entend que les observations, bien que collectées lors d'une même expérience, ne sont pas toutes issues de la même population homogène, mais plutôt de K populations. Deux exemples peuvent être considérés :

- l'ensemble des clients d'une banque est une collection de n observations, chacune de ces observations étant caractérisée par la nature des p transactions bancaires qu'elle réalise. Il existe certainement différents K "profils types" de clients. L'objectif est alors d'une part de retrouver ces profils types à partir de l'information sur les transactions bancaires, et d'autre part de déterminer, pour chaque observation, à quel profil type elle correspond.
- une cohorte de patients représente un ensemble de n observations, chacune décrite par p mesures physiologiques. Bien qu'ayant tous la même pathologie, ces patients n'ont pas tous le même historique médical. On souhaite donc dans un premier temps établir une typologie de ces patients en K groupes selon leurs caractéristiques physiologiques. Dans un deuxième temps, on étudiera si la réponse au traitement diffère pour des patients issus de groupes différents.

Comme le montrent ces deux exemples, la classification peut être un objectif en soit (exemple 1), ou ne représenter qu'une étape de l'analyse statistique (exemple 2).

En classification non supervisée, l'appartenance des observations à l'une des K populations n'est pas connue. C'est justement cette appartenance qu'il s'agit de retrouver à partir des p descripteurs disponibles. En classification supervisée au contraire, l'appartenance des n observations aux différentes populations est connue, et l'objectif est de construire une règle de classement pour prédire la population d'appartenance de nouvelles observations. On distinguera donc bien les deux problématiques. Dans ce polycopié, seule la classification non supervisée est abordée.

Il existe une très large famille de méthodes dédiées à la classification non supervisée. Dans ce polycopié, nous n'en présentons que 3. Les deux premières, la classification ascendante hiérarchique et les K-means, font partie des méthodes dites de partitionnement et seront présentées au chapitre 2. La troisième, appelée modèle de mélanges, se place dans un cadre probabiliste et fait l'objet du chapitre 3. Ces différentes méthodes sont illustrées au chapitre 4.

Chapitre 2

Méthodes de partitionnement

Les méthodes de partitionnement font partie des méthodes dites exploratoires, qui ne font appel à aucune modélisation statistique. Nous présentons ici le problème de la classification non supervisée du point de vue exploratoire.

En l'absence de toute hypothèse concernant la distribution des données, le regroupement des observations en classes se fait sur des considérations géométriques : on regroupe des observations qui sont "proches" les unes des autres dans l'espace de représentation (l'espace des variables). Cette première intuition montre la nécessité de définir une mesure de la proximité entre observations. C'est pourquoi nous commençons par introduire les notions de similarité, de dissimilarité et de distance entre points au paragraphe 2.1. Au paragraphe 2.2, nous montrons comment l'objectif de classification, faire des groupes homogènes et distincts, peut se formaliser du point de vue mathématique en employant les notions d'inerties intra-classe et inter-classe. Trouver les groupes se traduit alors comme un simple problème de minimisation. Enfin, aux paragraphes 2.3 et 2.4, nous présentons deux algorithmes couramment utilisés pour la résolution de ce problème de minimisation : l'algorithme des K-means et l'algorithme de classification ascendante hiérarchique.

2.1 Mesures de similarité et de dissimilarité, distances

2.1.1 Définitions

Afin de définir l'homogénéité d'un groupe d'observations, il est nécessaire de mesurer une ressemblance entre deux observations. On introduit ainsi les notions de dissimilarité et de similarité :

Définition 1 *Une dissimilarité est une fonction d qui à tout couple (x_1, x_2) associe une valeur dans $\mathbb{R}_+$, et telle que*
$\star$ $d(x_1, x_2) = d(x_2, x_1) \geq 0$,
$\star$ $d(x_1, x_2) = 0 \Rightarrow x_1 = x_2$.

Autrement dit, moins les unités x_1 et x_2 se ressemblent, plus le score est élevé. Remarquons qu'une distance est une dissimilarité, puisque toute distance possède les deux propriétés précédentes ainsi que l'inégalité triangulaire. Toutes les distances connues, en particulier la distance euclidienne, sont donc des exemples de dissimilarité.

À l'inverse, une autre possibilité consiste à mesurer la ressemblance entre observations à l'aide d'une similarité :

Définition 2 *Une similarité est une fonction s qui à tout couple (x_1, x_2) associe une valeur dans $\mathbb{R}^+$, et telle que*
$\star$ $s(x_1, x_2) = s(x_2, x_1) \geq 0$,
$\star$ $s(x_1, x_1) \geq s(x_1, x_2)$.

Contrairement à la dissimilarité, plus les unités x_1 et x_2 se ressemblent plus le score est élevé. On peut citer comme exemple de similarité la valeur absolue du coefficient de corrélation :

$$|\rho(x_1, x_2)| = \left| \frac{\sum_{j=1}^{p}(x_{1j} - x_{1\bullet})(x_{2j} - x_{2\bullet})}{\sqrt{\sum_{j=1}^{p}(x_{1j} - x_{1\bullet})^2 \sum_{j=1}^{p}(x_{2j} - x_{2\bullet})^2}} \right|$$

2.1.2 Quelle distance employer ?

Le choix de la distance est une question primordiale pour les méthodes exploratoires multivariées. En effet, c'est à cette étape qu'il est possible pour l'expérimentateur d'utiliser au mieux l'information *a priori* dont il dispose, afin de proposer une mesure pertinente de ressemblance entre observations. Pourtant, cette étape est bien souvent négligée. Commençons par illustrer l'importance de ce choix par un exemple très simple, tiré de [5].

L'étude comporte trois exploitations agricoles (les individus), pour lesquelles on dispose de la surface totale de l'exploitation, ainsi que de l'âge et du revenu de l'exploitant (les variables). Ces données sont résumées dans le tableau suivant :

	Age	Revenu	Surface
1	30	290 000	20
2	50	300 000	30
3	52	320 000	28

Pour déterminer quels sont les individus qui se ressemblent le plus, on utilise dans un premier temps la distance euclidienne canonique sur l'ensemble des 3 variables :

$$d^2(x_1, x_2) = \sum_{j=1}^{3}(x_{1j} - x_{2j})^2 \quad .$$

On obtient alors les distances entre individus données dans la table 2.1 (gauche).

On remarque que les deux exploitations les plus proches sont les exploitations 1 et 2. Cette forte ressemblance tient au fait que les différentes variables ont des unités de mesure différentes, et donc des poids différents dans le calcul des distances. En particulier, la variable revenu contribue plus fortement au calcul des distances que les autres. Afin d'éviter qu'une variable ne prenne trop d'importance du simple fait de son unité de mesure, il est donc recommandé de normaliser les données en centrant et en réduisant l'ensemble des variables. Les distances calculées sur les données centrées et réduites sont données dans la table 2.1 (droite). Cette fois, les exploitations 2 et 3 sont plus proches.

La normalisation des données donne un poids égal à toutes les variables dans le calcul des distances. Si cette égalité de traitement des variables évite tout effet d'échelle, elle a aussi pour

	1	2	3
1	0		
2	10^4	0	
3	3.10^4	2.10^4	0

	1	2	3
1	0		
2	22.4	0	
3	23.6	3.46	0

TABLE 2.1 – Deux calculs de distances entre exploitations, l'un sur les données brutes (gauche), l'autre sur les données normalisées (droite)

conséquence de ne pas distinguer les variables portant une information importante pour l'objectif de classification des variables non pertinentes pour cet objectif. Une fois les variables normalisées, il semble donc préférable d'utiliser une distance euclidienne pondérée

$$d^2(x_1, x_2) = \sum_{j=1}^{3} w_j (x_{1j} - x_{2j})^2 \quad ,$$

où chaque pondération w_j est fonction de l'information portée par la variable X^j. À l'extrême, si une variable X^j n'apporte aucune information pertinente, le poids w_j qui lui est associé peut être fixé à 0.

La distinction entre variables pertinentes et non pertinentes découle directement du problème du sujet de l'étude et des connaissances de l'expérimentateur. Dans l'exemple des exploitations, si l'objectif est de dresser une typologie des exploitations agricoles en fonction du profil socio-professionnel de leur exploitant, les variables pertinentes sont l'âge et le revenu de l'exploitant. Prendre en compte la surface de l'exploitation n'apporte aucune information, on attribuera donc à cette variable un poids nul (i.e. la variable sera éliminée).

Cet exemple montre d'une part la nécessité de choisir soigneusement les variables sur lesquelles baser le calcul des distances, et illustre d'autre part le fait que normaliser les données ne constitue pas une solution "miracle", garante de la validité de la classification. En résumé, le choix d'une distance nécessite un travail en trois temps :

- Normalisation : cette étape permet d'éviter tout effet d'échelle : les variables ont ainsi une contribution équivalente au calcul des distances, quelle que soit leur unité de mesure initiale.
- Sélection des variables : elle doit être basée sur l'analyse descriptive des données ainsi que sur les connaissances *a priori* dont on dispose sur le problème.
- Choix des poids : on fixe avec l'expérimentateur le poids à accorder à chacune des variables, si l'on souhaite que certaines variables soient plus influentes dans le calcul des distances.

Dans tous les cas, le choix de la distance doit être issu de la réflexion de l'expérimentateur et du statisticien, et non du choix par défaut du logiciel utilisé pour l'analyse.

2.2 Formalisation du problème

L'objectif de la classification non supervisée étant de déterminer des groupes - on désignera ces groupes dans la suite par "classes" - *homogènes* et *distincts*, il est nécessaire de formaliser ces deux notions du point de vue géométrique. Pour cela, nous repartons de la définition de l'inertie

d'un nuage de points. On dispose de n points $x_1, ..., x_n$, et on désigne par x_G le barycentre du nuage de ces points :

$$x_G = \frac{1}{n} \sum_{i=1}^{n} x_i \quad .$$

L'inertie totale est définie de la manière suivante :

$$I_T = \sum_{i=1}^{n} d^2(x_i, x_G) = \sum_{i=1}^{n} ||x_i - x_G||^2 \quad ,$$

où la distance choisie ici est la distance euclidienne. On suppose qu'en réalité le nuage de points est composé de K classes de points distincts C_1,...,C_K, chacune de ces classes ayant pour barycentre x_{C_k}. On peut alors décomposer l'inertie totale du nuage de la manière suivante :

$$\begin{aligned}
I_T &= \sum_{i=1}^{n} ||x_i - x_G||^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - x_G||^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - x_{C_k} + x_{C_k} - x_G||^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k} \left(||x_i - x_{C_k}||^2 + ||x_{C_k} - x_G||^2 \right) \quad \text{(théorème de Huygens)} \\
&= \underbrace{\sum_{k=1}^{K} \sum_{i \in C_k} d^2(x_i, x_{C_k})}_{1} + \underbrace{\sum_{k=1}^{K} n_k d^2(x_{C_k}, x_G)}_{2} \quad ,
\end{aligned}$$

où n_k est le nombre d'observations de la classe C_k. Le premier terme mesure la somme des distances entre les points d'une classe et leur barycentre. On appelle cette inertie l'inertie intra-classe, et on la note I_W (pour Within). Si les classes sont homogènes, les points d'une même classe sont proches les uns des autres et l'inertie intra-classe est faible. La deuxième somme mesure à quel point les barycentres des classes sont loin du barycentre global, c'est-à-dire à quel point les classes sont distantes les unes des autres. Cette somme est appelée l'inertie inter-classe, et notée I_B (pour Between).

Ainsi, s'il existe K classes bien identifiées, il est théoriquement possible de les retrouver en essayant tous les regroupements en K classes possibles et en choisissant celui qui minimise l'inertie intra-classe (ou qui maximise l'inertie inter-classe, ce qui revient au même puisque $I_W + I_B = I_T$ et que I_T ne dépend pas des classes). Du point de vue formel, la partition optimale $\mathcal{C}_K^*$ des observations en K classes est donc définie de la manière suivante :

$$\mathcal{C}_K^* = \underset{\mathcal{C} \in \mathcal{C}_\mathcal{K}}{\text{Argmin}} \sum_{k=1}^{K} \sum_{i \in C_k} d^2(x_i, x_{C_k}) \quad , \tag{2.1}$$

où $\mathcal{C}_\mathcal{K}$ est l'ensemble des partitions possibles des n observations en K classes. Pour répondre à notre objectif de classification, il ne reste plus qu'à identifier la partition optimale. Bien que

cet objectif soit simple, sa résolution pratique est impossible, car la complexité combinatoire de ce problème est bien trop élevée. En effet, le nombre de partitions de n observations en K classes devient rapidement très grand : si $n = 19$ et $K = 4$, il existe plus de 10^{10} possibilités. Les algorithmes de classification sont donc des algorithmes qui réduisent les temps de calcul en ne visitant qu'un nombre restreint de partitions.

Puisqu'on ne visite pas l'ensemble de toutes les partitions possibles, on ne peut pas garantir que l'on trouvera la partition optimale en utilisant ces algorithmes. Toutefois, le choix des partitions visitées est raisonné, on espère donc trouver la partition optimale, ou du moins une partition suffisamment proche de la partition optimale. Ces algorithmes proposant une solution approchée au problème sont dits heuristiques. Nous en présentons deux : l'algorithme des K-means et la classification hiérarchique ascendante (CAH).

Enfin, remarquons que le raisonnement précédent suppose le nombre de classes K connu *a priori*. En effet, la mesure de l'inertie intra-classe permet de comparer des partitions de même taille entre elles (et donc de choisir la meilleure), mais ne permet pas de comparer des partitions de taille différente. Pour s'en convaincre, il suffit d'observer que la partition d'un ensemble de n éléments en n classes a une inertie intra-classe nulle, elle est donc meilleure que toute autre partition de taille plus petite ! La détermination du nombre de classes à partir des données est un problème difficile qui sera succinctement abordé au paragraphe 2.5.

2.3 Méthode des K-means

Cet algorithme fut longtemps utilisé sur les grands jeux de données en raison de sa rapidité. On s'intéresse tout d'abord à l'algorithme même, puis à ses propriétés.

2.3.1 Principe

On suppose qu'il existe K classes distinctes. On commence par désigner K centres de classes $\mu_1, ..., \mu_K$ parmi les individus. Ces centres peuvent être soit choisis par l'utilisateur pour leur "représentativité", soit désignés aléatoirement. On réalise ensuite itérativement les deux étapes suivantes :

- Pour chaque individu qui n'est pas un centre de classe, on regarde quel est le centre de classe le plus proche. On définit ainsi K classes $C_1, ..., C_K$, où

$$C_i = \{\text{ensemble des points les plus proches du centre } \mu_i\} \quad .$$

- Dans chaque nouvelle classe C_i, on définit le nouveau centre de classe μ_i comme étant le barycentre des points de C_i.

L'algorithme s'arrête suivant un critère d'arrêt fixé par l'utilisateur qui peut être choisi parmi les suivants : soit le nombre limite d'itérations est atteint, soit l'algorithme a convergé, c'est-à-dire qu'entre deux itérations les classes formées restent les mêmes, soit l'algorithme a "presque" convergé, c'est-à-dire que l'inertie intra-classe ne s'améliore quasiment plus entre deux itérations.

2.3.2 Exemple

La figure 2.1 illustre l'algorithme sur un exemple où quatre points a (-1,1), b (0,1), c (3,0) et d (3,-1) doivent être classés en 2 classes. On remarque sur cet exemple que bien qu'à l'initialisation les centres de classes sont mal répartis, l'algorithme a convergé en retrouvant les "vraies" classes.

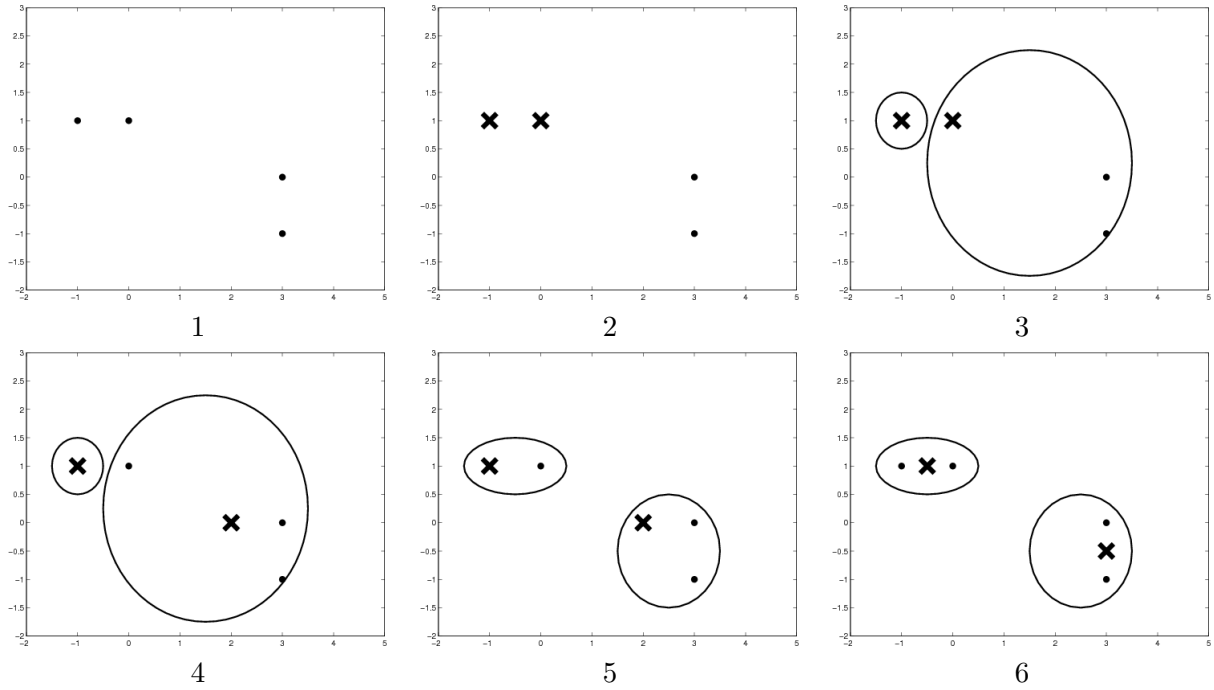


FIGURE 2.1 – Une illustration de l'algorithme K-means. (1) On dispose de 4 points à classer en deux classes. (2) À l'initialisation, deux de ces points sont choisis comme centres de classe. (3) Deux classes sont créées en regroupant les autres points en fonction du centre de classe le plus proche. (4) On définit les nouveaux centres de classe comme étant le barycentre des classes nouvellement créées. (5) On regroupe à nouveau les points. (6) On définit les nouveaux centres de classes. A l'étape suivante rien ne change, l'algorithme a convergé.

2.3.3 Propriétés de l'algorithme

Comment justifier l'algorithme des K-means au vu de notre objectif de minimisation de l'inertie intra-classe ? On montre (voir Annexes) qu'à chacune des deux sous-étapes de l'algorithme I_W diminue (on peut le constater visuellement sur l'exemple précédent). On est donc certain à chaque étape d'améliorer la classification, au sens du critère I_W. La marge d'amélioration étant finie (on ne peut pas obtenir une inertie intra-classe plus faible que celle de la partition optimale), l'algorithme converge forcément. En pratique, on constate que dans la majorité des cas très peu d'itérations sont nécessaires. Toutefois, il est important de se rappeler que le minimum atteint dépend de l'initialisation de l'algorithme : suivant les points initialement choisis comme centres de classe, on obtient des partitions qui peuvent être très différentes. Cette instabilité dans la composition des classes en fonction de l'initialisation est l'inconvénient majeur de la méthode.

L'algorithme ne permet donc pas de trouver la partition optimale, mais converge plutôt vers une partition localement optimale (minima locaux). Pour résoudre en partie ce problème, on peut choisir de faire tourner l'algorithme plusieurs fois avec différentes initialisations, et de choisir la meilleure des partitions obtenues au sens de l'inertie intra-classe.

2.4 Classification ascendante hiérarchique

La classification ascendante hiérarchique, notée dans la suite CAH, a pour objectif de construire une suite de partitions emboîtées des données en n classes, $n-1$ classes, ... , 1 classe. Ces méthodes peuvent être vues comme la traduction algorithmique de l'adage "qui se ressemble s'assemble".

2.4.1 Principe

- À l'étape initiale, les n individus constituent des classes à eux seuls.
- On calcule les distances deux à deux entre individus, et les deux individus les plus proches sont réunis en une classe.
- La distance entre cette nouvelle classe et les $n-2$ individus restants est ensuite calculée, et à nouveau les deux éléments (classes ou individus) les plus proches sont réunis.

Ce processus est réitéré jusqu'à ce qu'il ne reste plus qu'une unique classe constituée de tous les individus. On constate que la nouveauté ici vient de la nécessité de définir deux distances : la distance usuelle entre deux individus, et une distance entre classes. Le choix d'une distance entre individus ayant déjà été discuté, il reste à choisir une distance entre classes, en gardant à l'esprit que l'objectif est de trouver la partition en K classes des observations dont l'inertie intra-classe est minimale.

2.4.2 La distance de Ward

Considérons l'évolution de l'inertie intra-classe au fur et à mesure de la classification. À l'initialisation, toutes les classes sont composées d'une unique observation. Chaque classe est donc parfaitement homogène, et l'inertie intra-classe est nulle. À la dernière étape de l'algorithme, toutes les observations sont regroupées pour former une unique classe. L'inertie inter-classe est alors nulle, et l'inertie intra-classe est maximum (puisque $I_W + I_B = I_T$). Il n'est par ailleurs pas difficile de constater qu'à chaque étape de la classification, l'inertie intra-classe augmente alors que l'inertie inter-classe diminue, car chaque étape fusion fait perdre de l'homogénéité aux deux classes fusionnées. L'objectif étant d'aboutir à une partition en K classes d'inertie intra-classe minimum, la stratégie va consister à regrouper à chaque étape les deux classes dont la fusion entraîne le plus faible gain d'inertie intra-classe (ou de manière équivalente la plus faible perte d'inertie inter-classe).

Calculons maintenant la perte d'inertie inter-classe lorsque l'on passe de k à $k-1$ classes en fusionnant les deux classes C_j et C_ℓ. Lors de cette étape, les $k-2$ classes autres que C_j et C_ℓ ne sont pas modifiées, la différence entre les inerties inter-classe avant et après fusion s'écrit donc :

$$n_j d^2(x_{C_j}, x_G) + n_\ell d^2(x_{C_\ell}, x_G) - (n_j + n_\ell) d^2(x_{C_{j\ell}}, x_G) \quad ,$$

où $x_{C_{j\ell}}$ est le barycentre de la classe $C_{j\ell}$ issue de la fusion, et n_j et n_ℓ sont les effectifs respectifs des classes C_j et C_ℓ. Ce barycentre s'exprime en fonction des barycentres initiaux de la manière suivante

$$x_{C_{j\ell}} = \frac{n_j x_{C_j} + n_\ell x_{C_\ell}}{n_j + n_\ell} \quad .$$

Ainsi, la perte d'inertie vaut

$$\begin{aligned}
I_B^{k+1} - I_B^k &= n_j \left\|x_{C_j} - x_G\right\|^2 + n_\ell \left\|x_{C_\ell} - x_G\right\|^2 - (n_j + n_\ell) \left\|x_{C_{j\ell}} - x_G\right\|^2 \\
&= n_j \left\|x_{C_j} - x_G\right\|^2 + n_\ell \left\|x_{C_\ell} - x_G\right\|^2 - (n_j + n_\ell) \left\|\frac{n_j x_{C_j} + n_\ell x_{C_\ell}}{n_j + n_\ell} - x_G\right\|^2 \\
&= n_j \left\|x_{C_j} - x_G\right\|^2 + n_\ell \left\|x_{C_\ell} - x_G\right\|^2 - \frac{1}{n_j + n_\ell} \left\|n_j(x_{C_j} - x_G) + n_\ell(x_{C_\ell} - x_G)\right\|^2 \\
&= n_j \left\|x_{C_j} - x_G\right\|^2 + n_\ell \left\|x_{C_\ell} - x_G\right\|^2 - \frac{n_j^2}{n_j + n_\ell} \left\|x_{C_j} - x_G\right\|^2 - \frac{n_\ell^2}{n_j + n_\ell} \left\|x_{C_\ell} - x_G\right\|^2 \\
&\qquad - \frac{2n_j n_\ell}{n_j + n_\ell} \left\langle x_{C_j} - x_G,\ x_{C_\ell} - x_G \right\rangle \\
&= \frac{n_j n_\ell}{n_j + n_\ell} \left\|x_{C_j} - x_G\right\|^2 + \frac{n_j n_\ell}{n_j + n_\ell} \left\|x_{C_\ell} - x_G\right\|^2 - \frac{2n_j n_\ell}{n_j + n_\ell} \left\langle x_{C_j} - x_G,\ x_{C_\ell} - x_G \right\rangle \\
&= \frac{n_j n_\ell}{n_j + n_\ell} \left(\left\|x_{C_j} - x_G\right\|^2 + \left\|x_{C_\ell} - x_G\right\|^2 - 2 \left\langle x_{C_j} - x_G,\ x_{C_\ell} - x_G \right\rangle \right) \\
&= \frac{n_j n_\ell}{n_j + n_\ell} \left\|(x_{C_j} - x_G) - (x_{C_\ell} - x_G)\right\|^2 \\
&= \frac{n_j n_\ell}{n_j + n_\ell} \left\|x_{C_j} - x_{C_\ell}\right\|^2
\end{aligned}$$

C'est donc cette mesure qui va servir de distance entre classes pour réaliser la classification hiérarchique ascendante :

Définition 3 *La distance de Ward entre deux classes* $(C_j,\ C_\ell)$ *de barycentres respectifs* x_{C_j} *et* x_{C_ℓ} *est définie par*

$$D_W^2(C_j,\ C_\ell) = \frac{n_j n_\ell}{n_j + n_\ell} \left\|x_{C_j} - x_{C_\ell}\right\|^2 \ . \tag{2.2}$$

Nous verrons au paragraphe 2.4.5 qu'il existe d'autres distances entre classes.

Il est important de noter que lorsque l'on utilise la distance de Ward dans l'algorithme CAH, on réalise à chaque étape la fusion optimale au sens de la conservation de l'inertie intra-classe. Cette optimalité locale (à chaque étape) ne garantit aucunement l'optimalité globale, c'est-à-dire l'identification de la partition optimale en K classes. Toutefois, elle justifie l'utilisation de l'algorithme CAH comme heuristique : on espère que la partition trouvée en réalisant la fusion optimale à chaque étape sera proche de la partition optimale.

2.4.3 Exemple

On reprend ici l'exemple à quatre observations du paragraphe 2.3.2. On réalise la classification en choisissant comme distance entre individus la distance euclidienne et comme distance entre classes la distance de Ward. On commence par calculer toutes les distances deux à deux :

	a	b	c	d
a	0	1	4.1	4.5
b		0	3.2	3.6
c			0	1
d				0

Les distances entre a et b d'une part, et c et d d'autre part sont équivalentes et minimales, on peut donc considérer indifféremment la fusion de l'un de ces deux couples. On choisit arbitrairement de fusionner a et b. Il faut maintenant calculer les distances entre la nouvelle classe $\{a,\ b\}$ et c, en utilisant la formule (2.2). Le barycentre de la classe $\{a,\ b\}$ étant (0.5, 1), on a :

$$\begin{aligned} D_W^2\left(\{a,\ b\},\ c\right) &= \frac{2\times 1}{2+1}\, d^2\left(\begin{pmatrix} 0.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}\right) \\ &= 4.83 \\ \Rightarrow D_W\left(\{a,\ b\},\ c\right) &= 2.2 \end{aligned}$$

En calculant de la même manière la distance entre $\{a,\ b\}$ et d, on met à jour le tableau des distances :

	$\{a,\ b\}$	c	d
$\{a,\ b\}$	0	2.2	2.6
c		0	1
d			0

On fusionne maintenant les individus c et d, dont la distance est minimale, et on calcule la distance entre la classe $\{c,\ d\}$ créée à cette étape et la classe $\{a,\ b\}$:

	$\{a,\ b\}$	$\{c,\ d\}$
$\{a,\ b\}$	0	2.91
$\{c,\ d\}$		0

Á la dernière étape, les deux classes restantes sont fusionnées.

2.4.4 Le dendrogramme

Comme le nom de la méthode (CAH) l'indique, les partitions successivement obtenues sont hiérarchisées : elles sont emboîtées les unes dans les autres. De ce fait, il est possible de représenter l'historique des différentes étapes de l'algorithme à l'aide d'une arborescence, aussi appelée dendrogramme. La figure 2.2 montre l'arborescence liée à la CAH réalisée précédemment. En bas de l'arbre se trouvent les individus, jouant le rôle des feuilles. La fusion de deux éléments est représentée par une branche reliant ces deux éléments, dont la hauteur est proportionnelle à la distance entre les deux éléments fusionnés. Ainsi la plus petite branche de l'arbre relie les individus a et b, qui furent les premiers fusionnés. Plus l'on remonte dans l'arbre, plus les classes sont hétérogènes et les branches s'allongent. Dans le cas de la figure 2.2 correspondant à l'exemple considéré, on a choisi de représenter l'arbre avec des hauteurs de branches proportionnelles au pourcentage de perte d'inertie

$$\frac{I_W^{k+1} - I_W^k}{I_T}$$

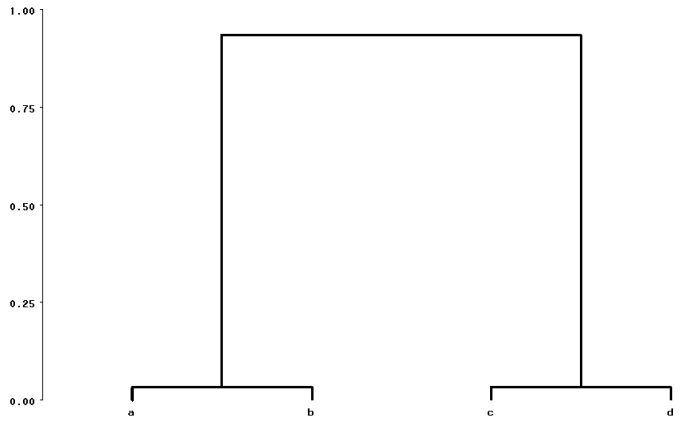


FIGURE 2.2 – Dendrogramme correspondant à la CAH sur les individus a, b, c et d

associée à la fusion des deux éléments (lisible à gauche sur l'échelle), plutôt que proportionnelles à la distance entre ces deux éléments.

Cette représentation donne une vision globale de la topologie des observations, et permet d'identifier des classes : on voit ici que les individus a et b forment une classe homogène et bien distincte de l'autre classe regroupant c et d. Dans l'exemple considéré, une classification en deux classes donne la même classification que celle trouvée avec l'algorithme des K-means. Ce consensus entre les deux méthodes n'est pas représentatif, il est simplement dû au fait que l'exemple considéré est facile : les deux classes sont très disjointes.

Par construction, l'algorithme CAH fournit une classification des données en K classes, pour tout K entre 1 et n. Ainsi, il n'est pas nécessaire de préciser *a priori* le nombre de classes que l'on souhaite. Ce choix peut être réalisé *a posteriori*, en considérant le dendrogramme. Les hauteurs des branches étant proportionnelles à la distance entre classes, on peut choisir une classification en "coupant" l'arborescence lorsque les branches sont jugées trop grandes. En effet, une grande branche indique que l'on regroupe des classes qui ne sont pas homogènes.

2.4.5 Généralisation

La classification hiérarchique ascendante, présentée ici sous l'aspect d'une procédure heuristique pour trouver une solution sous-optimale au problème de recherche de la meilleure partition des observations, fut initialement développée pour répondre au problème plus général de la construction d'une distance ultramétrique. Ce paragraphe est consacré à cette autre approche du problème de classification.

On dispose d'un tableau de distances entre n individus (ou de p mesures pour chacun des individus à partir desquelles on peut construire le tableau de distances). Le nombre d'individus n pouvant être grand, il serait trop fastidieux d'étudier ce tableau de distances pour rendre compte des ressemblances ou des dissemblances entre individus. C'est pourquoi on souhaiterait disposer d'une représentation graphique à la fois simple et fidèle des proximités entre observations ou entre classes d'observations contenues dans le tableau de distances. La stratégie générale de la CAH consiste à représenter ces proximités sous la forme d'une arborescence.

Ce choix peut se justifier empiriquement. Reprenons l'exemple à 4 individus du paragraphe 2.4.4. Supposons qu'au lieu de disposer du tableau des distances entre individus, l'expérimentateur ne dispose que de la représentation sous forme d'arborescence, i.e. du dendrogramme. Les conclusions que pourrait tirer l'expérimentateur de ce graphique seraient les suivantes :

⋆ les individus a et b se ressemblent,

⋆ les individus c et d se ressemblent,

⋆ les individus a et b sont très différents des individus c et d.

On voit donc que l'essentiel de l'information contenue dans le tableau de distance est conservé dans la représentation graphique. Toutefois, la représentation n'est pas parfaite, car si l'on ne dispose que de l'arborescence, il n'est pas possible de préciser qui de a et b est le plus proche de c. L'objectif est donc de construire l'arborescence la plus fidèle aux données.

Si l'on reprend l'exemple précédent, on remarque que nous avons simplement montré qu'il existe des tableaux de distances qui ne peuvent pas être parfaitement représentés par une arborescence. Cela tient en partie au choix de la mesure de distance choisie. Dès lors se pose la question de savoir s'il existe des mesures de distance qui puissent être parfaitement représentées (quel que soit l'échantillon de points sur lequel cette mesure de distance est appliquée) par une arborescence. On définit dans un premier temps les distances ultramétriques :

Définition 4 *Une distance ultramétrique est une fonction d qui à tout couple (x_1, x_2) associe une valeur dans $\mathbb{R}$, et telle que*

⋆ $d(x_1, x_2) = d(x_2, x_1) \geq 0$,

⋆ $d(x_1, x_1) \leq d(x_1, x_2)$,

⋆ $d(x_1, x_2) \leq \sup_{x_3} \{d(x_1, x_3),\ d(x_2, x_3)\}$.

La troisième propriété, plus forte que l'inégalité triangulaire définissant les distances usuelles, caractérise les distances ultramétriques.

Il a été établi (voir par exemple [5]) qu'il est équivalent de se donner, sur un ensemble de points, soit une distance ultramétrique, soit une hiérarchie indicée (une arborescence). Autrement dit, il est possible de parfaitement représenter un tableau de distances entre points dès lors que la mesure de distance utilisée est ultramétrique. On peut alors présenter l'objectif de la classification comme suit : à partir du tableau initial de dissimilarités entre observations, on cherche à définir une distance ultramétrique sur les observations pour pouvoir représenter leurs proximités sous la forme d'un dendrogramme. Bien sûr, on souhaite que la distance ultramétrique reflète au mieux les mesures de dissimilarité entre observations, c'est-à-dire que la distance ultramétrique élaborée soit aussi proche de la mesure de dissimilarité initiale que possible.

L'algorithme présenté au paragraphe 2.4.1 peut être considéré comme une stratégie de construction de l'ultramétrique. Nous avons vu que l'algorithme nécessite de définir à la fois une dissimilarité entre observations et une distance entre classes. L'objectif n'étant plus de conserver l'inertie intra-classe, mais simplement d'obtenir un dendrogramme, il est possible de considérer d'autres distances entre classes que la distance de Ward. De nombreuses distances entre classes ont été proposées et employées dans la littérature, dont voici trois exemples :

– Distance du lien simple : $D(C_j, C_\ell) = \min_{x \in C_j} \min_{x' \in C_\ell} d(x, x')$

– Distance du lien moyen : $D(C_j, C_\ell) = \dfrac{1}{n_j n_\ell} \sum_{x \in C_j} \sum_{x' \in C_\ell} d(x, x')$

– Distance du lien complet : $D(C_j, C_\ell) = \max_{x \in C_j} \max_{x' \in C_\ell} d(x, x')$

Une liste plus complète peut être trouvée dans les documentations de logiciel [1]. Le choix d'une distance particulière constitue la *stratégie d'agrégation.*

Toutes les distances entre classes ne garantissent pas de construire une ultramétrique. Il faut prendre garde à ce que la distance choisie garantisse que si les éléments a et b ont été réunis lors d'une fusion entre classes avant les éléments c et d, alors la branche reliant a à b est plus petite que celle reliant c et d. Il est difficile de caractériser les distances entre classes qui possèdent cette propriété. On se contentera de noter que la propriété est vérifiée pour les distances présentées dans ce document.

Nous avions souligné en partie 2.1.2 la nécessité de choisir avec soin la distance entre individus à employer pour établir la classification. Le choix d'une distance entre classes est tout aussi crucial que le choix d'une distance entre individus, et deux classifications réalisées sur les même individus avec des distances entre classes différentes donneront vraisemblablement des regroupements différents. Dès lors, comment choisir la distance entre classes ? Il n'y a pas de réponse universelle à cette question, car il est très difficile d'étudier les propriétés d'optimalité des distances entre classes. C'est pourquoi le choix de la distance entre classes est bien souvent basé sur l'expérience du statisticien ou sur des considérations pratiques (de temps de calcul par exemple). Dans ce contexte, la distance de Ward introduite en premier lieu a pour avantage d'être basée sur un objectif clairement identifié de minimisation de l'inertie intra-classe, ce qui rend l'algorithme et les résultats interprétables.

2.5 Quelques considérations supplémentaires

Comparaison CAH / K-means Le principal avantage de la CAH vis-à-vis des K-means réside dans sa stabilité. Contrairement aux K-means, la CAH ne nécessite aucune initialisation, en conséquence lancer deux fois l'algorithme CAH sur le même jeu de données donnera deux fois le même résultat. Remarquons que dans certains logiciels, l'initialisation des K-means n'est pas aléatoire : les centres de classes initiaux sont choisis suivant une stratégie déterminée. Ainsi, dans la procédure FASTCLUS du logiciel SAS, le premier centre de classe initial correspond à la première observation du tableau de données. Le second centre de classe initial correspond à la première observation distante de r du premier centre de classe, où r est une distance précisée par défaut. Les centres de classes suivants seront sélectionnés de telle sorte qu'ils soient séparés d'une distance de r de tous les centres de classes précédemment sélectionnés. Ainsi, si la procédure est lancée deux fois de suite sur une même table de données, les résultats obtenus seront identiques. Cette "identité" n'est toutefois qu'illusoire car elle dépend nécessairement de la stratégie d'initialisation choisie. Changer la valeur du paramètre r par exemple résulterait en une nouvelle classification, éventuellement différente de la première.

1. Voir par exemple le chapitre 23 du guide utilisateur du logiciel SAS : SAS/STAT User's Guide, Version 8, 1999. Ce guide est aussi disponible en ligne à l'adresse `http ://www.okstate.edu/sas/v8/saspdf/stat/chap23.pdf`

Utilisation jointe des K-means et de la CAH Il existe des cas où il peut être utile d'utiliser les deux algorithmes conjointement. Le premier cas est celui des grands jeux de données. L'algorithme de la CAH est un algorithme où les premières étapes sont les plus coûteuses, puisqu'elles nécessitent un grand nombre de calculs de distance. Pour de grands jeux de données, le temps de calcul de cet algorithme peut devenir prohibitif. On peut alors réduire le nombre d'individus initial en utilisant les K-means, par exemple pour passer de 1.000.000 d'individus à 10.000 classes, puis réaliser la CAH sur les classes obtenues. Le deuxième cas consiste à utiliser les K-means après la CAH : on constate qu'avec la CAH le classement de deux individus dans une même classe n'est plus remis en cause lors des étapes suivantes. Il peut alors être utile de réaliser la CAH, puis de permettre quelques réallocations des individus en faisant tourner l'algorithme des K-means en partant de la partition obtenue par CAH.

Choix du nombre de classes La difficulté de toute méthode de classification non supervisée résulte dans le choix du nombre de classes K. Il arrive que ce nombre soit directement fixé par la nature même de la question posée, ceci est illustré par l'exemple 4.1 du chapitre 4. Toutefois, dans la plupart des cas, ce nombre est inconnu. Concernant l'utilisation de l'algorithme des K-means, ou de l'algorithme CAH appliqué avec la distance de Ward, il est possible de tracer la courbe de l'inertie intra-classe $W_{C(K)}$ en fonction de K. On cherche alors à identifier les étapes où l'on observe une rupture dans cette courbe, synonyme d'une forte dégradation de l'inertie intra-classe. Cette dégradation résulte de la forte hétérogénéité des deux classes réunies lors de l'étape considérée, il est alors naturel de considérer un nombre de classes supérieur à celui pour lequel la rupture a lieu. Cette stratégie, parfois dénommée "critère du coude", donne des résultats satisfaisants lorsqu'elle est appliquée à l'algorithme CAH où les partitions successives sont emboîtées. Lorsqu'appliquée à l'algorithme K-means (où les partitions successives ne sont pas emboîtées), l'identification des ruptures peut s'avérer plus difficile, rendant le critère du coude moins performant. Remarquons que l'objectif étant *in fine* la mise en évidence de la structure sous-jacente des données, une méthode pragmatique pour le choix du nombre de classes est de choisir une partition dont il sera possible d'interpréter les classes.

Chapitre 3

Méthode probabiliste : le modèle de mélange

Dans cette partie, nous abordons le problème de classification par une approche probabiliste. Cette approche, comme son nom l'indique, fait appel à la modélisation probabiliste.
L'objectif est toujours le même : établir une classification automatique des individus en groupes "homogènes". Le sens donné à l'homogénéité des groupes est ici différent : il ne se base plus sur des considérations géométriques mais s'appuie sur l'analyse de la distribution de probabilité de la population. Nous présentons ici les modèles les plus utilisés qui sont les modèles de mélanges de distributions. La notion d'homogénéité se traduit par le fait que les observations qui sont dans un même groupe sont issues d'une même distribution. Dans ce chapitre, nous parlerons plutôt de populations que de groupes.
Cette approche probabiliste présente deux avantages majeurs. D'une part, il permet d'avoir accès à des probabilités d'appartenance des individus aux différentes populations. C'est d'ailleurs à partir de ces probabilités que s'établit la classification. Il est en effet intéressant de disposer de ces probabilités en plus de la classification pour pouvoir par exemple comprendre le classement d'observations qui peut paraître suspect. D'autre part, le cadre formel de cette approche permet de proposer des solutions théoriques au problème du choix du nombre de populations, qui est en pratique inconnu. On peut en effet utiliser des critères classiques de sélection de modèles.

Au paragraphe 3.1, nous présentons le modèle sous sa forme générale et donnons la règle de classification des individus dans les populations. Une fois le modèle considéré, il s'agit d'estimer les paramètres du modèle. La méthode classique du maximum de vraisemblance ne pouvant être directement utilisée, nous présentons au paragraphe 3.2 l'algorithme utilisé qui est l'algorithme EM. Enfin nous présentons au paragraphe 3.3 deux critères classiques pour le choix du nombre de populations.

3.1 Présentation générale

3.1.1 Modèle

On s'intéresse à n individus pour lesquels on dispose d'observations pour une variable x qui sont notées $x_1, \ldots, x_n$. On suppose qu'en réalité ces individus sont issus de K populations. Dans un premier temps, supposons connu ce nombre de populations. Du point de vue de la modélisation, on suppose que ces observations $x_1, \ldots, x_n$ sont des réalisations de n variables

aléatoires, notées $X_1, \ldots, X_n$, dont chacune est supposée être issue d'une distribution propre à la population à laquelle appartient l'individu associé. Pour le formaliser, on introduit une variable notée Z qui va servir de label pour chaque individu t à classer : à chaque X_t est associé un vecteur de dimension K, noté $Z_t = \{Z_{t1}, \ldots, Z_{tK}\}$ tel que :

$$Z_{tk} = \begin{cases} 1 & \text{si l'individu } t \text{ appartient à la population } k \\ \\ 0 & \text{sinon.} \end{cases}$$

On note π_k la probabilité que cette variable aléatoire prenne la valeur 1, c'est-à-dire la probabilité que l'individu appartienne à la population k :

$$\pi_k = P(Z_{tk} = 1).$$

Cette probabilité est appelée probabilité *a priori* d'appartenance à la population k puisqu'elle ne prend pas en compte l'information dont on dispose, c'est-à-dire l'observation x_t. Elle représente donc tout simplement la probabilité qu'une observation prise au hasard appartienne à la population k. La somme des événements possibles vaut 1, c'est-à-dire que

$$\sum_{k=1}^{K} \pi_k = 1.$$

Cela revient à dire que les variables aléatoires Z_t ont pour distribution une loi multinomiale de paramètres les probabilités *a priori* :

$$Z_t \sim \mathcal{M}(1; \pi_1, \ldots, \pi_K).$$

Une fois définie l'appartenance des individus aux populations, il s'agit de définir la distribution des observations dans chacune des populations : la distribution de X_t sachant que l'individu t appartient à la population k est notée

$$X_t | Z_{tk} = 1 \sim f_k(x_t),$$

où f_k est la distribution de probabilité attribuée à la population k. Ici on se place dans un cadre paramétrique, c'est-à-dire que f_k est supposée appartenir à une famille de lois paramétrées :

$$f_k(.) = f(.; \theta_k),$$

où θ_k sont les paramètres de la distribution f dans la population k. Il faut donc choisir la famille à laquelle appartient f. La différence entre populations se fait alors au travers des paramètres θ_k. Le choix de cette distribution aura bien sûr des conséquences sur la classification finale. En pratique, ce choix se fait selon la même démarche que pour une modélisation plus classique : on peut s'aider de l'histogramme des observations comme des connaissances *a priori* que l'on a des observations. Il faut cependant préciser que le choix d'une distribution assez complexe pourra poser des difficultés dans l'étape d'estimation des paramètres. C'est pourquoi le choix d'une distribution simple est souvent privilégiée même si elle reflète moins bien la réalité.

On dispose donc de la loi du couple (X_t, Z_t) :

$$f(x_t, z_t) = P(z_t) f(x_t | z_t) = \pi_{z_t} f(x_t | z_t),$$

où $\pi_{z_t} = P(Z_t = z_t)$ ($\pi_{z_t} = \pi_k$ si $z_{tk} = 1$ par exemple), et $f(x_t|z_t)$ est la densité de X_t conditionnellement à $Z_t = z_t$. On peut écrire à partir de la loi du couple (X_t, Z_t) la loi marginale de X_t (la distribution de X_t en considérant toutes les populations) : soit l'individu t appartient à la population 1 avec probabilité π_1 et donc la distribution est $f(.;\theta_1)$,... , soit à la population K avec probabilité π_K et donc la distribution est $f(.;\theta_K)$. Formellement, la distribution de X_t peut s'écrire sous la forme :

$$\begin{aligned} f(x_t;\phi) &= P(Z_{t1}=1)\times f(x_t;\theta_1)+\ldots+P(Z_{tK}=1)\times f(x_t;\theta_K), \\ &= \sum_{k=1}^{K} P(Z_{tk}=1)\times f(x_t;\theta_k), \\ &= \sum_{k=1}^{K} \pi_k f(x_t;\theta_k). \end{aligned} \tag{3.1}$$

C'est ce qu'on appelle *un mélange de distributions* : c'est la somme des distributions des K populations pondérées par la taille de ces populations π_k (la proportion d'individus appartenant à ces populations).
Ainsi dans ce modèle, chaque population k est caractérisée par

- π_k qui représente la proportion d'individus appartenant à la population k,
- θ_k qui sont les paramètres de la distribution de la population k.

Par la suite, on notera $\phi = (\phi_1, \ldots, \phi_K)$ l'ensemble de tous les paramètres du modèle où ϕ_k sont les paramètres spécifiques à la population k : $\phi_k = (\pi_k, \theta_k)$.

Les variables mises en jeu dans un modèle de mélange sont donc les X_t et Z_t, représentées sous forme d'un couple $Y_t = (X_t, Z_t)$, que l'on appelle *données complètes*. Cependant, seules les X_t sont observées et sont appelées *données incomplètes*. On souhaite donc reconstruire les variables d'appartenance Z_{tk}. La figure (Table 3.1.1) illustre le problème de classification de données bidimensionnelles (au lieu de considérer une observation par individus, on en a deux). Ces données sont issues de 3 populations que l'on cherche à retrouver.

3.1.2 Affectation

L'idée naturelle est de classer l'individu t dans la population dont il a le plus de chance d'être issu au vu de sa valeur x_t observée et des caractéristiques des populations. On s'intéresse donc à la probabilité que l'individu t appartienne à la population k sachant que l'on a observé pour cet individu la valeur x_t de la variable X. Cette probabilité est notée τ_{tk} :

$$\tau_{tk} = P(Z_{tk} = 1 | X_t = x_t). \tag{3.2}$$

Elle est appelée la probabilité *a posteriori* que l'individu soit dans la population k (contrairement à la probabilité *a priori* on prend en compte l'information dont on dispose). Par la formule de Bayes, elle s'écrit :

$$\tau_{tk} = \frac{P(Z_{tk}=1) f(x_t;\theta_k)}{f(x_t)},$$

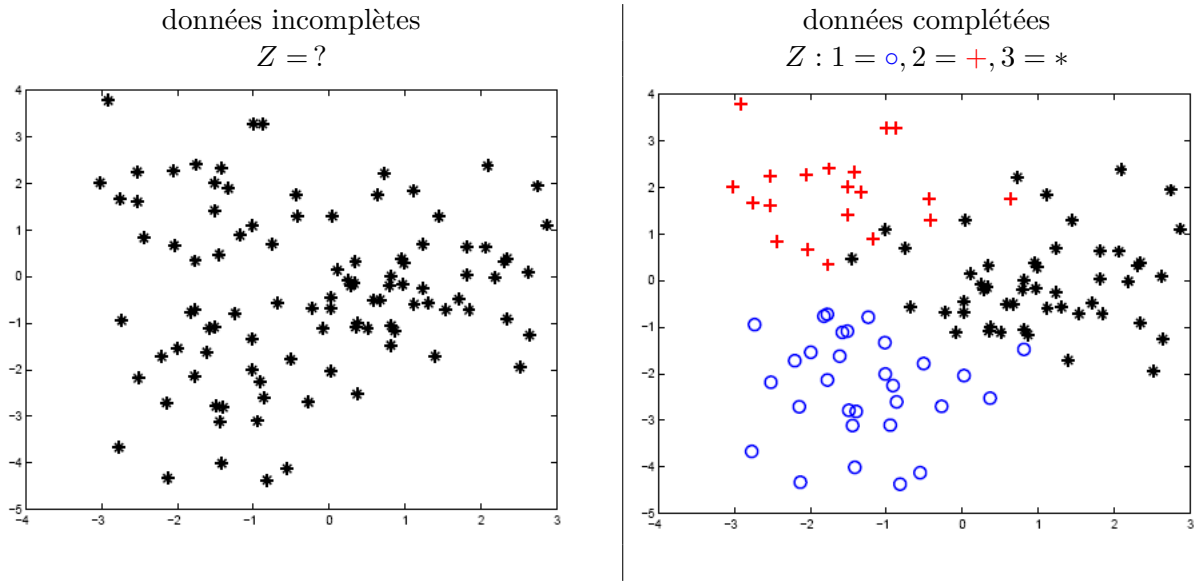


TABLE 3.1 – Illustration du problème.

où $f(x_t)$ est donnée par (3.1). On obtient

$$\tau_{tk} = \frac{\pi_k f(x_t;\theta_k)}{\sum_{l=1}^{K} \pi_l f(x_t;\theta_l)}. \tag{3.3}$$

Pour classer les individus, on utilise la règle du Maximum *A Posteriori* (MAP) : on classe l'individu t dans la population k correspondant à la probabilité τ_{tk} maximale pour cet individu. Si cette quantité est proche de 1 pour une population, on dira que l'individu est classé avec certitude dans cette population. Si par contre toutes les probabilités sont à peu près égales (par exemple 0.51 et 0.49 pour deux populations) alors il est plus difficile de classer avec certitude. On peut remarquer que comme $f(x_t)$ ne dépend pas de la population, la règle du MAP consiste simplement à choisir la population k maximisant $\pi_k f(x_t;\theta_k)$. Ainsi, plus la population k est grande (valeur de π_k élevée), plus elle aura tendance à être attractive.

Pour pouvoir classer les individus dans les différentes populations, il faut connaître les caractéristiques de ces populations, à savoir les paramètres π_k et θ_k. On va donc chercher à les estimer.

3.1.3 Exemple d'un modèle de mélange de distributions gaussiennes

Bien que n'importe quelle loi puisse être utilisée pour modéliser les observations, la plus courante est la distribution gaussienne :

$$\theta_k = (\mu_k, \sigma_k^2) \quad , \quad f_k = \mathcal{N}(\mu_k, \sigma_k^2),$$

dont la fonction de densité s'écrit :

$$f(x;\theta_k) = \frac{1}{\sigma_k\sqrt{2\pi}} \exp\left[-(x-\mu_k)^2/(2\sigma_k^2)\right].$$

Distributions	Probabilités *a posteriori*
$f(x) = \pi_1 f(x;\theta_1) + \pi_2 f(x;\theta_2) + \pi_3 f(x;\theta_3)$	$\tau_{tk} = \Pr(Z_{tk} = 1 \mid x_t)$

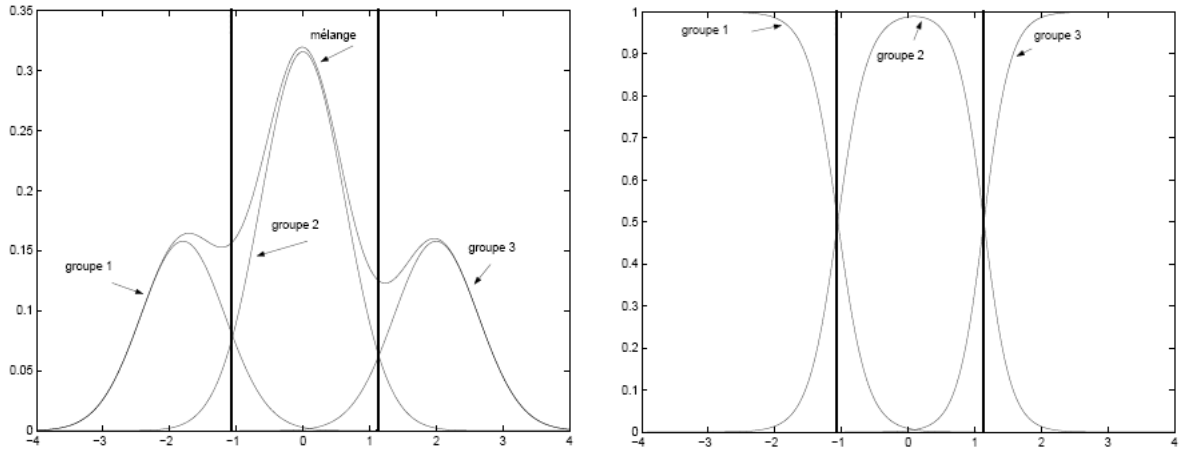


TABLE 3.2 – Exemple d'un distribution de mélange gaussien à 3 populations.

Ici on a considéré que la variance dépendait aussi de la population mais on aurait pu ne pas le supposer et donc avoir une variance σ^2 commune.

La figure (Table 3.1.3) est un exemple de modèle de mélange gaussien à 3 populations. La figure de gauche représente la distribution du mélange $(\pi_1 f(x;\theta_1) + \pi_2 f(x;\theta_2) + \pi_3 f(x;\theta_3))$ avec en détail les distributions dans chaque population pondérées par leur proportion $(\pi_k f(x;\theta_k))$. Sur la figure de droite sont représentées les probabilités *a posteriori* d'appartenir aux 3 populations. Ce dernier graphique permet d'avoir une idée de la classification effectuée : les lignes verticales correspondent aux points frontières (aux valeurs de x pour lesquelles le classement change) : par exemple, tous les individus dont la valeur de x est inférieure à la valeur associée à la première ligne verticale (environ -1 sur la figure) seront classés dans la population 1. C'est d'ailleurs au niveau de ces lignes que le classement se fait avec beaucoup moins de certitude. En effet, c'est à ce niveau que les probabilités sont proches de 0.5. Comme on l'a vu dans le paragraphe précédent, la classification se fait à partir des probabilités τ_{tk} ou plus simplement à partir des valeurs de $\pi_k f(x;\theta_k)$. C'est pourquoi on retrouve les points frontières sur la figure des distributions qui se situent au niveau du croisement des distributions pondérées.

3.2 Estimation des paramètres du modèle

Dans ce paragraphe, on cherche à estimer les paramètres du modèle ϕ par la méthode classique du maximum de vraisemblance. Cette méthode consiste à rechercher les valeurs des paramètres qui maximisent la vraisemblance (ou plutôt le logarithme de la vraisemblance) des données observées (c'est-à-dire des données incomplètes). Compte tenu de (3.1) et de l'indépendance

des observations, la vraisemblance s'écrit :

$$\begin{aligned}\mathcal{L}(X_1,\ldots,X_n;\phi) &= \prod_{t=1}^{n} f(X_t;\phi),\\ &= \prod_{t=1}^{n}\left\{\sum_{k=1}^{K}\pi_k f(X_t;\theta_k)\right\}.\end{aligned}$$

En passant au logarithme, on obtient

$$\log\mathcal{L}(X_1,\ldots,X_n;\phi) = \sum_{t=1}^{n}\log\left\{\sum_{k=1}^{K}\pi_k f(X_t;\theta_k)\right\}. \tag{3.4}$$

On cherche ensuite les valeurs des paramètres θ_k et π_k en annulant la dérivée de la log-vraisemblance par rapport à tous ces paramètres. De par la forme particulièrement complexe de la log-vraisemblance (à cause du logarithme de la somme), on ne peut obtenir des expressions explicites de ces estimateurs.
La solution est d'avoir recours à des algorithmes itératifs de recherche de maximum ou de minimum d'une fonction. Il existe plusieurs algorithmes qui ont cet objectif mais celui qui est utilisé dans le cadre des modèles de mélange est un algorithme appelé *algorithme EM*. Son succès tient dans le fait que cet algorithme est simple à mettre en oeuvre et qu'il mène à des formes explicites des estimateurs.

Cet algorithme est présenté dans le paragraphe suivant.

3.2.1 Algorithme EM

Nous présentons dans ce paragraphe l'algorithme "Expectation-Maximisation" (Espérance-Maximisation en français), abrégé par "EM", proposé par [4]. Puisque la vraisemblance des données incomplètes n'est pas simple à manipuler, l'idée est de travailler plutôt avec la vraisemblance des données complètes qui s'écrit :

$$\begin{aligned}\log\mathcal{L}(X,Z;\phi) &= \log\left\{\prod_{t=1}^{n}\prod_{k=1}^{K}[\pi_k f(X_t;\theta_k)]^{Z_{tk}}\right\},\\ &= \sum_{t=1}^{n}\sum_{k=1}^{K} Z_{tk}\log\{\pi_k f(X_t;\theta_k)\}.\end{aligned} \tag{3.5}$$

Cette quantité n'est pas choisie au hasard. En effet la maximisation de cette quantité va permettre d'atteindre le but fixé au départ, à savoir "maximiser" la log-vraisemblance des données incomplètes (cf paragraphe 3.2.3).
Comme on le voit dans l'expression de la log-vraisemblance (3.5), les variables Z_{tk} apparaissent. Or ces variables ne sont pas observées puisque ce sont celles que l'on cherche à reconstruire. La stratégie consiste alors à remplacer les Z_{tk} non observés par la meilleure prédiction que l'on puisse en faire sachant les données observées X_t, donnée par $E(Z_{tk}|X_t = x_t) = \tau_{tk}$, dans l'expression (3.5). On s'intéressera donc à la quantité suivante :

$$\sum_{t=1}^{n}\sum_{k=1}^{K}\tau_{tk}\log\{\pi_k f(X_t;\theta_k)\}. \tag{3.6}$$

D'après (3.3), les probabilités *a posteriori* τ_{tk} dépendent des paramètres du modèle ϕ_k. Ainsi si on connaissait ces paramètres, on pourrait calculer les probabilités τ_{tk}, et inversement si on connaissait les probabilités τ_{tk}, on pourrait obtenir des valeurs des estimations des paramètres en maximisant la quantité précédente (3.6). L'algorithme EM suit complètement cette démarche. En effet, cet algorithme est un algorithme itératif qui, si on note $\phi^{(h)} = (\pi^{(h)}, \theta^{(h)})$ la valeur du paramètre courant, consiste en deux étapes à l'itération $(h+1)$:

1. **Étape E (Estimation) :** on calcule les probabilités *a posteriori* τ_{tk} à partir de la valeur courante des paramètres $\phi^{(h)}$:

$$\tau_{tk}^{(h+1)} = \frac{\pi_k^{(h)} f(X_t; \theta_k^{(h)})}{\sum_{l=1}^{K} \pi_l^{(h)} f(X_t; \theta_l^{(h)})}.$$

2. **Étape M (Maximisation) :** on actualise les paramètres en maximisant la quantité donnée par (3.6) dans laquelle on a remplacé les τ_{tk} par les valeurs que l'on a obtenues à l'étape E :

$$\phi^{(h+1)} = \underset{\phi}{\text{Argmax}} \sum_{t=1}^{n} \sum_{k=1}^{K} \tau_{tk}^{(h+1)} \log \{\pi_k f(X_t; \theta_k)\}.$$

   Argmax signifie l'argument (valeur des paramètres ϕ) qui maximise la quantité d'intérêt. Il ne faut pas oublier qu'il existe une contrainte sur les proportions π_k, à savoir que $\sum_{k=1}^{K} \pi_k = 1$. On doit alors maximiser la fonction d'intérêt en prenant en compte cette contrainte. La méthode utilisée pour cela est la méthode du multiplicateur de Lagrange dont le principe est de dériver non pas la quantité d'intérêt Q uniquement mais

$$Q + \lambda \sum_{k=1}^{K} \pi_k.$$

   Pour obtenir l'estimateur de π_k, il faut donc résoudre

$$\sum_{t} \tau_{tk}^{(h+1)} + \pi_k \lambda = 0 \quad \Leftrightarrow \quad \pi_k = -\frac{\sum_t \tau_{tk}^{(h+1)}}{\lambda}.$$

   Pour obtenir λ, il suffit d'utiliser la contrainte considérée : en sommant la quantité précédente sur toutes les valeurs de k, on obtient $\sum_k \sum_t \tau_{tk}^{(h)} + \lambda = 0$ et donc $\lambda = -\sum_k \sum_t \tau_{tk} = -n$. En reportant la valeur de λ dans l'équation précédente, on trouve

$$\pi_k^{(h+1)} = \frac{\sum_{t=1}^{n} \tau_{tk}^{(h+1)}}{n}.$$

   Cette estimation a une interprétation naturelle : elle résume la contribution de chaque individu à la population k par leur probabilité *a posteriori* d'appartenance à cette population. Pour les paramètres θ_k, l'expression des estimateurs dépend de la distribution choisie. Notons cependant que ces estimateurs ne dépendent pas de la contrainte ajoutée à Q puisque celle-ci ne porte que sur les paramètres $\pi_1, ..., \pi_K$.

Remarque. On a décrit l'algorithme EM dans le cas particulier des modèles de mélange de distributions en donnant directement dans ce cadre les quantités à calculer dans les deux étapes. D'un point de vue plus général, cet algorithme s'appuie sur l'espérance de la log-vraisemblance des données complètes sachant les données observées X et une valeur des paramètres ϕ' :

$$Q(\phi, \phi') = \mathbb{E}\left\{\log \mathcal{L}(X, Z; \phi) | X, \phi'\right\}. \tag{3.7}$$

Chaque itération de l'algorithme EM consiste dans un premier temps (étape E) à remplacer la log-vraisemblance des données complètes $\log \mathcal{L}(X, Z; \phi)$ par son espérance conditionnelle sachant les données observées X et une valeur des paramètres ϕ', candidate au maximum (3.7). Dans un second temps (étape M), une nouvelle valeur des paramètres ϕ est obtenue en maximisant cette espérance conditionnelle. Ainsi si on note $\phi^{(h)}$ la valeur courante des paramètres, à l'itération $(h+1)$ l'étape E consiste à calculer $Q(\phi, \phi^{(h)})$.

Dans le cas particulier des modèles de mélange de distributions, d'après l'expression de la log-vraisemblance des données complètes donnée par (3.4), on obtient

$$\begin{aligned} Q(\phi, \phi^{(h)}) &= \mathbb{E}\left\{\sum_{t=1}^{n}\sum_{k=1}^{K} Z_{tk} \log\{\pi_k f(X_t; \theta_k)\} | X, \phi^{(h)}\right\}, \\ &= \sum_{t=1}^{n}\sum_{k=1}^{K} \mathbb{E}\left\{Z_{tk} | X, \phi^{(h)}\right\} \log\{\pi_k f(X_t; \theta_k)\}. \end{aligned}$$

Puisque la variable Z_{tk} est une variable aléatoire valant 0 ou 1, $\mathbb{E}\left\{Z_{tk} | X, \phi^{(h)}\right\}$ n'est autre que la probabilité *a posteriori* d'appartenance de l'individu t à la population k en prenant comme valeur des paramètres $\phi^{(h)}$: $P_{\phi^{(h)}}(Z_{tk} = 1 | X_t)$. Ainsi cette étape se réduit au calcul des probabilités *a posteriori*, les $\tau_{tk}^{(h)}$ comme on a vu précédemment.

3.2.2 Calculs pour un modèle de mélange de distributions gaussiennes

Ce modèle est décrit dans le paragraphe 3.1.3. Les paramètres à estimer sont les moyennes et les variances de chaque population. À l'étape M de l'itération $(h+1)$, on obtient :

$$\mu_k^{(h+1)} = \frac{\sum_{t=1}^{n} \tau_{tk}^{(h+1)} x_t}{\sum_{t=1}^{n} \tau_{tk}^{(h+1)}} \quad \text{et} \quad \sigma_k^{2,(h+1)} = \frac{\sum_{t=1}^{n} \tau_{tk}^{(h+1)} (x_t - \mu_k^{(h+1)})^2}{\sum_{t=1}^{n} \tau_{tk}^{(h+1)}}.$$

Ces estimations correspondent aux versions classiques des estimateurs des moyennes et variances mais pondérées par la probabilité *a posteriori* d'appartenance aux populations considérées.

3.2.3 Propriétés de l'algorithme EM

Bien que l'algorithme EM ne consiste pas à maximiser directement la vraisemblance des données observées, il assure que l'on obtiendra des estimations du "maximum" de cette vraisemblance. Ce résultat est énoncé dans la propriété suivante qui montre que la log-vraisemblance des données incomplètes augmente à chaque itération de l'algorithme.

Propriété 1 *Soit une suite d'itérations de EM : $\phi^{(0)}, \phi^{(1)}, \ldots, \phi^{(h)}, \ldots,$, on montre que*

$$\log \mathcal{L}(X; \phi^{(h+1)}) \geq \log \mathcal{L}(X; \phi^{(h)}) \quad \textit{pour tout } h.$$

La démonstration de ce résultat est donnée dans le paragraphe 3.2.6. Cette propriété est fondamentale pour garantir une bonne évolution des valeurs de la log-vraisemblance des données incomplètes (si bien sur, elle est bornée). N'oublions pas, en effet, que l'on cherche les valeurs des paramètres qui maximisent la log-vraisemblance des données incomplètes. Ainsi cette propriété assure la convergence de l'algorithme EM vers une valeur $\phi^\star$ des paramètres réalisant "un maximum" ([4]). Notons que cette valeur n'est pas forcément le maximum de vraisemblance mais peut correspondre à un maximum local.

3.2.4 En pratique

Comme tous les algorithmes itératifs, l'algorithme EM nécessite

- l'initialisation des paramètres $\phi^{(0)}$ ou des probabilités *a posteriori* $\tau_{tk}^{(0)}$ (dans le cadre particulier des modèles de mélange de distributions). En pratique, le plus simple est souvent de choisir ces probabilités. Si on ne dispose d'aucune information *a priori*, l'usage est de les choisir au hasard : $\tau_{tk}^{(0)} = 1/K$.
- une règle d'arrêt : on s'arrête quand les valeurs des paramètres ou de la quantité à maximiser entre deux itérations successives ne varie "presque" plus.

Bien que cet algorithme soit très performant et souvent simple à mettre en oeuvre, son principal problème est sa forte dépendance aux valeurs initiales : pour différentes initialisations, on obtient différentes valeurs des estimations des paramètres ϕ, ce qui le rend peu stable. La raison de cette sensibilité est que l'algorithme converge vers des maxima locaux de la vraisemblance. Pour s'affranchir de ce problème, une solution consiste :

- soit à lancer l'algorithme à partir de plusieurs valeurs initiales et à retenir la meilleure,
- soit à lancer l'algorithme un grand nombre de fois à partir de plusieurs valeurs initiales, à moyenner les valeurs obtenues pour les probabilités *a posteriori* ou pour les paramètres (suivant le mode d'initialisation choisi), puis à lancer une dernière fois l'algorithme EM en utilisant les valeurs moyennes comme valeurs initiales.

Notons qu'une autre solution serait d'initialiser l'algorithme en utilisant les méthodes heuristiques présentées dans le chapitre précédent.

3.2.5 Variantes de l'algorithme EM

Le problème d'initialisation a donné lieu à des développements d'algorithmes dérivés de l'algorithme EM, dont deux sont présentés dans ce paragraphe. Ces algorithmes correspondent à des versions stochastiques de l'algorithme EM, comme par exemple l'algorithme SEM. Pour éviter les maxima locaux, l'idée est d'introduire des perturbations pour permettre à l'algorithme de pouvoir "sortir" de ces maxima, de manière à avoir de plus grandes chances d'atteindre le maximum global de vraisemblance.

Algorithme SEM. C'est une version stochastique de l'algorithme EM qui a été développée par [2]. Il consiste à intercaler une étape S (pour Stochastique) entre les étapes E et M : une fois les τ_{tk} calculées, l'appartenance des individus aux populations, i.e. les Z_{tk}, sont tirées aléatoirement selon ces probabilités. L'estimation des paramètres se fait ensuite sur la base de la classification obtenue : l'étape M consiste à actualiser les paramètres en maximisant

la log-vraisemblance des données complètes (3.5) (cela revient à remplacer τ_{tk} par Z_{tk}).

Algorithme CEM. C'est une version classifiante de l'algorithme EM (C pour Classification) qui a été proposée par [3]. Elle consiste à maximiser directement la log-vraisemblance des données complétées. Plus précisément, à la suite de l'étape E, on effectue la classification par la règle du MAP. Ensuite, dans l'étape M, comme précédemment, c'est la log-vraisemblance complète qui est considérée.

3.2.6 Démonstration de la propriété 1

Dans cette partie, nous présentons la démonstration de la propriété 1. En utilisant la formule de Bayes, on décompose la log-vraisemblance des données incomplètes en la somme de deux termes :

$$\log \mathcal{L}(X;\phi) = \log \mathcal{L}(X,Z;\phi) - \log \mathcal{L}(Z|X;\phi).$$

En passant à l'espérance conditionnelle sachant les données observées et la valeur courante des paramètres $\phi^{(h)}$, notée $\mathbb{E}\left\{.|X,\phi^{(h)}\right\}$, il vient

$$\mathbb{E}\left\{\log \mathcal{L}(X;\phi)|X,\phi^{(h)}\right\} = \mathbb{E}\left\{\log \mathcal{L}(X,Z;\phi)|X,\phi^{(h)}\right\} - \mathbb{E}\left\{\log \mathcal{L}(Z|X;\phi)|X,\phi^{(h)}\right\}.$$

On reconnaît le premier terme qui est $Q(\phi,\phi^{(h)})$ et on note $H(\phi,\phi^{(h)})$ le second. De plus, comme l'espérance d'une fonction de X conditionnellement à X est égale à elle-même, on a

$$\log \mathcal{L}(X;\phi) = Q(\phi,\phi^{(h)}) - H(\phi,\phi^{(h)}). \tag{3.8}$$

Pour montrer l'augmentation de la log-vraisemblance, on s'intéresse à la variation de cette quantité quand on passe d'une itération EM à la suivante. D'après (3.8), cette variation s'écrit :

$$\begin{aligned}\log \mathcal{L}(X;\phi^{(h+1)}) - \log \mathcal{L}(X;\phi^{(h)}) &= \left[Q(\phi^{(h+1)},\phi^{(h)}) - Q(\phi^{(h)},\phi^{(h)})\right] \\ &- \left[H(\phi^{(h+1)},\phi^{(h)}) - H(\phi^{(h)},\phi^{(h)})\right].\end{aligned}$$

Par définition de l'étape M de l'algorithme, Q augmente à chaque itération, c'est-à-dire que la quantité $Q(\phi^{(h+1)},\phi^{(h)}) - Q(\phi^{(h)},\phi^{(h)})$ est positive ou nulle. Il suffit donc maintenant de montrer que le second terme est négatif (c'est-à-dire que H diminue) :

$$\begin{aligned}H(\phi^{(h+1)},\phi^{(h)}) - H(\phi^{(h)},\phi^{(h)}) &= \mathbb{E}\left\{\log \mathcal{L}(Z|X;\phi^{(h+1)}) - \log \mathcal{L}(Z|X;\phi^{(h)})|X,\phi^{(h)}\right\}, \\ &= \mathbb{E}\left\{\log\left(\frac{\mathcal{L}(Z|X;\phi^{(h+1)})}{\mathcal{L}(Z|X;\phi^{(h)})}\right)|X,\phi^{(h)}\right\}.\end{aligned}$$

On majore H en utilisant l'inégalité de Jensen qui dit que étant donnée g une fonction convexe, $g\left(E[X]\right) \leq E[g(X)]$. Le logarithme étant une fonction convexe, on obtient

$$\begin{aligned}H(\phi^{(h+1)},\phi^{(h)}) - H(\phi^{(h)},\phi^{(h)}) &\leq \log\left[\mathbb{E}\left\{\frac{\mathcal{L}(Z|X;\phi^{(h+1)})}{\mathcal{L}(Z|X;\phi^{(h)})}|X,\phi^{(h)}\right\}\right], \\ &\leq \log\left(\int \frac{\mathcal{L}(z|X;\phi^{(h+1)})}{\mathcal{L}(z|X;\phi^{(h)})}\mathcal{L}(z|X;\phi^{(h)})dz\right), \\ &\leq \log 1 = 0.\end{aligned}$$

Ce qui termine la preuve.

3.3 Choix du nombre de populations

La procédure précédente permet d'obtenir une classification des n individus en K populations. Cependant, en pratique le nombre de populations est inconnu. Même si l'on dispose d'informations *a priori* sur les données, il est difficile de le fixer à l'avance. Il faut donc l'estimer. L'avantage du cadre probabiliste des modèles de mélange par rapport aux méthodes exploratoires présentées dans le chapitre précédent est que l'on dispose de critères théoriques pour choisir ce nombre. Ces critères sont appelés *critères pénalisés*.

3.3.1 Critères de sélection

Notons $\hat{\phi}_K$ les estimations du maximum de vraisemblance pour un modèle de mélange à K populations (obtenus par l'algorithme EM). Pour un modèle de mélange à K populations, les critères pénalisés s'écrivent sous la forme générale suivante :

$$Crit(K) = \log \mathcal{L}(X; \hat{\phi}_K) - pen(K),$$

où

- $\log \mathcal{L}(X; \hat{\phi}_K)$ est la log-vraisemblance des données observées prise en son maximum (cf equation (3.4) calculée en $\hat{\pi}_k$ et $\hat{\theta}_k$),
- $pen(K)$ est ce que l'on appelle *la pénalité*.

Le premier terme traduit l'ajustement du modèle aux données. Ce terme augmente avec le nombre de populations K : plus on considère de populations, mieux on s'ajuste. Pourquoi a-t-on besoin de pénaliser cette vraisemblance ? Comme l'ajustement augmente avec le nombre de populations, il suffit de choisir le nombre de populations maximal pour avoir le meilleur ajustement. Le problème est que plus on a de populations, plus il y a de paramètres à estimer et plus les erreurs d'estimation sont nombreuses. On ne veut donc surtout pas sélectionner un nombre de populations trop grand. On voit apparaître le rôle de la pénalité qui va être de faire payer le coût statistique de l'ajustement et donc de contrôler le nombre de populations à sélectionner. Cette pénalité doit donc être une fonction qui augmente avec le coût statistique. Dans la plupart des critères pénalisés connus, ce coût statistique est fonction du nombre de paramètres à estimer du modèle considéré.
On choisit alors le nombre de populations $\hat{K}$ comme étant celui qui maximise le critère pénalisé considéré :

$$\hat{K} = \underset{K=1,\ldots,n}{\text{Argmax}}\, Crit(K).$$

Le critère le plus utilisé pour les modèles de mélange est le critère BIC (Bayesian Information Criterion [10]) qui s'écrit

$$BIC(K) \quad = \quad \log \mathcal{L}(X; \hat{\phi}_K) - \frac{\log n}{2} \times \text{nombre de paramètres du modèle à } K \text{ populations}$$

Par exemple, dans le cas du modèle de mélange de distributions gaussiennes, la pénalité est

$$pen(K) = \frac{\log n}{2} \times ((K-1) + 2\ K).$$

En effet, il y a $K-1$ proportions π_k (K proportions moins une contrainte $\sum_{k=1}^{K} \pi_k = 1$), et 2 K paramètres (K moyennes et K variances).
Ce critère est en général utilisé lorsque l'on se place dans l'objectif de l'ajustement d'une distribution complexe par une distribution de mélange.
Dans un objectif de classification, un critère ressemblant a été proposé par [1] : le critère ICL (Integrate Classification Likelihood). Il consiste à pénaliser la log-vraisemblance complète et prend ainsi en compte la classification à travers les Z :

$$ICL(K) \quad = \quad \log \mathcal{L}(X, \hat{Z}; \hat{\phi}_K) - \frac{\log n}{2} \times \text{nombre de paramètres du modèle à } K \text{ populations},$$

où $\hat{Z}$ correspond à la classification obtenue. On peut montrer que

$$\log \mathcal{L}(X, Z; \phi_K) = \log \mathcal{L}(X; \phi_K) + \sum_{t=1}^{n} \sum_{k=1}^{K} Z_{tk} \log \tau_{tk}.$$

Le critère ICL pour un mélange de distributions à K populations s'écrit :

$$\begin{aligned} ICL(K) \quad &= \quad \log \mathcal{L}(X; \hat{\phi}_K) + \sum_{t=1}^{n} \sum_{k=1}^{K} \hat{\tau}_{tk} \log \hat{\tau}_{tk} - \frac{\log n}{2} \times \text{nombre de paramètres du modèle } K, \\ &= \quad BIC(K) + \sum_{t=1}^{n} \sum_{k=1}^{K} \hat{\tau}_{tk} \log \hat{\tau}_{tk} \end{aligned}$$

en remplaçant $\hat{Z}_{tk}$ par $\hat{\tau}_{tk}$.
Le nouveau terme (par rapport au critère BIC) est appelé *entropie*. Il mesure la séparabilité des populations : pour des populations très séparées, les probabilités *a posteriori* τ_{tk} sont proches de 0 ou 1 et ce terme d'entropie est proche de 0. Dans le cas contraire, ce terme est négatif. On peut donc voir l'entropie comme un terme de pénalisation de la séparabilité des populations : on pénalise les modèles dont les populations sont "mal séparées".

Remarque. La classification par modèle de mélange nécessite le choix de la distribution des données à classer f_k. Cependant, il n'est pas toujours facile de se donner une distribution *a priori* et on peut vouloir regarder plusieurs distributions. Si les distributions en compétition possèdent un même nombre de paramètres, on choisira celle qui permet d'obtenir le meilleur ajustement du modèle aux données, c'est-à-dire celle qui a la plus grande vraisemblance. Par contre si elles ne possèdent pas le même nombre, il faudra prendre en compte le nombre de paramètres en utilisant un critère pénalisé : on choisira la distribution qui mène aux plus grandes valeurs du critère.

3.3.2 En pratique

En pratique, on effectue la classification des données pour différentes valeurs de nombre de populations K. Pour chacune, on calcule l'ajustement (soit $\mathcal{L}(X; \hat{\phi}_K)$). Ensuite, on choisit le K qui maximise le critère pénalisé, c'est-à-dire qui mène au meilleur compromis entre un assez bon ajustement aux données et un nombre raisonnable de paramètres à estimer.

Quelques points pratiques :

– on ne fait pas varier la valeur du nombre de populations de 1 à n. Un K bien trop grand sera de toute façon éliminé, c'est pourquoi on se restreint souvent à une valeur maximale du nombre de populations K, K_{max} qui ne doit bien sûr pas être trop petite. Il n'y a pas de règle pour choisir ce nombre maximal de populations à visiter. Il est souvent fixé comme supérieur au nombre de populations attendu.
– il est toujours informatif de réaliser le graphique représentant le comportement de la log-vraisemblance $\mathcal{L}(X; \hat{\phi}_K)$ en fonction du nombre de populations. Ce graphique est instructif pour deux raisons : d'une part, il permet de voir si l'ajustement augmente bien en fonction du nombre de populations. Comme on l'a vu dans le paragraphe 3.2.4, l'algorithme EM ne garantit pas d'obtenir le maximum de vraisemblance mais peut fournir un maximum local. Ainsi il se peut que la vraisemblance n'augmente pas pour deux valeurs de K successives. Un comportement trop erratique de la vraisemblance peut poser un souci pour la sélection du nombre de populations par critère pénalisé. Les variantes de l'algorithme EM présentées au paragraphe 3.2.5 permettent en général d'obtenir des valeurs de vraisemblance plus proches du maximum. D'autre part, le graphique peut permettre d'obtenir très rapidement une gamme de valeurs raisonnables pour le nombre de populations à sélectionner. En effet, idéalement la vraisemblance augmente fortement pour les premières valeurs de K, signifiant que l'on gagne fortement en ajustement. Puis à partir d'une certaine valeur K_m, cette augmentation devient beaucoup moins importante : augmenter le nombre de populations K n'améliore plus que marginalement l'ajustement mais continue d'augmenter le nombre de paramètres à estimer. Théoriquement, la valeur K_m à partir de laquelle on observe cette rupture (appelée "coude") dans l'amélioration de la vraisemblance se trouve au voisinage du "vrai" nombre de populations. On prendra souvent la valeur K_m comme estimation de cette vraie valeur.

3.4 Logiciels

L'utilisation de plus en plus intensive de ce modèle a suscité l'apparition de logiciels ou packages. On peut citer *mclust*, une fonction du logiciel gratuit R et *Mixmod*, un logiciel entièrement dédié aux modèles de mélange et qui est interfacé avec Scilab ou Matlab. Ces différents outils sont téléchargeables sur internet. Remarquons que ces outils considèrent des choix de distributions assez classiques. En particulier, le modèle de mélange gaussien est intégré dans chacun de ces outils. Par contre, une distribution plus complexe peut ne pas y être.

Chapitre 4

Exemples d'application

4.1 Caractérisation de races anciennes de poules

4.1.1 Présentation du problème

Dans le contexte récent de diversification des activités agricoles et de traçabilité, les démarches de qualification des produits impliquant des races anciennes se multiplient en France dans une perspective de développement local. La motivation initiale est d'associer la sauvegarde de la race avec l'offre d'un produit de qualité. Ainsi, lorsqu'une race satisfait à la fois aux exigences de conservation et de valorisation, éleveurs et producteurs peuvent s'associer pour définir une stratégie de valorisation de cette race.

Encore faut-il que la race ait un sens, c'est-à-dire que la spécificité de cette race par rapport aux autres puisse être établie. Les producteurs de races anciennes se basent en général sur des outils très frustes (standards phénotypiques, périmètre géographique) pour garantir l'authenticité de la race. Le but est donc ici de réaliser une classification de races de poules à partir de données génétiques (marqueurs moléculaires), et de vérifier si la classification basée sur les génotypes recouvre bien la classification en races établie actuellement sur la base des phénotypes [1]. Pour cela, 414 poules ont été génotypées en 22 marqueurs microsatellites. Ces animaux appartiennent à 14 races différentes, avec les effectifs suivants :

Barbezieux (BAZ) : 30	Gauloise Grise (GLG) : 30
Bourbonnaise (BNA) : 30	Gauloise Noire (GLN) : 30
Bresse Blanche (B99) : 30	Géline de Touraine (GLT) : 30
Coucou de Rennes (COU) : 30	Gournay (GOU) : 30
Crèvecoeur (CRC) : 26	Houdan (HOU) : 30
Gasconne (GAS) : 30	Marans (MR) : 30
Gauloise dorée (GLD) : 28	Noire de Challans (NC) : 30

La figure 4.1 représente la répartition géographique de ces 14 races en France.

1. Apport des Marqueurs Moléculaires à la Caractérisation des Races Anciennes de Poules, X. Rognon, C. Berthouly, G. Coquerelle, H. Legros, M. Tixier-Boichard, Septièmes Journées de la Recherche Avicole, 2007

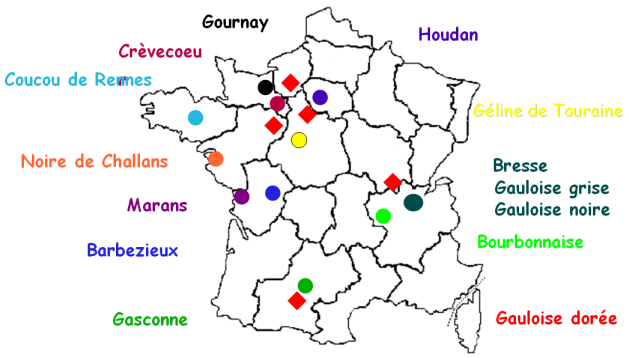


FIGURE 4.1 – Localisation des races étudiées.

4.1.2 Formalisation

Le tableau de données est le suivant :

Race	Identifiant	Locus 1	Locus 2	Locus 3	...
B99	1366	127 127	107 111	116 119	...
B99	1367	127 127	107 107	110 120	...
B99	1368	127 127	107 111	119 120	...

Chaque observation est ici décrite par la combinaison d'allèles observée en différents loci. Par exemple, la poule d'identifiant 1366 est de race Bresse Blanche (B99), et elle est homozygote au locus 1 (elle possède deux fois l'allèle 127). On peut supposer qu'il existe K sous-populations, se distinguant les unes des autres par des fréquences alléliques spécifiques en chacun des loci.
Il est clair que contrairement à l'exemple 2.4.3 décrit au chapitre 2.5, les différentes variables décrivant ici les observations ne sont pas quantitatives. Il n'est donc pas possible d'appliquer l'algorithme des K-means pour classer les individus. Il est en revanche possible de modéliser les données en utilisant un modèle de mélange, au prix d'une adaptation de la méthode décrite au chapitre 3. Les données ne suivant pas ici des lois normales multivariées (puisque les variables ne sont pas quantitatives), la distribution des allèles aux différents loci peut être modélisée par un modèle de mélange de lois multinomiales. Une fois ce changement de modélisation opérer, il est toujours possible d'utiliser l'algorithme EM pour estimer les paramètres des différentes lois, ainsi que les probabilités *a posteriori* associées à chaque individu.

Toutefois, ce n'est pas la stratégie qui est envisagée ici. Bien que les variables ne soient pas quantitatives, il n'en demeure pas moins possible de calculer des distances génétiques entre observations. Ces distances sont calculées en fonction des fréquences alléliques en chaque locus, le lecteur intéressé pourra consulter [6] ou [7].

A partir du calcul des distances entre observations, il va être possible d'appliquer la stratégie de classification hiérarchique ascendante décrite au chapitre 2.4. Le seul paramètre à choisir est la stratégie d'agrégation, c'est-à-dire la distance entre classes. Nous travaillerons dans la suite avec la distance de Ward.

4.1.3 Classification

Nous présentons ici le détail des différentes étapes de l'analyse, réalisée à l'aide du logiciel SAS. Le programme correspondant à cette analyse est disponible en annexe, ainsi que le programme R pour réaliser la même analyse.

Le tableau de données correspond à la matrice de distance entre individus. Autrement dit, ici les individus sont représentés à la fois en ligne et en colonne, le croisement de la ligne i et de la colonne j donnant la distance entre les individus i et j :

Obs	identifiant	race	var1	var2	var3	var4	var5	
1	1366	B99	0.00000	0.43182	0.38636	0.43182	0.36364	...
2	1367	B99	0.43182	0.00000	0.38636	0.43182	0.47727	...
3	1368	B99	0.38636	0.38636	0.00000	0.40909	0.38636	...
4	1369	B99	0.43182	0.43182	0.40909	0.00000	0.36364	...
5	1370	B99	0.36364	0.47727	0.38636	0.36364	0.00000	...
			...					
			...					

Comme on le voit, le logiciel SAS ne renomme pas les colonnes par l'identifiant des individus par défaut : "var1" correspond à la poule 1366, "var2" à la poule 1367 et ainsi de suite. Par ailleurs, outre la matrice de distance, le tableau contient la race correspondant à chaque observation. A partir de ce jeu de données, la classification ascendante hiérarchique s'effectue à l'aide de la procédure CLUSTER de SAS. On obtient alors l'historique de la classification suivant :

The CLUSTER Procedure
Ward's Minimum Variance Cluster Analysis

Historique des classifications

NCL	--Classifications jointes---		FREQ	SPRSQ	RSQ	Tie
413	5938	5939	2	0.0000	1.00	T
412	5941	5942	2	0.0000	1.00	T
411	5944	5945	2	0.0000	1.00	
410	1572	1573	2	0.0000	1.00	
409	1571	1574	2	0.0000	1.00	T
408	CL409	1575	3	0.0001	1.00	
407	1544	1549	2	0.0001	1.00	T
		...				

On rappelle qu'à l'initialisation, chaque observation est classée dans sa propre classe (cette étape initiale n'apparaît pas dans l'historique). Chaque ligne correspond ici à une étape de fusion de l'algorithme de classification, et se lit comme suit. À la première étape, les classes (i.e. les observations) 5938 et 5939 sont fusionnées, donnant naissance à la classe CL413, CL signifiant "cluster", et 413 correspondant à la fois au nombre de classes à la fin de cette étape et au nom de l'étape (colonne NCL). Cette nouvelle classe est composée de 2 observations (colonne FREQ). Le pourcentage de diminution de l'inertie inter-classe I_B à l'issue de cette étape, c'est-à-dire le rapport

$$\frac{I_B^0 - I_B^1}{I_T}$$

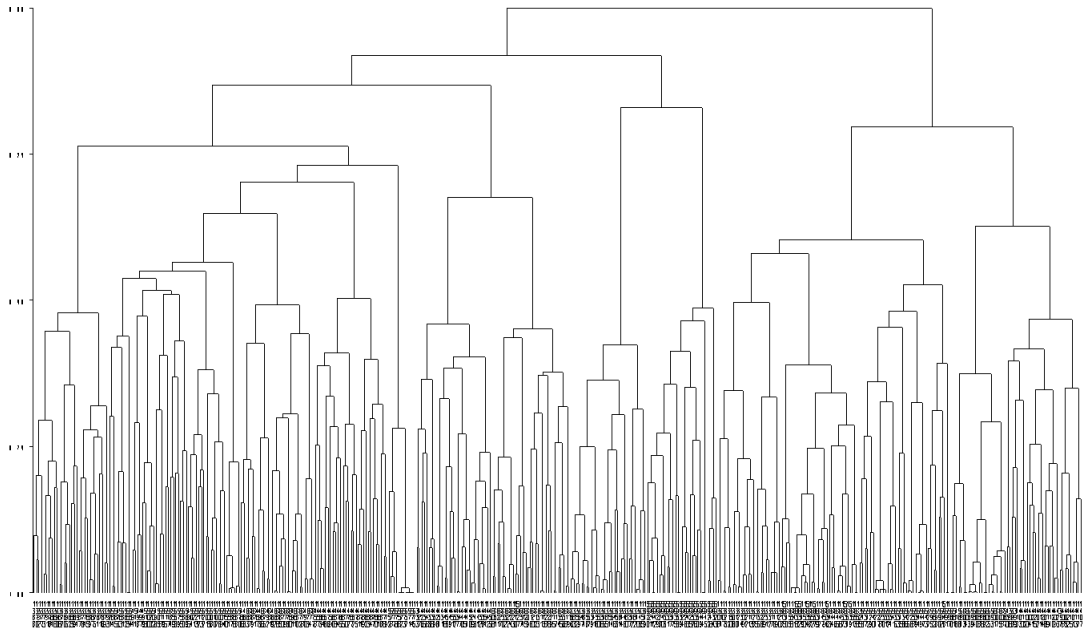


FIGURE 4.2 – Dendrogramme résultant de la classification hiérarchique par distance de Ward sur 414 poules.

est de 0 (colonne SPRSQ), tandis que la diminution du rapport I_B^1/I_T depuis le début se lit dans la colonne RSQ. Enfin, le logiciel SAS précise qu'à cette étape, plusieurs fusions équivalentes étaient possibles, car il existe plusieurs paires d'observations pour lesquelles la distance est minimale (colonne Tie). Dans ce cas, l'une des paires est choisie au hasard.

Plus tard, à l'étape 408, la classe CL409 est fusionnée avec la classe 1575. La classe 1575 contient l'observation 1575, et la classe CL409 a été construite à l'étape 409, en fusionnant les observations 1571 et 1574. À l'étape 408, nous regroupons donc les observations {1571, 1574} et 1575, pour obtenir une classe de 3 observations. Le pourcentage de diminution de l'inertie inter-classe associée à cette étape est de 0.0001. Ainsi, les classes construites au fur et à mesure peuvent être reconstituées en considérant les différentes étapes de fusion. Le dendrogramme résume cet historique (Figure 4.2). Comme on le voit, cette représentation est difficilement exploitable dans son ensemble, mais nous verrons au paragraphe suivant qu'il peut être fructueux d'interpréter une sous-partie du dendrogramme.

4.1.4 Interprétation des résultats

L'objectif premier de l'étude était de vérifier si la classification à partir des distances génétiques permet de reconstituer la classification en race connue. Pour cela, nous pouvons comparer la classification en 14 classes issue de la CAH à la classification en 14 races. La répartition obtenue est donnée par la table 4.1. Cette répartition montre clairement une bonne cohérence entre classifications génotypiques et phénotypiques : deux races forment des classes parfaitement homogènes (BNA et B99), et 8 autres races forment des classes homogènes à 1 ou 2 individus près (GLD, HOU, MR, GAS, GLG, GOU, GLN et GLT).

On constate deux exceptions notables : les races Noire de Challans et Gaulloise Dorée se répartissent en deux classes différentes. Concernant les Noires de Challans, la classe 2 contient 11 poules, toutes de cette race, tandis que la classe 6 contient 19 Noires de Challans et l'ensemble des poules de race Coucou de Rennes. Une étude plus poussée peut ici être réalisée en prenant en compte l'éleveur d'origine des différentes poules. Rappelons qu'ici les poules d'une même espèce proviennent de différents élevages. Les différents éleveurs achetant fréquemment des poules de la même espèce mais provenant d'un élevage différent, le brassage génétique est assuré entre les poules d'une même espèce.

Les Noires de Challans constituent une exception. En effet, pendant plusieurs décennies seuls

Table de race par CLUSTER														
race	CLUSTER													
Fréquence Pourcentage Pourct. en ligne Pourct. en col.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
B99	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	30 7.25 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
BAZ	0 0.00 0.00 0.00	0 0.00 0.00 0.00	2 0.48 6.67 5.56	1 0.24 3.33 3.23	0 0.00 0.00 0.00	1 0.24 3.33 1.92	25 6.04 83.33 100.00	0 0.00 0.00 0.00	1 0.24 3.33 3.33	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
BNA	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	30 7.25 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
COU	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	30 7.25 100.00 57.69	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
CRC	0 0.00 0.00 0.00	0 0.00 0.00 0.00	26 6.28 100.00 72.22	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
GAS	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	1 0.24 3.33 1.92	0 0.00 0.00 0.00	29 7.00 96.67 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
GLD	0 0.00 0.00 0.00	0 0.00 0.00 0.00	7 1.69 25.00 19.44	0 0.00 0.00 0.00	21 5.07 75.00 95.45	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
GLG	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	29 7.00 96.67 96.67	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	1 0.24 3.33 3.33	0 0.00 0.00 0.00
GLN	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	1 0.24 3.33 4.55	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	29 7.00 96.67 96.67	0 0.00 0.00 0.00
GLT	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	1 0.24 3.33 1.92	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	29 7.00 96.67 100.00
GOU	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	30 7.25 100.00 96.77	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00

Table de race par CLUSTER														
race	CLUSTER													
Fréquence Pourcentage Pourct. en ligne Pourct. en col.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
HOU	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	30 7.25 100.00 96.77	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
MR	28 6.76 93.33 100.00	0 0.00 0.00 0.00	1 0.24 3.33 2.78	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	1 0.24 3.33 3.23	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
NC	0 0.00 0.00 0.00	11 2.66 36.67 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	19 4.59 63.33 36.54	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00
Total	28 6.76	11 2.66	36 8.70	31 7.49	22 5.31	52 12.56	25 6.04	29 7.00	30 7.25	31 7.49	30 7.25	30 7.25	30 7.25	29 7.00

TABLE 4.1 – Croisement classification - races.

3 éleveurs (Daviaud, Soret et Milcent, appelés dans la suite "éleveurs noyaux") ont élevé des poules de cette espèce, en n'opérant que très peu d'échanges d'un élevage à l'autre. Puis d'autres éleveurs ont acquis des spécimens de l'espèce auprès des 3 éleveurs noyaux, toujours sans opérer de brassage génétique par croisement. Dans l'étude considérée, les poules Noires de Challans ont été collectées dans 6 élevages différents, mais il est possible de retracer leur appartenance à l'un des 3 élevages noyaux. Cette appartenance est donnée dans le tableau suivant (l'appartenance est indiquée par la colonne "Origine") :

Obs	identifiant	origine	Eleveur
1	1550	BoisJoub	Milcent
2	1551	BoisJoub	Milcent
3	1552	BoisJoub	Soret
4	1553	BoisJoub	Daviaud
5	1554	BoisJoub	Daviaud
6	1555	BoisJoub	Milcent
7	1556	BoisJoub	Soret
8	1557	BoisJoub	Soret
9	1558	Durancea	Milcent
10	1559	Durancea	Milcent
11	1560	Durancea	Milcent
12	1561	Milcent	Daviaud
13	1562	Milcent	Daviaud
14	1563	Milcent	Daviaud
15	1564	Milcent	Daviaud
16	1565	Milcent	Daviaud
17	1566	Daviaud	Milcent
18	1567	Daviaud	Milcent
19	1568	Daviaud	Milcent
20	1569	Daviaud	Milcent
21	1570	Daviaud	Milcent
22	1571	Soret	Soret
23	1572	Soret	Soret
24	1573	Soret	Soret
25	1574	Soret	Soret
26	1575	Soret	Soret
27	1576	Deloison	Soret
28	1577	Deloison	Soret
29	1578	Deloison	Soret
30	1579	Deloison	Milcent

Ainsi, la poule 1555, génotypée dans l'élevage du Bois Joubert, est issue de l'élevage noyau Daviaud.

La comparaison du classement des Noires de Challans obtenu par CAH avec le classement résultant de l'élevage d'origine est représentée en Figure 4.3 (gauche). On constate que la classe

Eleveur	CLUSTER		
Fréquence Pourcentage Pourct. en ligne Pourct. en col.	2	6	Total
Daviaud	0 0.00 0.00 0.00	7 23.33 100.00 36.84	7 23.33
Milcent	0 0.00 0.00 0.00	12 40.00 100.00 63.16	12 40.00
Soret	11 36.67 100.00 100.00	0 0.00 0.00 0.00	11 36.67
Total	11 36.67	19 63.33	30 100.00

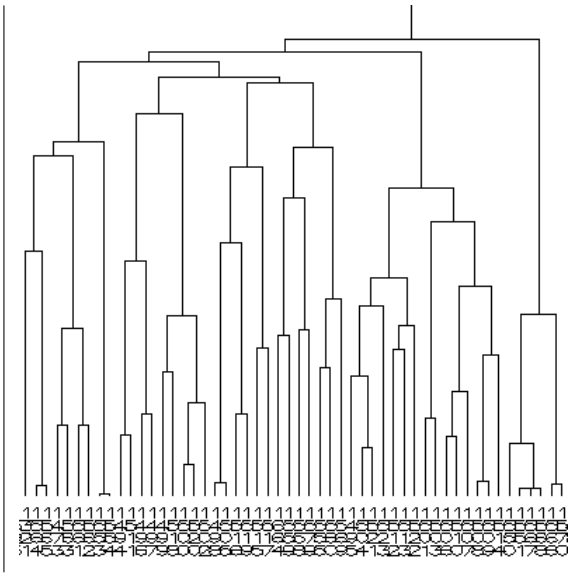


FIGURE 4.3 – **Gauche :** Comparaison classe / élevage d'origine pour les Noires de Challans. **Droite :** Sous-partie du dendrogramme représentant la classe contenant 19 Noires de Challans et 30 Coucous de Rennes.

2 est uniquement composée des poules issues de l'élevage Soret. Par ailleurs, l'analyse de la sous-partie du dendrogramme contenant la classe 6 (Figure 4.3, droite) montre que les Coucous de Rennes et les Noires de Challans, bien que réunies dans une même classe, forment des sous-classes bien séparées. En particulier, une classification ne portant que sur les poules de la classe 6 ferait apparaître un découpage entre Noires de Challans issues de l'élevage noyau Milcent, Noires de Challans issues de l'élevage noyau Daviaud, et Coucous de Rennes.

Une telle typologie, relevant plus des élevages d'origine que des races, pourrait faire douter de l'homogénéité de la race Noire de Challans. En fait, on observe ici un effet de dérive génétique. Les différents animaux d'une même espèce n'ont pas été brassés pendant de nombreuses générations du fait de la politique des éleveurs. Les poules des différents élevages noyaux ont ainsi accumulé des différences génétiques telles qu'elles transparaissent dans la typologie des races. Nonobstant ce cas très particulier, le reste de la typologie illustre que les différentes races ont bien des caractéristiques génétiques différentes.

4.2 Caractérisation de la phyllotaxie d'Arabidopsis

4.2.1 Présentation du problème

La phyllotaxie est l'ordre dans lequel sont implantés les feuilles ou les rameaux sur la tige d'une plante. En effet, la disposition des feuilles le long de la tige ne se fait pas au hasard : pour optimiser la capture des photons, il faut éviter que les feuilles du dessus viennent masquer ou ombrager les feuilles du dessous. Cette disposition est très variée selon les espèces. Pour la plupart, les structures phyllotaxiques appartiennent à deux familles : les structures verticillées (les feuilles apparaissent en même temps au même endroit sur la tige) ou les structures spirallées (les feuilles apparaissent une à une le long de la tige en formant des spirales). Pour cette phyllotaxie spiralée

plantes	angles
1	110
1	130
1	130
1	160
⋮	⋮

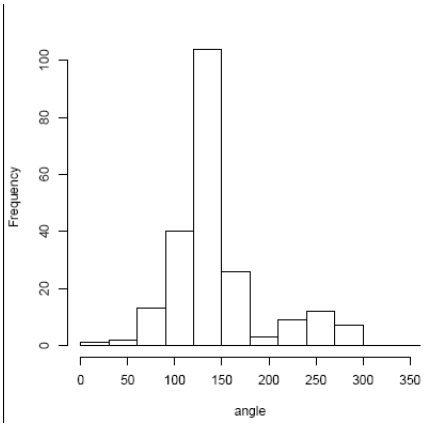


FIGURE 4.4 – Jeu de données (gauche) et histogramme des angles (droite)

(observée chez la plupart des végétaux), l'origine d'une telle structure s'explique par le fait que les feuilles se placent successivement le long de la tige avec un angle constant qui est *l'angle d'or*, égal à 137.5° (cf $http://www.unice.fr/cours\_biologie/Cours\_DSO/cours\_DSO/06-3\_phyllotaxie.htm$ ou $http://www.math.smith.edu/phyllo/$).

Dans cette étude, nous nous intéressons à la structure phyllotaxique des fleurs chez *Arabidopsis thaliana*. Cette plante appartient à la phyllotaxie spiralée [9], on s'attend donc à observer que les angles successifs entre les fleurs soient proches de l'angle d'or. Nous disposons de 9 plantes. Pour chaque plante, les angles successifs à partir de la formation des fleurs ont été mesurés. Le nombre de données (angles) total est de $n = 217$. La table donnée dans la figure 4.4 (gauche) présente les premières lignes du jeu de données : la colonne *plantes* correspond au numéro de la plante et la colonne *angles* aux angles successifs des fleurs (en degré).

La moyenne des angles est de 149.4°. Cette valeur est surprenante, on s'attendait en effet à retrouver une valeur plus proche de la valeur théorique de 137.5°. Quand on regarde l'histogramme de la distribution des angles (figure 4.4), on observe deux modes, un autour de la valeur 140 et un second plutôt autour de la valeur 250. Ces deux modes indiqueraient l'existence de deux groupes d'angles (un groupe d'angles homogène aurait conduit à une seule distribution). De plus, ces deux groupes semblent bien séparés. On peut cependant remarquer que le premier groupe correspond à ce qui était attendu, à savoir une distribution homogène dont l'angle moyen serait proche de la valeur 140.
On cherche à voir si il existe réellement deux groupes d'angles. Ainsi le but est de réaliser une classification. La variable d'intérêt (variable "angles") étant une variable quantitative, il est donc possible d'utiliser toutes les méthodes de classification présentées dans les chapitres 2 et 3.
Nous proposons ici de considérer la classification par modèle de mélange de distributions gaussiennes en considérant des variances communes $\sigma_k^2 = \sigma^2$. Au vu de l'histogramme, supposer la normalité de la distribution de chaque groupe d'angles semble satisfaisant.

4.2.2 Classification par modèle de mélange et interprétation des résultats

Cette classification est réalisée à l'aide du logiciel R. Nous avons utilisé une fonction déjà existante dans ce logiciel, la fonction Mclust. Quelques détails sur cette fonction ainsi que le programme sont donnés en partie 5.3. Avant de présenter les résultats de la classification, nous illustrons le problème de la dépendance de l'algorithme EM à l'initialisation en comparant les résultats de deux classifications obtenues à partir de deux initialisations possibles.

Influence l'initialisation de l'algorithme EM

Dans cette partie, nous illustrons le fait que l'initialisation peut avoir une forte influence à la fois sur les résultats de la classification et sur la sélection du nombre de groupes. Nous considérons deux méthodes d'initialisation :

- init1 : celle utilisée par défaut dans Mclust (cf partie 5.3),
- init2 : une Classification Hierarchique Ascendante. Pour appliquer cette stratégie, il est nécessaire de choisir une distance entre groupes (cf chapitre 2). Nous proposons ici d'utiliser une distance basée sur le rapport de vraisemblances : les groupes C et C' sont réunis si la distance

$$d(C, C') = \log \left\{ \frac{\widehat{\mathcal{L}}(X_C) \times \widehat{\mathcal{L}}(X_{C'})}{\widehat{\mathcal{L}}(X_{C \bigcup C'})} \right\}$$

  est la plus petite parmi toutes les distances entre groupes deux à deux possibles et où $\hat{\mathcal{L}}(X_C)$ est la vraisemblance des données appartenant au groupe C calculée en son maximum.

Influence sur les résultats de la classification. Nous fixons le nombre de groupes à 3 pour illustrer que l'initialisation peut avoir une forte influence sur la classification (avec un nombre de groupes fixé à 2, les deux initialisations donnent le même résultat). La figure 4.5 représente les résultats des classifications obtenues avec init1 (colonne de gauche) et init2 (colonne de droite). Sont représentés en haut les probabilités *a posteriori*, les $\hat{\tau}_{kt}$, pour chaque groupe k (noir : groupe 1, gris foncé : groupe 2 et gris clair : groupe 3) et en bas l'histogramme des données avec les densités des 3 groupes pondérées par les proportions $\hat{\pi}_k f(.; \hat{\mu}_k, \hat{\sigma}^2)$. Rappelons que c'est à partir des probabilités *a posteriori* que l'on obtient la classification : on classe un individu dans le groupe qui a la plus forte probabilité *a posteriori*. On voit clairement que les classifications que l'on obtiendra seront différentes de part les deux premiers groupes (noir et gris foncé) :

- avec init2, les 2 premiers groupes correspondent respectivement à un petit groupe (avec peu de mesures) de petits angles (compris entre 20 et 60) et un grand groupe avec des angles compris entre 80 et 200.
- avec init1, c'est le contraire : le groupe 1 correspond à un groupe d'angles compris entre 20 et 160 et le groupe 2 à un petit groupe d'angles compris entre 170 et 200. Cependant comme on le voit sur le graphe des probabilités *a posteriori*, le classement dans les groupes 1 et 2 pour des angles inférieurs à 200 se fera avec beaucoup moins de certitude (probabilité maximale à 0.8). Cela s'explique par le fait que les deux densités associées modélisent la même distribution (les densités estimées de ces deux groupes sont pratiquement superposées). La classification ainsi obtenue n'est donc pas pertinente.

	$\log \mathcal{L}(X; \hat{\phi}_K)$
init1	-1089.313
init2	-1082.355

TABLE 4.2 – Log-vraisemblances des classifications à 3 groupes obtenues pour les deux initialisations considérées.

La table 4.2 donne les log-vraisemblances calculées pour ces deux classifications : la classification à 3 groupes obtenue avec init2 est donc plus probable que celle obtenue avec init1.

Influence sur la sélection du nombre de groupes. Avec init2, le critère BIC sélectionne 3 groupes alors qu'avec init1 il n'en sélectionne que 2, la classification a 3 groupes avec cette initialisation n'étant "pas assez" probable.

Classification obtenue et interprétation des résultats

Seules les classifications obtenues avec l'initialisation init2 sont considérées. Tout d'abord, il nous faut choisir le nombre de groupes. En effet, l'existence de groupes d'angles différents n'était pas au départ attendue mais a été suggérée par les données. Ainsi la notion de groupe n'a pas de signification précise et donc le nombre de groupes ne peut être déterminé à l'avance (même si l'histogramme de la distribution des angles suggère l'existence de deux groupes seulement). Nous sommes ici dans un objectif de classification, c'est pourquoi nous utilisons le critère ICL qui sélectionne ici 2 groupes.

La table 4.3 résume l'estimation des différents paramètres du mélange à 2 groupes : les moyennes estimées $\hat{\mu}_k$, les groupes d'angles obtenus et le nombre d'individus dans chacun des groupes :

	groupe 1	groupe 2
$\hat{\mu}_k$	132.7	257.27
valeurs de angles	[20 ;200]	[210 ;300]
Effectif	188	29

TABLE 4.3 – Estimation des paramètres du mélange à 2 groupes.

Ainsi nous obtenons deux groupes d'angles : un premier groupe d'angles dont la moyenne est 132.7, c'est le type d'angle le plus observé (86.6% des angles) et un second groupe d'angles plutôt élevé, avec une moyenne de 257.27 et qui contient 13.3% des mesures d'angles. *Arabidopsis* étant de la famille des phyllotaxies spiralées, on s'attendait à ne trouver qu'un seul groupe dont la moyenne des angles serait centrée en 137.5. Une explication de l'existence de ce second groupe est qu'il correspond à l'angle attendu dans le cas d'une inversion de direction de la phyllotaxie. En effet, l'angle attendu dans ce cas est de $360 - 137.5 = 222.5^\circ$.

On pourrait penser que certaines plantes ont une inversion complète de direction de leur phyllotaxie, c'est-à-dire que tous leurs angles successifs sont proches de l'angle 222.5°. La figure 4.6 (gauche) représente les angles par plante codés selon la classification obtenue, triangle pour

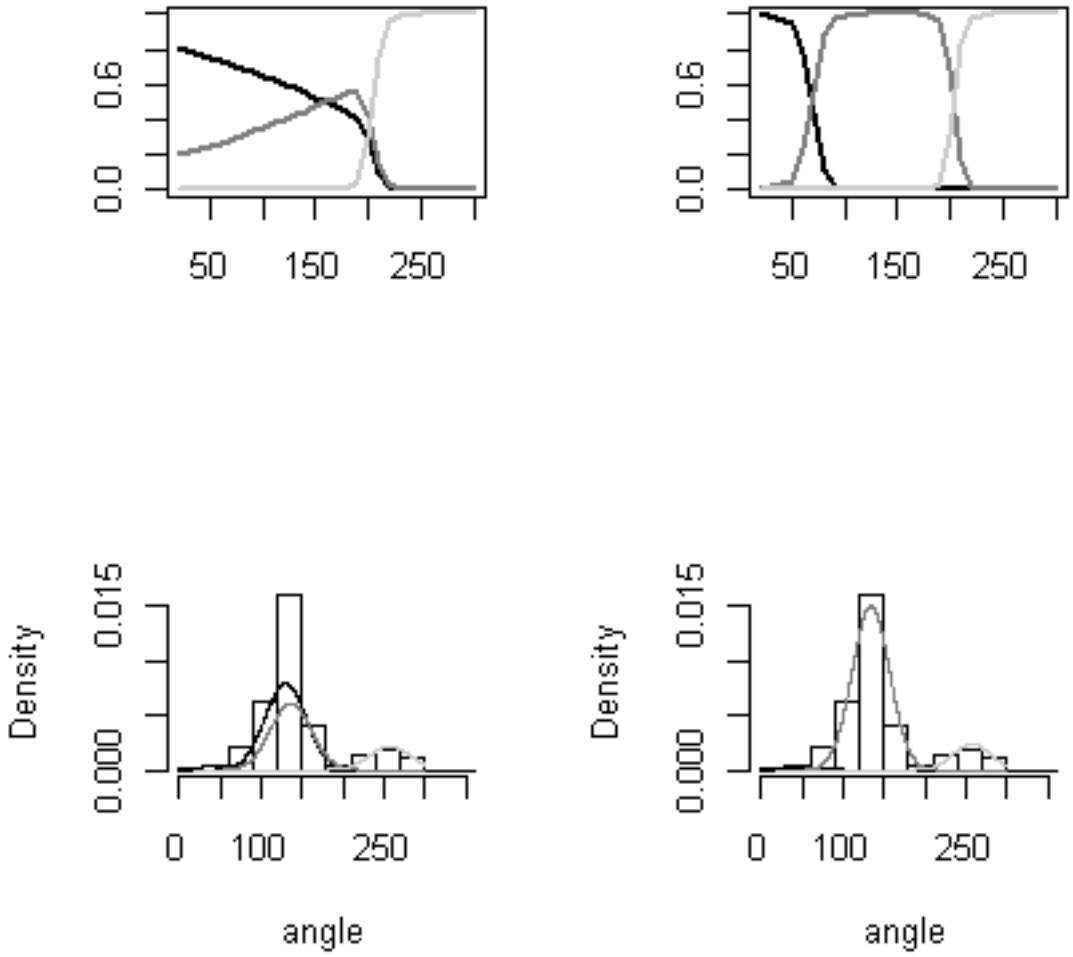


FIGURE 4.5 – Probabilités *a posteriori* et densités pondérées des 3 groupes avec respectivement l'initialisation 1 (à gauche) et 2 (à droite)

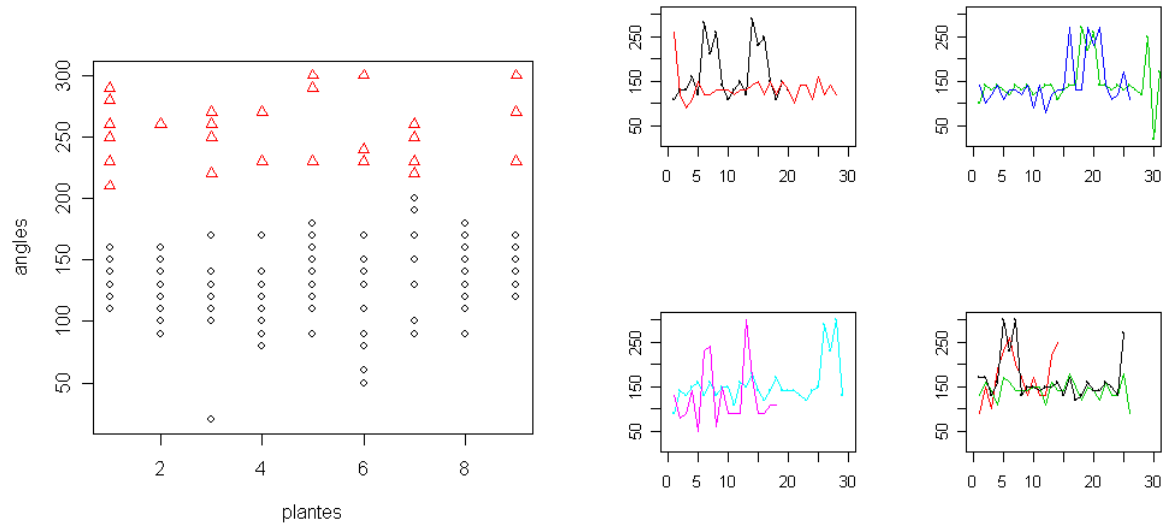


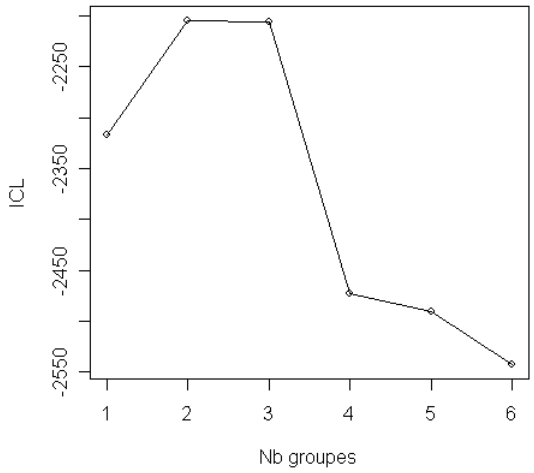
FIGURE 4.6 – Classification à 2 groupes : classification des angles en fonction de la plante (à gauche) et profils des angles par plante (à droite).

le groupe 2 et rond pour le groupe 1. On observe que ce n'est pas le cas, toutes les plantes sauf une ont des angles qui appartiennent aux deux groupes. De plus, comme on le voit sur la figure des profils des plantes (figure 4.6 à droite) qui représente le profil des angles pour chaque plante, il y a des phases transitoires de changements de direction de la plante.

Ainsi l'apparition de ce second groupe a permis de mettre en evidence que le modèle de phyllotaxie (phyllotaxie spiralée) d'*Arabidopsis* était transitoirement inversé [8].

Discussion sur le choix du nombre de groupes.

Comme on peut le voir sur la figure 4.7 (haut) qui représente l'évolution du critère ICL en fonction du nombre de groupes, ICL(2) et ICL(3) sont très proches. De plus, le critère BIC sélectionne 3 groupes (cf 4.2.2). Il peut donc être intéressant de regarder la classification à 3 groupes. La table donnée en bas de la figure 4.7 résume les estimations des paramètres de ce mélange à 3 groupes. On peut remarquer que le groupe 1 de la classification à 2 groupes est scindé en deux dans la classification à 3 groupes avec un premier groupe tout petit (avec 3 angles). Cependant, ce groupe n'a pas d'interprétation biologique. Nous conservons donc la classification à 2 groupes.



	groupe 1	groupe 2	groupe 3
$\hat{\mu}_k$	49.7	132.7	257.27
valeurs de angles	[20 ;60]	[80 ;200]	[210 ;300]
Effectif	3	185	29

FIGURE 4.7 – Estimation des paramètres du mélange à 3 populations (bas) et critère ICL en fonction du nombre de groupes (haut).

Chapitre 5

Annexes

5.1 Etude des deux étapes de la procédure K-means

L'objectif est de démontrer que l'enchaînement des deux sous-étapes de l'algorithme K-means (redéfinition des centres de classes, réaffectation des points) résulte en une amélioration (i.e. une diminution) de l'intertie intra-classe. On suppose que la distance entre individus employée est la distance euclidienne canonique.

Considérons la fin de l'étape j : les centres de classes $\mu_1^j, ..., \mu_K^j$ ont été calculés, et les classes $C_1^j, ..., C_K^j$ ont été obtenues en classant chacun des n points de l'échantillon avec le centre de classe dont il est le plus proche. À la fin de cette étape, on définit la quantité

$$Q_j = \sum_{k=1}^{K} \sum_{i \in C_k^j} ||x_i - \mu_k^j||^2 \quad .$$

Dans un premier temps (sous-étape de redéfinition), on recalcule les centres de classe $\mu_1^{j+1}, ..., \mu_K^{j+1}$ en fonction des points appartenant à chacune des classes $C_1^j, ..., C_K^j$, d'effectifs respectifs n_k^j. On a :

$$\mu_k^{j+1} = \frac{1}{n_k^j} \sum_{i \in C_k^j} x_i \quad \text{et} \quad W_{j+1} = \sum_{k=1}^{K} \sum_{i \in C_k^j} ||x_i - \mu_k^{j+1}||^2 \quad ,$$

où W_{j+1} est l'inertie intra-classe associée aux classes $C_1^j, ..., C_K^j$. On a :

$$\begin{aligned}
Q_j &= \sum_{k=1}^{K} \sum_{i \in C_k^j} ||x_i - \mu_k^j||^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k^j} ||x_i - \mu_k^{j+1} + \mu_k^{j+1} - \mu_k^j||^2 \\
&= \sum_{k=1}^{K} \sum_{i \in C_k^j} \left(||x_i - \mu_k^{j+1}|| + ||\mu_k^{j+1} - \mu_k^j||^2 + 2\langle x_i - \mu_k^{j+1},\ \mu_k^{j+1} - \mu_k^j \rangle \right) \\
&= W_{j+1} + \sum_{k=1}^{K} \sum_{i \in C_k^j} ||\mu_k^{j+1} - \mu_k^j||^2 + 2 \sum_{k=1}^{K} \left\langle \sum_{i \in C_k^j} (x_i - \mu_k^{j+1}),\ \mu_k^{j+1} - \mu_k^j \right\rangle \\
&\geq W_{j+1} \quad ,
\end{aligned}$$

le vecteur $\sum_{i \in C_k^j} (x_i - \mu_k^{j+1})$ étant le vecteur nul, par définition de μ_k^{j+1}. Dans un deuxième temps (sous-étape de réaffectation), on réaffecte les points aux centres dont ils sont les plus proches. On obtient alors de nouvelles classes $C_1^{j+1}, ..., C_K^{j+1}$ et on définit

$$Q_{j+1} = \sum_{k=1}^{K} \sum_{i \in C_k^{j+1}} ||x_i - \mu_k^{j+1}||^2 \quad .$$

Lors de cette sous-étape, toutes les distances diminuent puisque chaque point x_i de l'échantillon est affecté au centre de classe μ_k^{j+1} minimisant l'écart $||x_i - \mu_k^{j+1}||^2$. On a donc $Q_{j+1} \leq W_{j+1}$. Ainsi, pour tout j, nous avons prouvé l'inégalité suivante

$$Q_{j+1} \leq W_{j+1} \leq Q_j \quad .$$

On a donc en particulier $W_{j+2} \leq Q_{j+1} \leq W_{j+1}$, ce qui achève la démonstration.

5.2 Programmes pour l'analyse des données Poules

5.2.1 Programme SAS

```
/* Importation des donnees */
/* l'option lrecl permet de lire des données (des lignes de fichier) */
/* d'une longueur supérieure à 256 caractères. */
data Poules;
infile 'france_jra_distance.txt' firstobs=5 lrecl=10000 expandtabs;
input identifiant var1-var414;
run;

/* Mise en forme des donnees */
/* L'option type=distance de l'étape data est cruciale : elle spécifie, */
/* pour la proc cluster qui va être utilisée */
/* ensuite, que la table n'est pas une table avec observations en ligne et variables */
```

```
/* en colonne, mais une table où chaque ligne et chaque colonne correspondent  */
/* à des individus, le croisement ligne-colonne donnant la distance entre ces individus.  */

data Poules (type=distance);
set Poules;
race = "RACE";
if (identifiant ge 1306) and (identifiant le 1335) then race = "BAZ";
if (identifiant ge 1336) and (identifiant le 1365) then race = "BNA";
if (identifiant ge 1366) and (identifiant le 1395) then race = "B99";
if (identifiant ge 1396) and (identifiant le 1425) then race = "GLN";
if (identifiant ge 1426) and (identifiant le 1455) then race = "GLG";
if (identifiant ge 1456) and (identifiant le 1485) then race = "GLT";
if (identifiant ge 1486) and (identifiant le 1493) then race = "GAS";
if (identifiant ge 1494) and (identifiant le 1523) then race = "COU";
if (identifiant ge 1524) and (identifiant le 1549) then race = "CRC";
if (identifiant ge 1550) and (identifiant le 1579) then race = "NC";
if (identifiant ge 1580) and (identifiant le 1607) then race = "GLD";
if (identifiant ge 1608) and (identifiant le 1637) then race = "GOU";
if (identifiant ge 1638) and (identifiant le 1659) then race = "HOU";
if (identifiant ge 1660) and (identifiant le 1681) then race = "GAS";
if (identifiant ge 5832) and (identifiant le 5839) then race = "HOU";
if (identifiant ge 5920) and (identifiant le 5949) then race = "MR";
run;

/* Verification */
proc print data = Poules (obs=10);
id identifiant;
run;

/* Mise en oeuvre de la CAH */
/* L'option copy permet de garder la variable race dans le fichier de sortie "tree", */
/* même si cette variable n'est pas utilisée dans le calcul des distances. */

proc cluster data=Poules method=ward outtree=tree;
id identifiant;
var var1-var414;
copy race;
run;

/* Representation de l'historique par le dendrogramme */
/* L'option height=rsq sert à spécifier que la hauteur des branches doit être */
/* proportionnelle au pourcentage InertieInter / Inertie totale.  */
/* L'option hpages=4 précise que dans la sortie SAS l'arbre doit être représenté */
/* sur 4 pages plutot qu'une, pour être plus lisible. */
/* L'option goptions htext=0.5 spécifie la taille des caractères */
/* dans la sortie SAS, pour éviter que les noms des feuilles */
/* du dendrogramme se superposent. */
goptions htext=0.5 ;
proc tree data = tree nclusters=14 out = classement height=rsq hpages=4  ;
copy race;
run;
```

```
/* Croisement classe*race */
ods rtf file ='P:\math_enseignement\PolysMath\Classif\ExemplePoules\poule.rtf';
proc freq data = classement;
tables race * cluster;
run;
ods rtf close;

/* Etude des Noires de Challans */
/* La table eleveur contient l'élevage et l'élevage d'origine des 30 Noires de Challans*/
data Eleveur;
infile 'liste_poules_NC.txt' firstobs=2 lrecl=10000 expandtabs;
format eleveur identifiant $32. ;
input GENETIX$  identifiant autre$  origine$    eleveur;
drop autre;
run;

/* L'objectif des étapes suivantes est l'obtention d'une table contenant */
/* à la fois l'information sur l'origine des poules et */
/* sur l'appartenance des poules aux différents clusters. */
/* Pour cela, on crée la table NC en fusionnant les tables classementNC et Eleveur. */
/* Cette fusion se fait par identifiant (i.e. par poule),*/
/* l'identifiant doit donc avoir le même format dans chacune des */
/* deux tables à fusionner. */
data classementNC;
set classement;
if race='NC';
identifiant=input(_name_,$32.);
drop _name_;
run;
proc sort data=Eleveur;
by identifiant;
run;
proc sort data=classementNC;
by identifiant;
run;
data NC;
merge eleveur classementNC;
by identifiant;
run;

/* Comparaison Eleveur / Cluster */
proc freq data = NC;
tables eleveur*cluster;
run;
```

5.2.2 Programme R

```
## Importation et mise en forme des donnees ##
rm(list=ls())
Poules <- read.table(file='france_jra_distance.txt',sep='\t',header=F,skip=4)
colnames(Poules) <- c("identifiant",Poules[,1])
race <- rep("BAZ",414)
```

```
race[which((Poules$identifiant >= 1336) & (Poules$identifiant <= 1365))] = "BNA";
race[which((Poules$identifiant >= 1366) & (Poules$identifiant <= 1395) )] = "B99";
race[which((Poules$identifiant >= 1396) & (Poules$identifiant <= 1425) )] = "GLN";
race[which((Poules$identifiant >= 1426) & (Poules$identifiant <= 1455) )] = "GLG";
race[which((Poules$identifiant >= 1456) & (Poules$identifiant <= 1485) )] = "GLT";
race[which((Poules$identifiant >= 1486) & (Poules$identifiant <= 1493) )] = "GAS";
race[which((Poules$identifiant >= 1494) & (Poules$identifiant <= 1523) )] = "COU";
race[which((Poules$identifiant >= 1524) & (Poules$identifiant <= 1549) )] = "CRC";
race[which((Poules$identifiant >= 1550) & (Poules$identifiant <= 1579) )] = "NC ";
race[which((Poules$identifiant >= 1580) & (Poules$identifiant <= 1607) )] = "GLD";
race[which((Poules$identifiant >= 1608) & (Poules$identifiant <= 1637) )] = "GOU";
race[which((Poules$identifiant >= 1638) & (Poules$identifiant <= 1659) )] = "HOU";
race[which((Poules$identifiant >= 1660) & (Poules$identifiant <= 1681) )] = "GAS";
race[which((Poules$identifiant >= 5832) & (Poules$identifiant <= 5839) )] = "HOU";
race[which((Poules$identifiant >= 5920) & (Poules$identifiant <= 5949) )] = "MR ";
head(Poules)

## Mise en oeuvre de la CAH ##
## La fonction hclust travaille à partir d'une matrice de distance ##
## où seule la partie triangulaire inférieure est donnée. Pour     ##
## obtenir une matrice de distance sous cette forme à partir de la ##
## matrice de distance Poules, on utilise la fonction as.dist      ##
Dist <- as.dist(Poules[,-1], diag = FALSE, upper = FALSE)
cah.ward <- hclust(Dist^2,method="ward")
plot(cah.ward)
Membership <- cutree(cah.ward,k=14)

## Croisement classe - race ##
table(race,Membership)

## Etude des Noires de Challans ##
Eleveur <- read.table(file='liste_poules_NC.txt',skip=1,sep='\t',header=F)
colnames(Eleveur)<-c("genetix","adn","identifiant","origine","eleveur")
OrdreEl <- order(Eleveur$identifiant)

ClassementNC = cbind(Poules$identifiant,Membership)[race=="NC ",]
OrdreCl <- order(ClassementNC[,1])
table(ClassementNC[OrdreCl,2],Eleveur[OrdreEl,5])
```

5.3 Programme R pour l'analyse des données de Phyllotaxie

5.3.1 Programme

```
# Importation des donnees
donnees<-read.table("Donnees_angles.txt",header = TRUE, sep = " ")

# Vérification
head(donnees) #donne les premières lignes du tableau de données
names(donnees) #donne les noms des colonnes

# Moyenne, écart-type et histogramme des angles
```

```
mean(donnees$angles)
sd(donnees$angles)
hist(donnees$angles, breaks= seq(0, 360, by =30), xlab="angle")

# Mise en oeuvre du modèle de mélange
angles.Mclust1 <-Mclust(donnees$angles, G = 1:6, modelName = "E")
# L'initialisation de l'algorithme EM est celle utilisée par défaut dans Mclust
# (cf la description de la fonction Mclust donnée dans la section suivante)
# Initialisaion par CAH
hc.angles <- hc(modelName = "E", data= donnees$angles)
angles.Mclust2 <- Mclust(donnees$angles, 1:6,modelName = "E",...
... initialization = list(hcPairs = hc.angles, subset =NULL))
# L'option G=1:6 indique que les classifications à 1,2,...,6 groupes sont réalisées.
# L'option modelName="E" signifie que l'on fait l'hypothèse
# que les variances des groupes sont les mêmes.
names(angles.Mclust1) # permet de savoir quels sont les résultats dont on dispose
# Par exemple
angles.Mclust1$G
# donne le nombre de groupes sélectionné par BIC (par défaut dans Mclust).
angles.Mclust2$parameters$mean
# donne les moyennes estimées des deux groupes


# Calcule du critère ICL pour une classification à 1,2,...,6 groupes
Entropie = c()
Entropie[1] = 0
Gseq = 2:6
for (G in Gseq)
{
res2 <- Mclust(donnees$angles, G,modelName = "E",...
...initialization = list(hcPairs = hc.angles, subset =NULL))
Entropie[G] = 2 * sum(sum(res2$z * log(res2$z))) + }
ICL = angles.Mclust2$BIC + Entropie
#graphe du critère
plot(ICL, xlab = "Nb groupes", ylab ="ICL")
lines(ICL, lty = "solid")


# Représentation graphique des résultats
plot(donnees$plantes, donnees$angles,...
... col = angles.Mclust2$classification,pch =angles.Mclust2$classification)
```

5.3.2 Détails sur la fonction Mclust

La fonction Mclust ne réalise que des classifications par modèle de mélange de distributions gaussiennes. Des hypothèses sur ces distributions peuvent cependant être spécifiées, comme par exemple supposer des variances communes dans les groupes, ou des structures particulières de la matrices de variance-covariance dans le cas multivarié. Pour avoir accès à la description de la fonction Mclust, il faut utiliser la commande ?Mclust. De plus, l'utilisation de cette fonction nécessite l'installation du package *mclust* (cf 5.3.3 pour l'installation de ce package).

Nous précisons ici les méthodes utilisées par défaut dans Mclust concernant tout d'abord l'initialisation de l'EM (cf paragraphe 3.2), puis le critère du choix du nombre de groupes.

Initialisation de l'EM. Comme vu dans le paragraphe 3.2, l'algorithme EM nécessite une étape d'initialisation. La fonction Mclust considère une méthode d'initialisaion par défaut qui diffère selon le cas multivarié (où la classification des individus est réalisée sur plusieurs variables) et le cas univarié qui sont respectivement les suivantes :

- Dans le cas multivarié , Mclust considère comme classification initiale celle obtenue par CAH. La distance utilisée est une distance basée sur le rapport de vraisemblances (cf paragraphe 4.2.2 où cette distance est précisée).
- Dans le cas univarié, c'est une initialisation basée sur les quantiles qui est considérée : l'idée consiste à trier par ordre croissant les données qui seront séparées en K groupes de tailles égales pour une classification en K groupes.

Il est possible de considérer une autre méthode d'initialisation (option *initialization*).

Choix du nombre de groupes. Le nombre de groupes est obtenu par le critère BIC. Seul ce critère est proposé dans Mclust mais il est possible d'avoir accès à la vraisemblance des différentes classifications et ainsi recalculer un autre critère.

5.3.3 Installation du package *mclust*

Pour installer le package *mclust*, il faut

- dans le volet "Packages" de R, aller dans "installer le(s) package(s)", sélectionner par exemple France(Toulouse), une liste de packages apparaît, il ne vous reste plus qu'à sélectionner le package recherché,
- dans le volet "Packages", sélectionner "Mettre à jour les packages",
- tapez dans R la commande suivante :
  `> library(mclust)`
  pour que R connaisse les fonctions de ce package.

Bibliographie

[1] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Trans. on PAMI*, 22 :719–725, 2000.

[2] G. Celeux and D. Diebolt. The SEM algorithm : a probabilistic teacher algorithm derived from the em algorithm for the mixture problem. *Computational Statistics Quarterly*, 2(1) :73–82, 1985.

[3] G. Celeux and G. Govaert. A Classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14 :315–332, 1992.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39 :1–38, 1977.

[5] Philippe Michel. *Polycopié de Statistique Exploratoire Multivariée, Année 1999-2000*. Ecole Nationale de la Statistique et de l'Analyse de l'Information.

[6] M. Nei. Genetic distance between populations. *Am. Nat.*, 106 :283–292, 1972.

[7] M. Nei. Estimation of average heterozygosity and genetic distance from a small number of individuals. *Genetics*, 89 :583–590, 1978.

[8] A. Peaucelle, H. Morin, J. Traas, and P Laufs. Plants expressing a mir164 resistant cuc2 gene reveal the importance of post-meristematic maintenance of phyllotaxy in Arabidopsis. *Development*, 134 :1045–1050, 2007.

[9] D. Reinhardt. Regulation of phyllotaxis. *Int. J. Dev. Biol.*, 49 :539–546, 2005.

[10] Y.C. Yao. Estimating the number of change-points via Schwarz criterion. *Stat. & Probab. Lett.*, 6 :181–189, 1988.