Apprentissage non supervisé
Chapitre 1 : Classification hiérarchique

Master "Machine Learning for Data Science", Paris V

Allou Samé
allou.same@ifsttar.fr

2017/2018

Plan

Données

- Ensemble de n individus décrits par p variables

$$E = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$$

avec $\mathbf{x}_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{ip})$

- Cet ensemble peut également être représenté sous la forme d'un tableau $\mathbf{X}$ de n lignes et p colonnes

$\mathbf{X} =$

	var 1	...	var j	...	var p
indiv 1	x_{11}	...	x_{1j}	...	x_{1p}
indiv i	x_{i1}	...	x_{ij}	...	x_{ip}
indiv n	x_{n1}	...	x_{nj}	...	x_{np}

Structures associées à la classification

Partition

L'ensemble $P = (P_1, \ldots, P_K)$, avec $P_k \subset E$, est une partition de E en K classes si :

(i) $\forall k \neq \ell \quad P_k \cap P_\ell = \emptyset$

(ii) La réunion des classes $P_1, \ldots, P_K$ est l'ensemble E

Représentation équivalente

$$\mathbf{z} = \begin{pmatrix} z_{11} & \cdots & z_{1K} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nK} \end{pmatrix} \quad \text{avec} \quad z_{ik} = \begin{cases} 1 \text{ si } \mathbf{x}_i \in P_k \\ 0 \text{ sinon.} \end{cases}$$

La somme des valeurs la i^{e} ligne vaut 1 (un élément appartient à une seule classe)

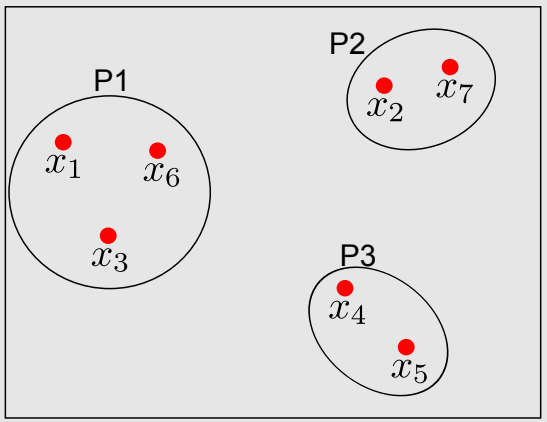
La somme des valeurs de la k^{e} colonne vaut n_k le nombre d'éléments de la classe P_k

Structures associées à la classification

Exemple

donnée	classe
$\mathbf{x}_1$	P_1
$\mathbf{x}_2$	P_2
$\mathbf{x}_3$	P_1
$\mathbf{x}_4$	P_3
$\mathbf{x}_5$	P_3
$\mathbf{x}_6$	P_1
$\mathbf{x}_7$	P_2

$$\Leftrightarrow \quad \mathbf{z} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$



Structures associées à la classification

Partition floue

Extension de la notion d'appartenance à une classe

Une partition floue est définie par une matrice de classification floue $\mathbf{C} = (c_{ik})_{i,k}$ vérifiant les propriétés suivantes :

(i) $\forall i, k, \quad c_{ik} \in [0; 1]$

(ii) $\forall k, \quad \sum_{i=1}^{n} c_{ik} > 0$

(iii) $\forall i, \quad \sum_{k=1}^{K} c_{ik} = 1$

Condition (i) : relachement de la contrainte de binarité
Condition (ii) : aucune classe ne doit être vide
Condition (iii) : appartenance totale

Structures associées à la classification

Hiérarchie

Un ensemble H de parties non vides de E est une hiérarchie si :

(i) L'ensemble E appartient à H

(ii) Toutes les parties formées d'un singleton appartiennent à H : $\forall i \quad \{\mathbf{x}_i\} \in H$

(iii) Deux éléments de H sont disjoints ou bien l'un contient l'autre

Exemple

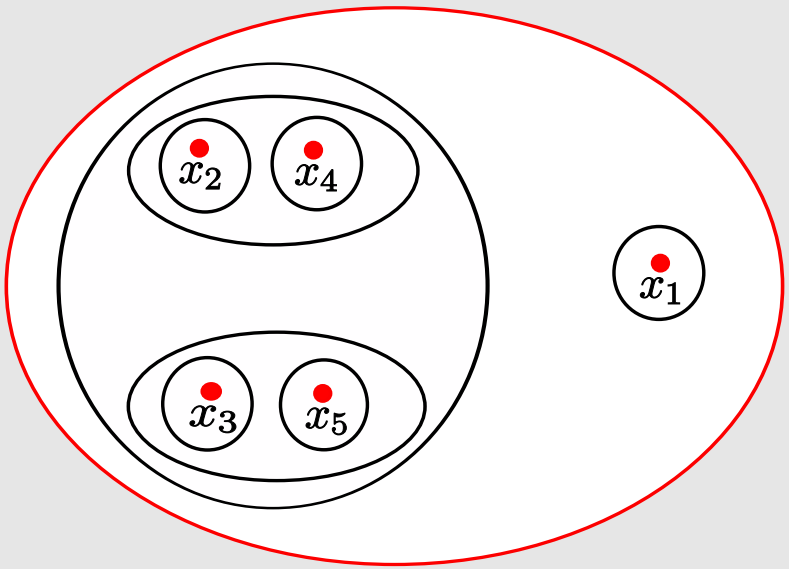
$$H = \Big\{ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_5\}, \{\mathbf{x}_2, \mathbf{x}_4\}, \{\mathbf{x}_3, \mathbf{x}_5\}, \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}, \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\} \Big\}$$

est une hiérarchie définie sur $E = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5\}$

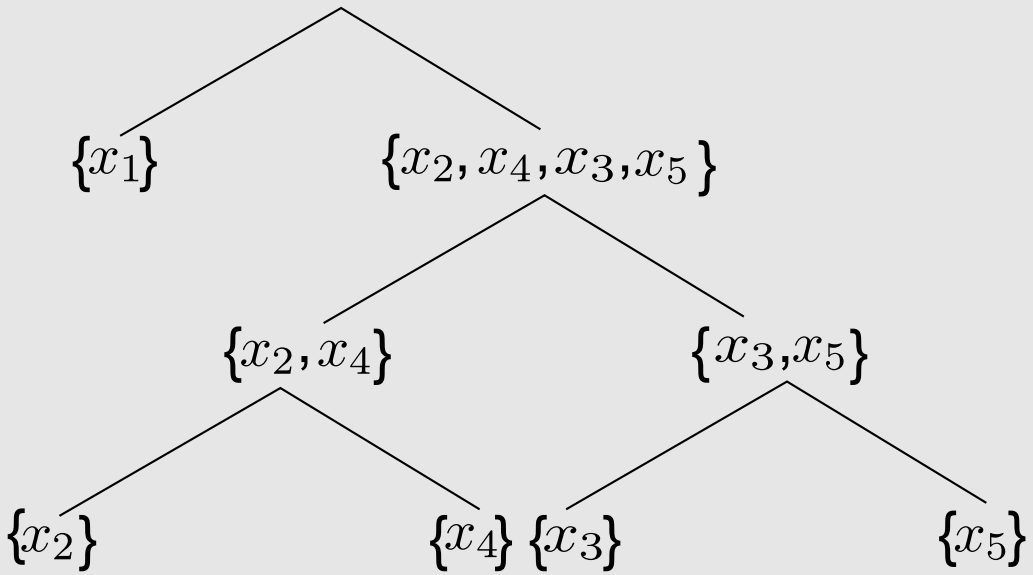
Structures associées à la classification

Représentation graphique associée à une hiérarchie

Représentation ensembliste



Représentation par arbre



Structures associées à la classification

Remarque

- Les représentations précédentes sont rarement utilisées
- On préfère leur adjoindre un indice, pour rendre la représentation plus lisible

Indice sur une hiérarchie H

Fonction i de H dans $\mathbb{R}^+$ vérifiant :

(i) $h \subset h'$ et $h \neq h' \Rightarrow i(h) < i(h')$
(i est une fonction strictement croissante)

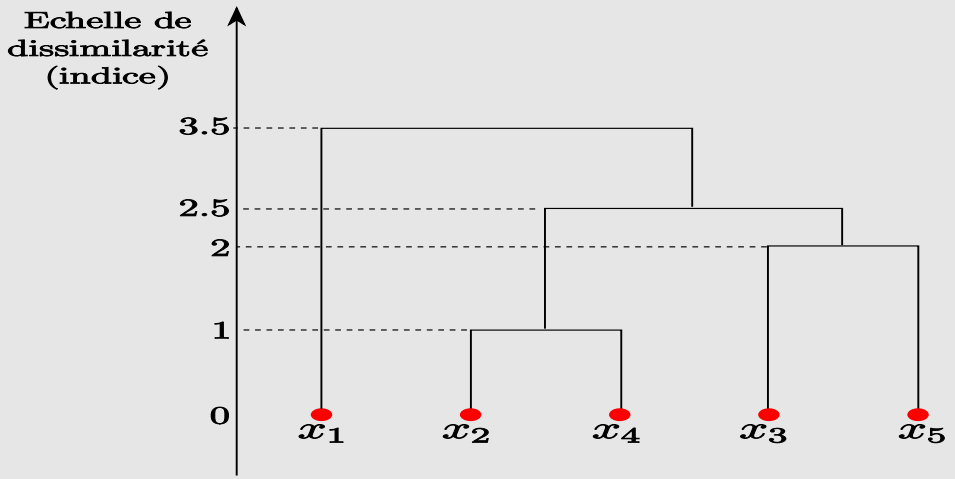
(ii) $i(\{\mathbf{x}_i\}) = 0 \quad \forall \mathbf{x}_i \in E$

On dit que (H, i) est une hiérarchie indicée sur E.

Structures associées à la classification

Représentation graphique d'une hiérarchie

- On représente généralement une hiérarchie à l'aide d'un dendrogramme, arbre qui montre comment les données sont regroupées
- On associe aux différents niveaux de cet arbre une échelle de mesure (indice)
- La coupe de l'arbre à une certain niveau (indice) permet d'obtenir une partition



Ultramétrique et hiérarchie indicée

Proposition

A toute ultramétrique on peut associer une hiérarchie indicée et inversement.

Preuve

Soit δ une ultramétrique sur E.
Notons $\mathcal{E}_\delta$ l'ensemble des valeurs prises par δ.
Considérons la relation d'équivalence définie par :
$\mathbf{x}\ \mathcal{R}_\varepsilon\ \mathbf{y} \Leftrightarrow \delta(\mathbf{x},\mathbf{y}) \leq \varepsilon \quad \forall \mathbf{x},\mathbf{y} \in E$.
Notons $\mathcal{C}_\varepsilon$ l'ensemble des classes d'équivalence de $\mathcal{R}_\varepsilon$.
Posons $H = \bigcup_{\varepsilon \in \mathcal{E}_\delta} \mathcal{C}_\varepsilon$ et $i(h) = \max_{\mathbf{x},\mathbf{y} \in h} \delta(\mathbf{x},\mathbf{y})$ (diametre de h).
Alors on peut montrer que (H, i) est bien une hiérarchie indicée.

Réciproquement, si (H, i) est une hiérarchie indicée sur E alors on peut montrer que l'application définie par
$\delta(\mathbf{x},\mathbf{y}) = \min_{h \in H}\{i(h) \mid \mathbf{x},\mathbf{y} \in h\}$ est une ultramétrique.

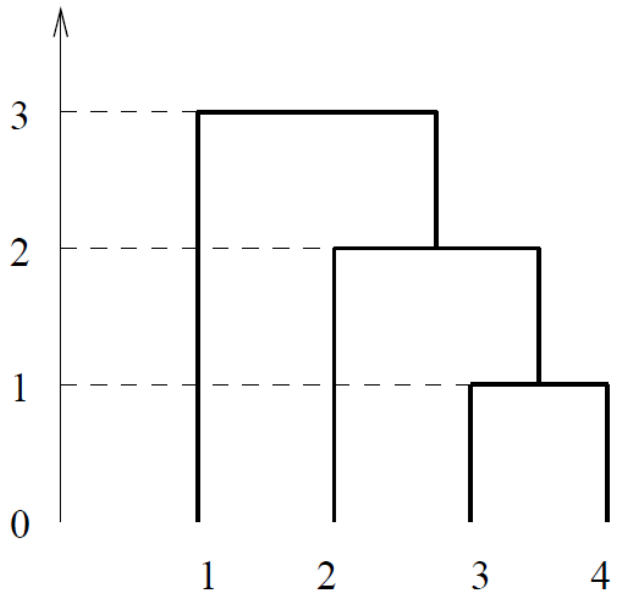
Illustration de la propriété

$$H = \Big\{ \{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \{\mathbf{x}_3\}, \{\mathbf{x}_4\}, \{\mathbf{x}_3, \mathbf{x}_4\}, \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\} \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\} \Big\}$$

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$
$\mathbf{x}_1$	0			
$\mathbf{x}_2$	3	0		
$\mathbf{x}_3$	3	2	0	
$\mathbf{x}_4$	3	2	1	0

$\Longleftrightarrow$

Indices : $0, 0, 0, 0, 1, 2, 3.$



Démarche habituelle des méthodes de classification

- Optimisation d'un critère numérique qui mesure l'homogénéité d'une partition
  - Exemple : minimisation de l'inertie intra-classe

$$I_W = \frac{1}{n} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in P_k} d^2(\mathbf{x}_i, g_k)$$

    où g_k est la moyenne empirique de la classe P_k
- Démarche algorithmique : construction itérative d'une « bonne » solution (proche de la solution optimale)
- Deux grandes familles de méthodes :
  - Méthode de classification hiérarchique
  - Méthode par partitionnement

Aspects combinatoires liés à la classification

- Nombre total de partitions d'un ensemble de n éléments en K classes (nombre de Stirling de 2^{e} espèce) :

$$P_{n,K} = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} C_K^k \, k^n$$

Quand n devient grand, ce nombre devient trop élevé !

- Nombre total de partitions d'un ensemble de n éléments (nombre de Bell) :

$$P_n = \sum_{K=1}^{n} P_{n,K} = \sum_{k=0}^{n-1} C_n^k P_k$$

Aspects combinatoires liés à la classification

Exemple de calcul de $P_{n,K}$ et P_n

n \ K	1	2	3	4	5	6	7	8	P_n
1	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	0	0	0	2
3	1	3	1	0	0	0	0	0	5
4	1	7	6	1	0	0	0	0	15
5	1	15	25	10	1	0	0	0	52
6	1	31	90	65	15	1	0	0	203
7	1	63	301	350	140	21	1	0	877
8	1	127	966	1701	1050	266	28	1	4140

$$P_{100,5} \approx 10^{67}$$

Rechercher la meilleure partition en examinant toutes les partitions possibles est une tâche quasiment insurmontable

Aspects combinatoires liés à la classification

n	P_n
1	1
2	2
3	5
4	15
5	52
6	203
7	877
8	4140
9	21147
10	115975
21	474869816156751
22	4506715738447323
23	44152005855084346
24	445958869294805289
25	4638590332229999353
26	49631246523618756274
27	545717047936059989389
28	6160539404599934652455
29	713398019388602751911 72
30	846749014511809332450147
40	1574505883912049312893243447025310 67
41	2351152507740617628200694077243788988
42	3574254919887261729135350865566266425 67
43	5529501187971654843217146932807377673 85
44	8701963427387055089023600531855797148876
45	13925850526626366960234705399365407969341 5
46	22654182193344940029284844447053922761583 55
47	37450059502461511196505342096431510120174682
48	628919796303118415420210454071849537746015761
49	1072613715457335840034221551859000263391724728 1
50	1857242687710782704382577671819089174992218527 70

Classification hiérarchique

Objectif

Construire une hiérarchie indicée (H, i) à partir d'une matrice de dissimilarités

Types d'approches

Il existe deux types d'approches :

- La classification ascendante : partir de la partition où chaque classe est un singleton puis fusionner successivement les classes qui se ressemblent jusqu'à obtenir une seule classe (l'ensemble E)
- La classification descendante : diviser l'ensemble E en classes puis recommencer sur chacune des classes jusqu'à obtenir des singletons

La plus utilisée en pratique est la classification ascendante qui donne de meilleurs résultats.

Classification Ascendante Hiérarchique (CAH)

Principe

- En partant de la partition la plus élémentaire (1 singleton par classe), fusionner successivement les classes jusqu'à obtenir 1 classe
- Les regroupements successifs sont représentés sur le dendrogramme

Algorithme CAH

1. ***Initialisation***
   - Former les classes initiales : singletons $\{\mathbf{x}_1\}, \{\mathbf{x}_2\}, \ldots, \{\mathbf{x}_n\}$
   - Calculer la matrice des distances entre singletons
2. ***Tant que*** le nombre de classes est > 1
   - Regrouper les deux classes les plus proches
   - Mettre à jour le tableau des distances

Hiérarchie indicée issue de la CAH

Hiérarchie

L'ensemble des classes définies au cours de l'algorithme forment une hiérarchie sur E.

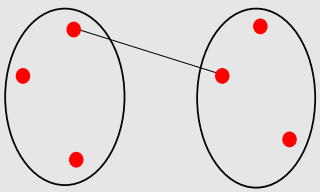
Indice

- Pour les classes du bas de la hiérarchie (singletons), on associe un indice nul.
- Pour toute autre classe, on associe l'indice égal à la distance entre les deux classes fusionnées pour obtenir cette dernière classe.

Agrégation de classes

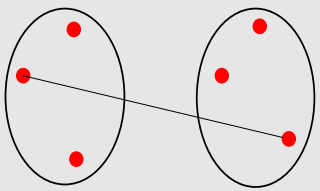
Distance minimale : D_{min}

$$D_{min}(A,B) = \min_{\mathbf{x}_i \in A \,;\, \mathbf{x}_{i'} \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$



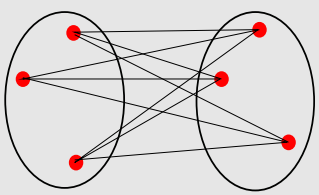
Distance maximale : D_{max}

$$D_{max}(A,B) = \max_{\mathbf{x}_i \in A \,;\, \mathbf{x}_{i'} \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$



Distance moyenne : D_{moy}

$$D_{moy}(A,B) = \frac{\sum_{\mathbf{x}_i \in A} \sum_{\mathbf{x}_{i'} \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})}{n_A \cdot n_B}$$



Formules récurrentes de Lance et Williams

Formules récurrentes (calcul plus rapide)

- $D_{min}(A, B \cup C) = \min(D_{min}(A, B), D_{min}(A, C))$
- $D_{max}(A, B \cup C) = \max(D_{max}(A, B), D_{max}(A, C))$
- $D_{moy}(A, B \cup C) = \frac{n_B D_{moy}(A,B) + n_C D_{moy}(A,C)}{n_B + n_C}$

Optimalité de la CAH

On peut montrer que la hiérarchie issue de l'algorithme CAH-Dmin est équivalente à l'ultramétrique δ qui minimise le critère

$$\mathcal{C}(\delta) = \sum_{\mathbf{x},\mathbf{y} \in E} \Big(d(\mathbf{x},\mathbf{y}) - \delta(\mathbf{x},\mathbf{y}) \Big)^2$$

sous la contrainte $\delta < d$.

Illustration CAH - critère d'agrégation D_{min}

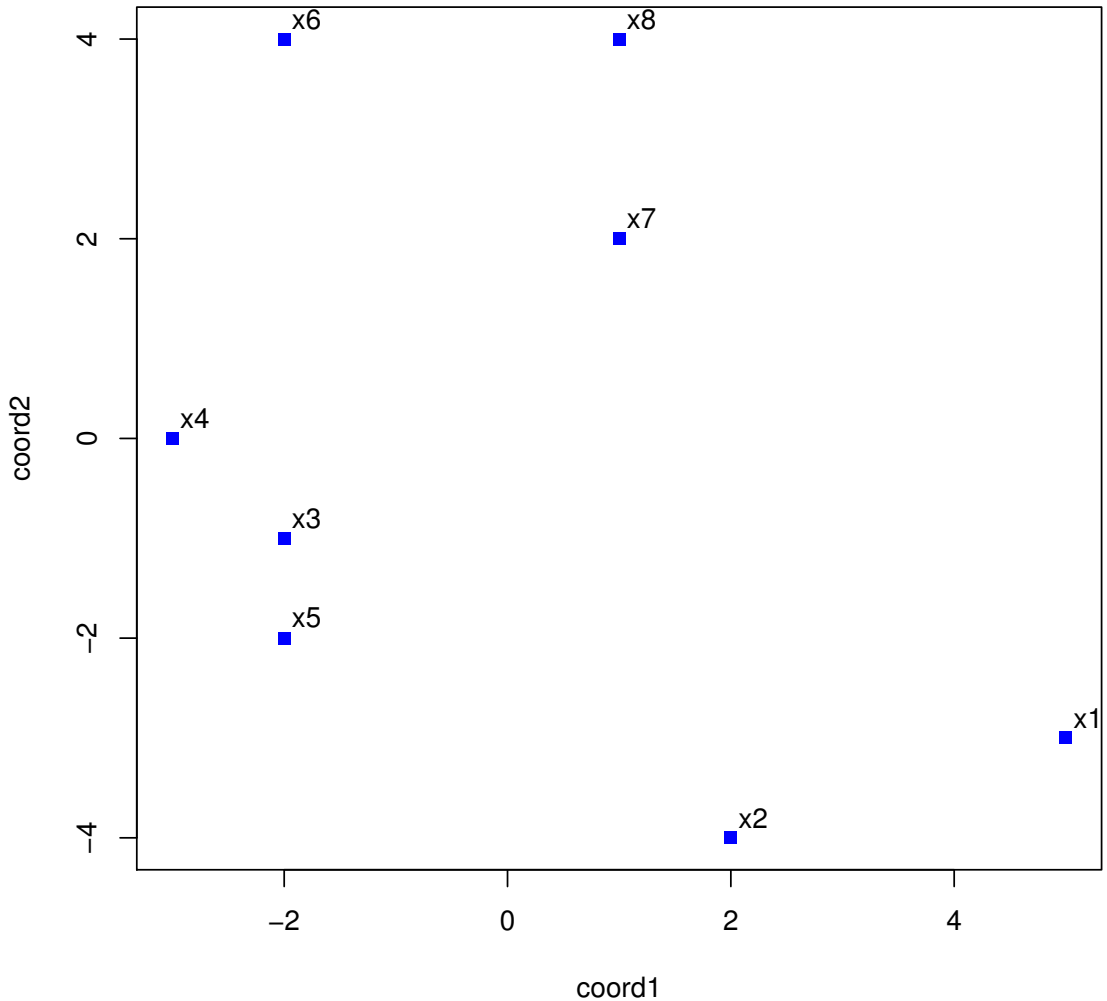


Illustration CAH - critère d'agrégation D_{min}

Matrice initiale des distances entre singletons

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	0							
x_2	3.16	0						
x_3	7.28	5	0					
x_4	8.54	6.40	1.41	0				
x_5	7.07	4.47	1	2.24	0			
x_6	9.90	8.94	5	4.12	6	0		
x_7	6.40	6.08	4.24	4.47	5	3.61	0	
x_8	8.06	8.06	5.83	5.66	6.71	3	2	0

Illustration CAH - critère d'agrégation D_{min} (0/7)

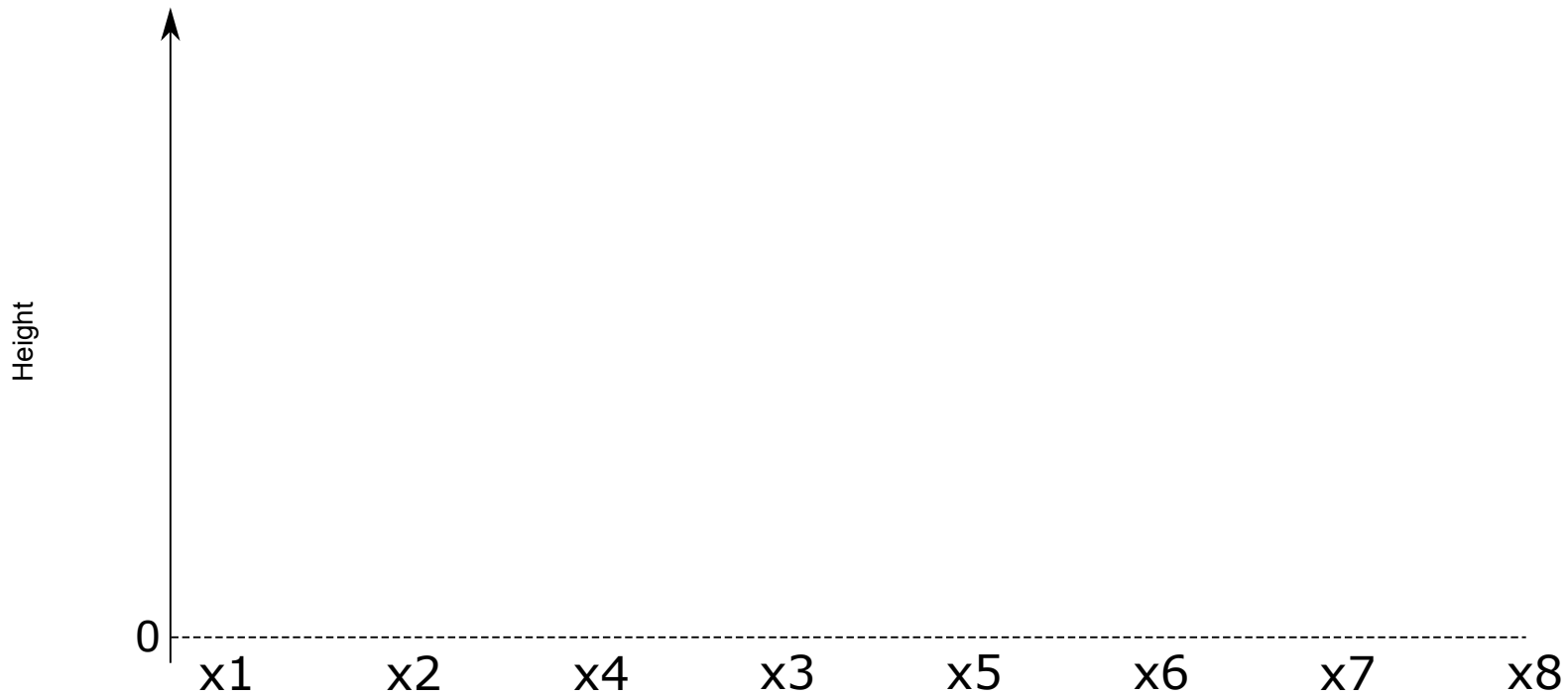


Illustration CAH - critère d'agrégation D_{min} (1/7)

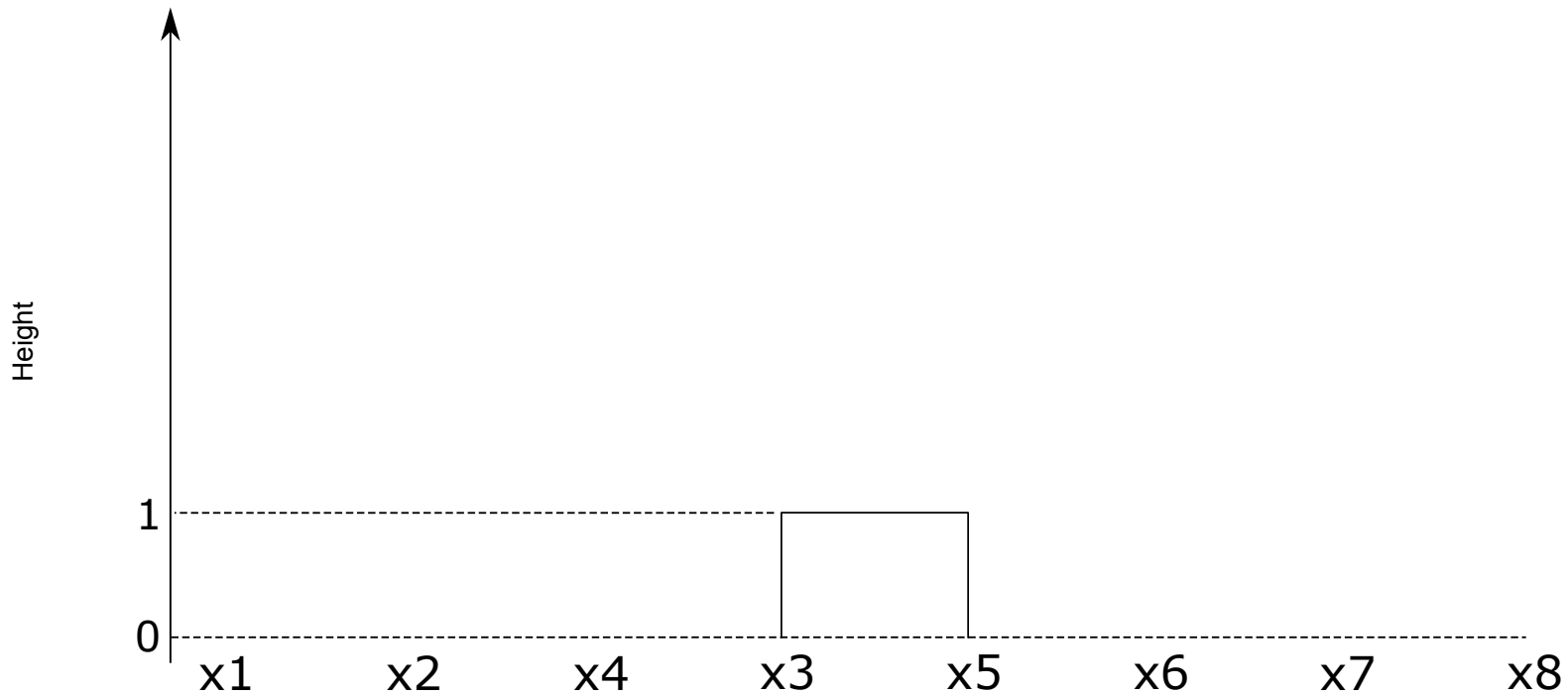


Illustration CAH - critère d'agrégation D_{min} (2/7)

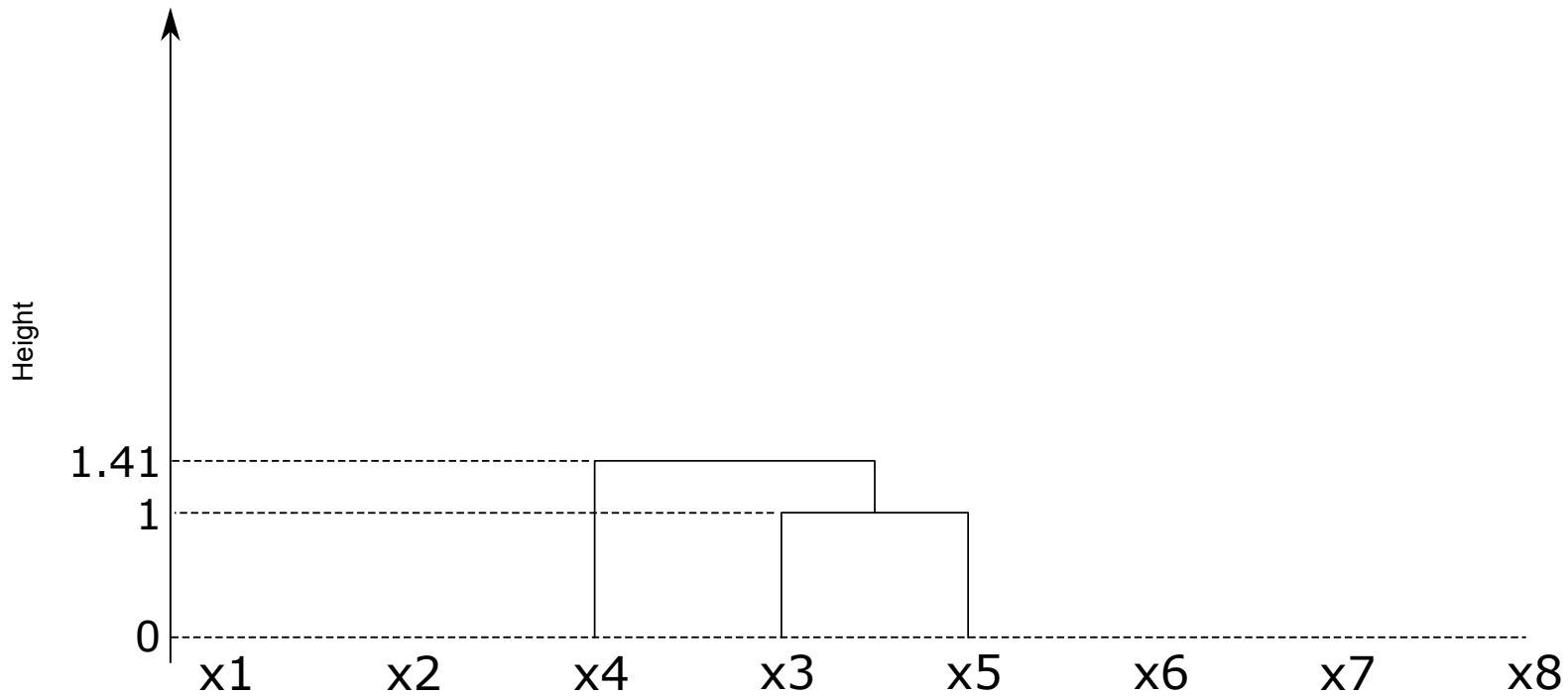


Illustration CAH - critère d'agrégation D_{min} (3/7)

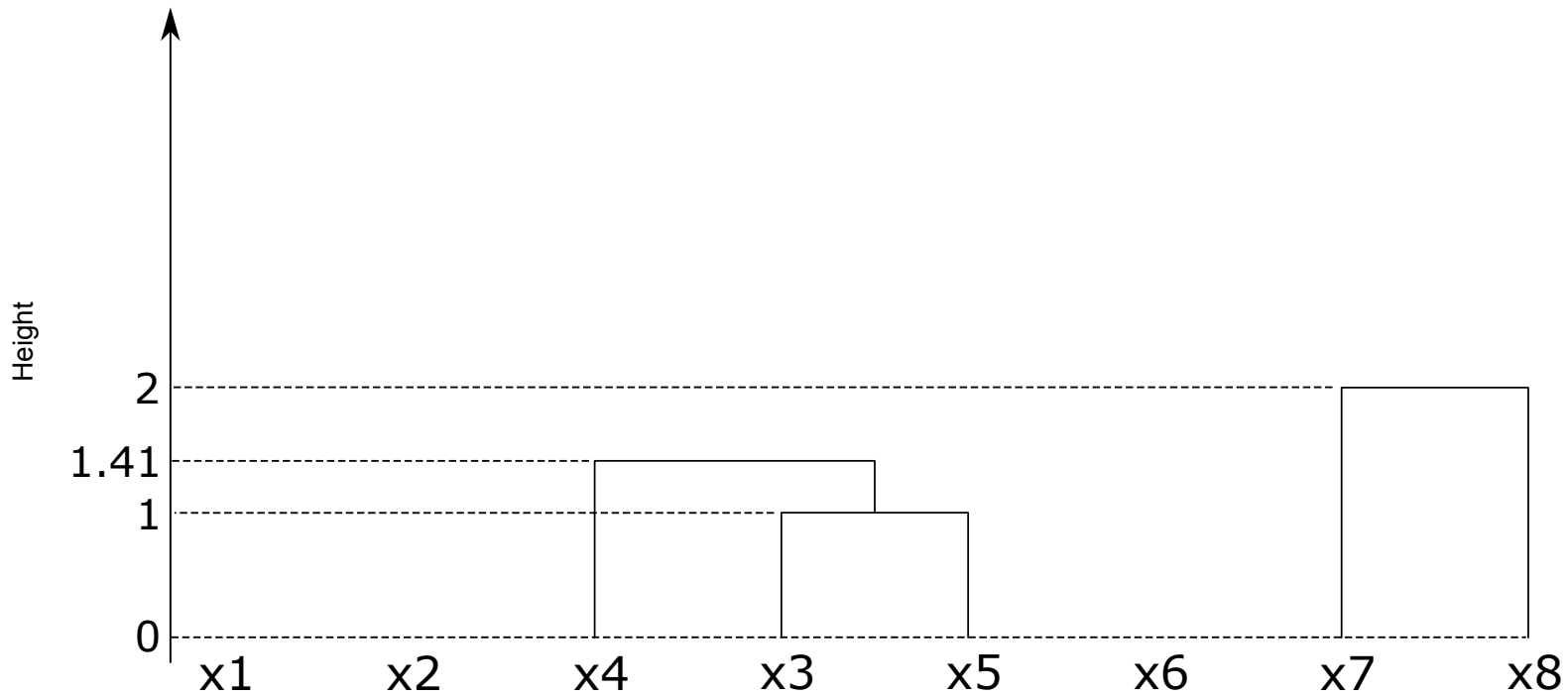


Illustration CAH - critère d'agrégation D_{min} (4/7)

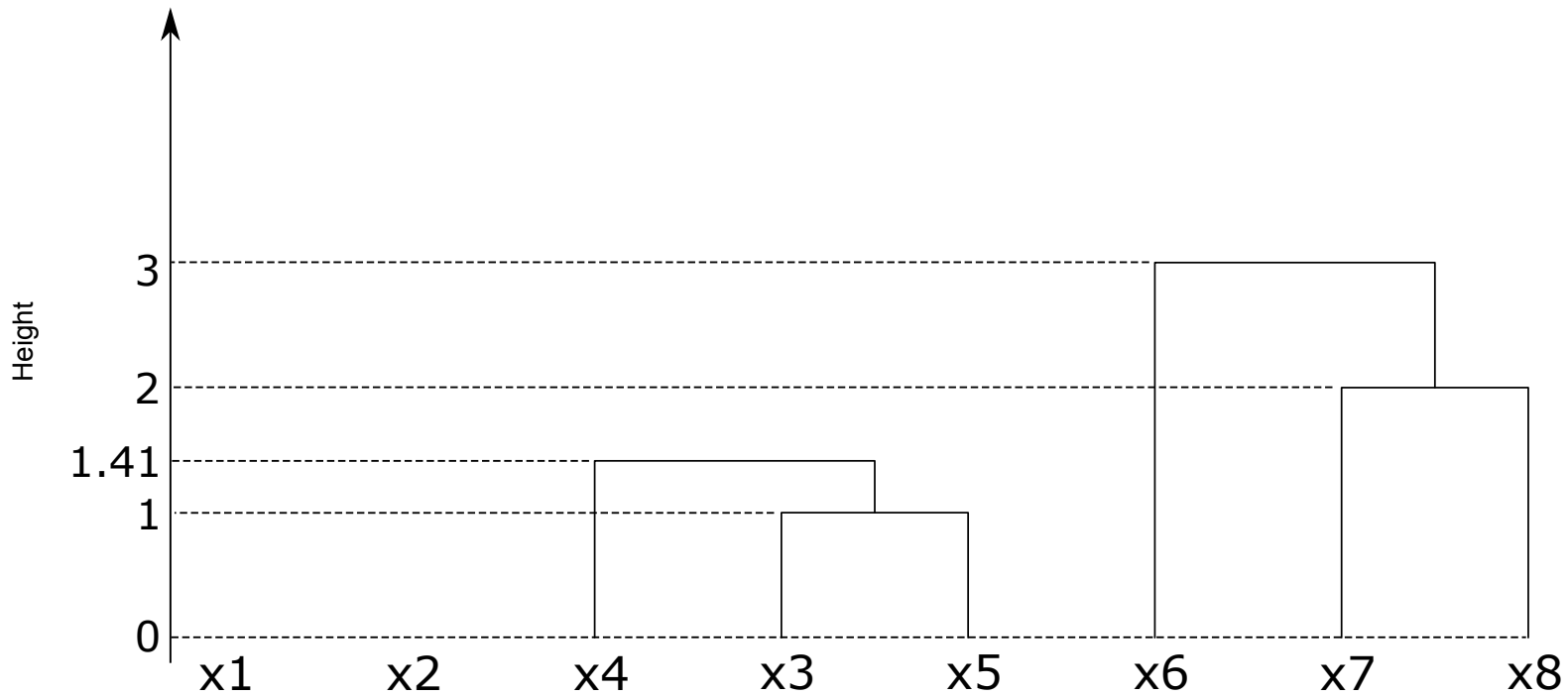


Illustration CAH - critère d'agrégation D_{min} (5/7)

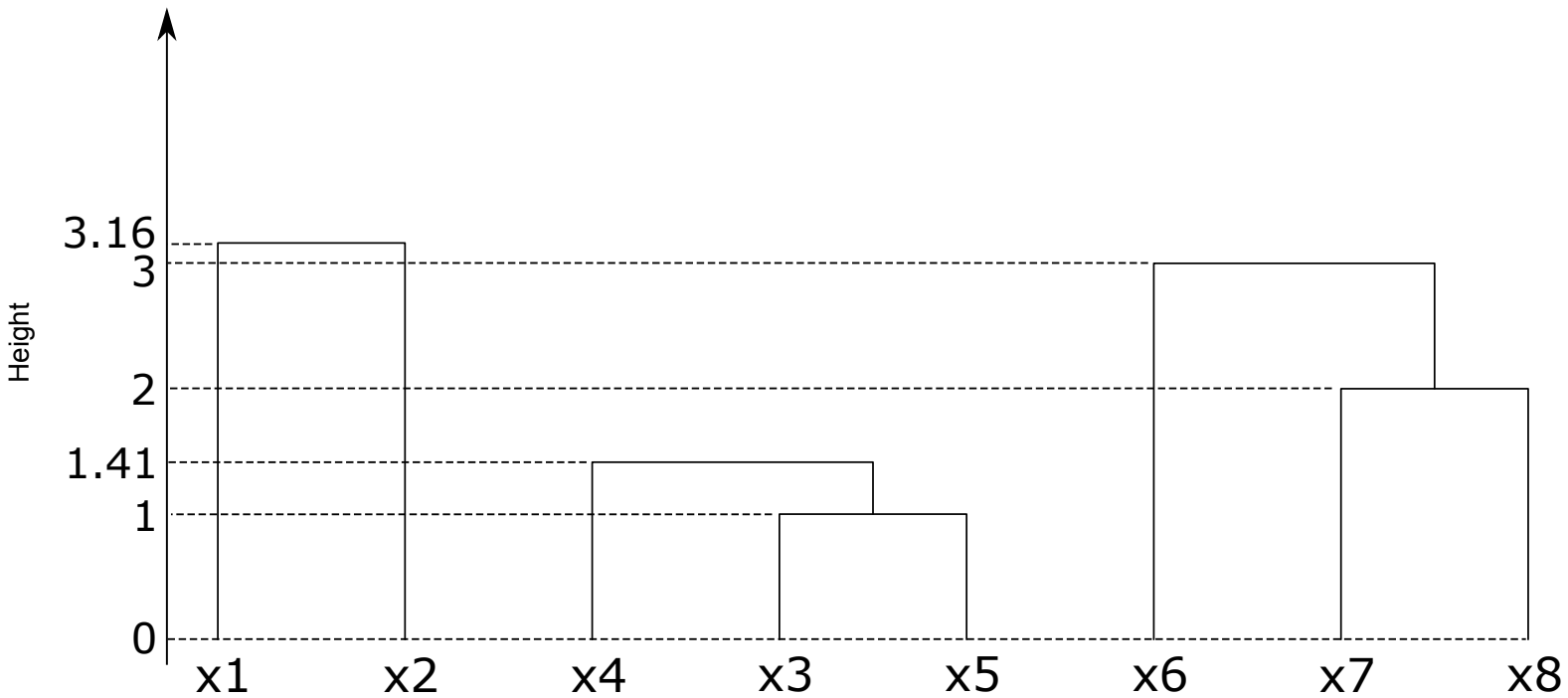


Illustration CAH - critère d'agrégation D_{min} (6/7)

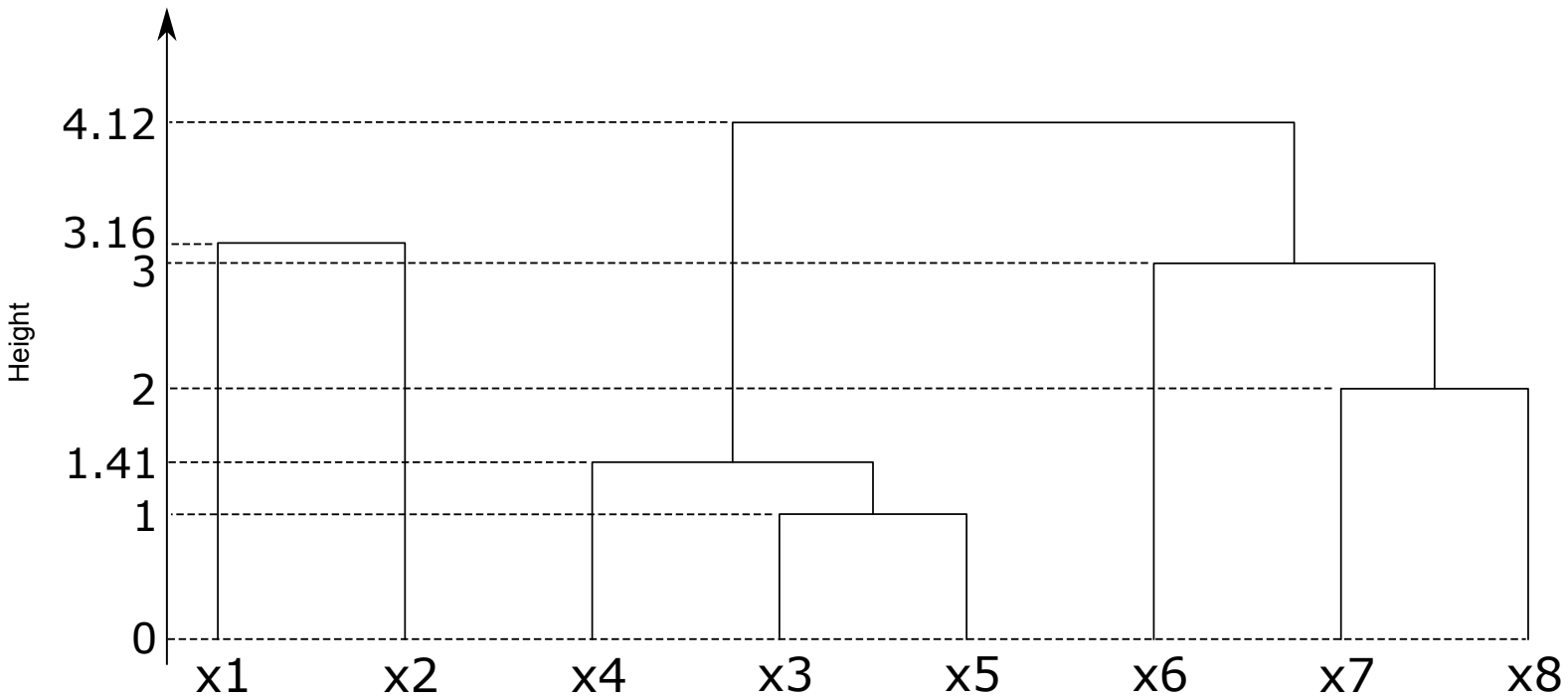
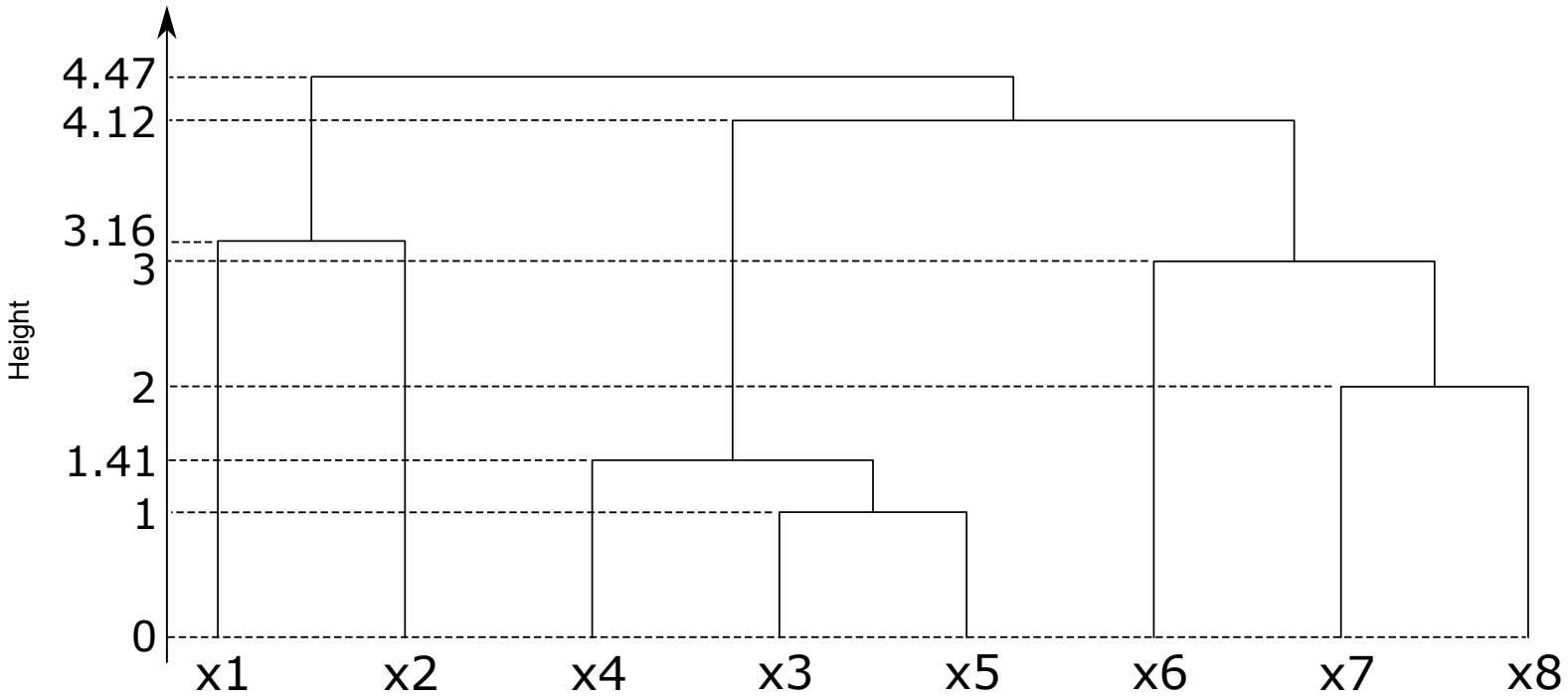
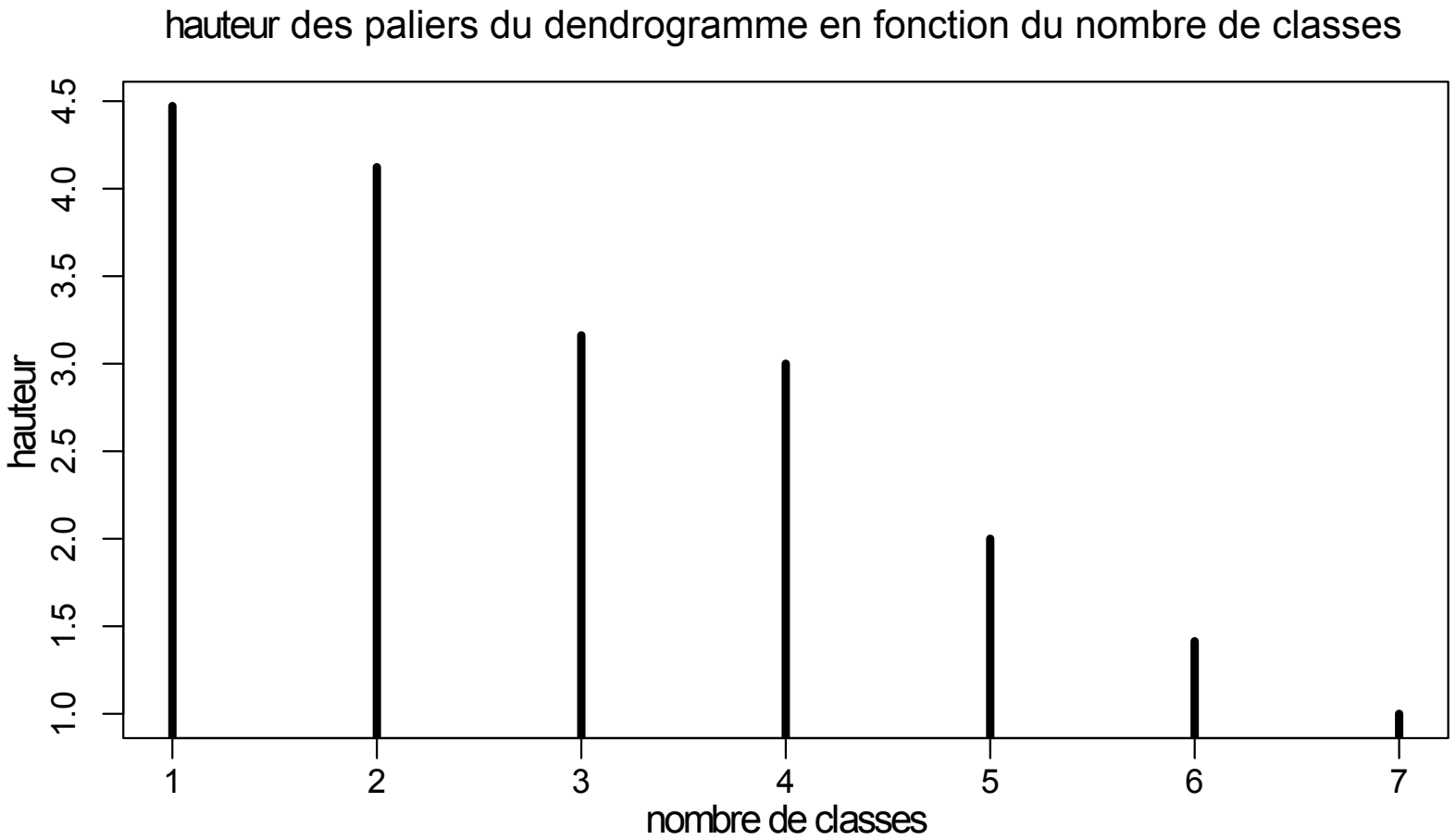


Illustration CAH - critère d'agrégation D_{min} (7/7)



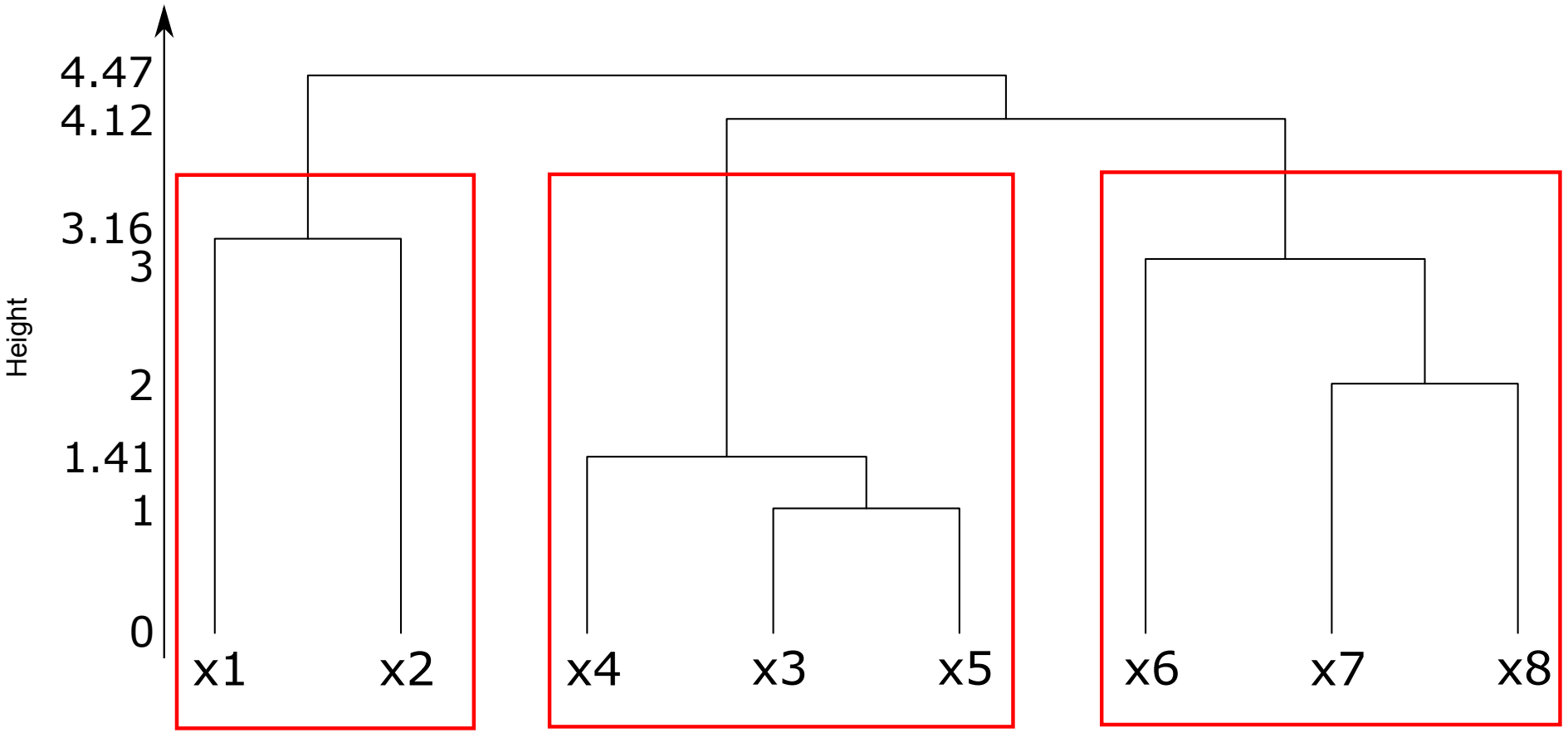
Partition issue d'une hiérarchie

Il est possible de déterminer une partition en effectuant une coupe au premier saut d'indice jugé significatif (les sous arbres obtenus constituent les classes)



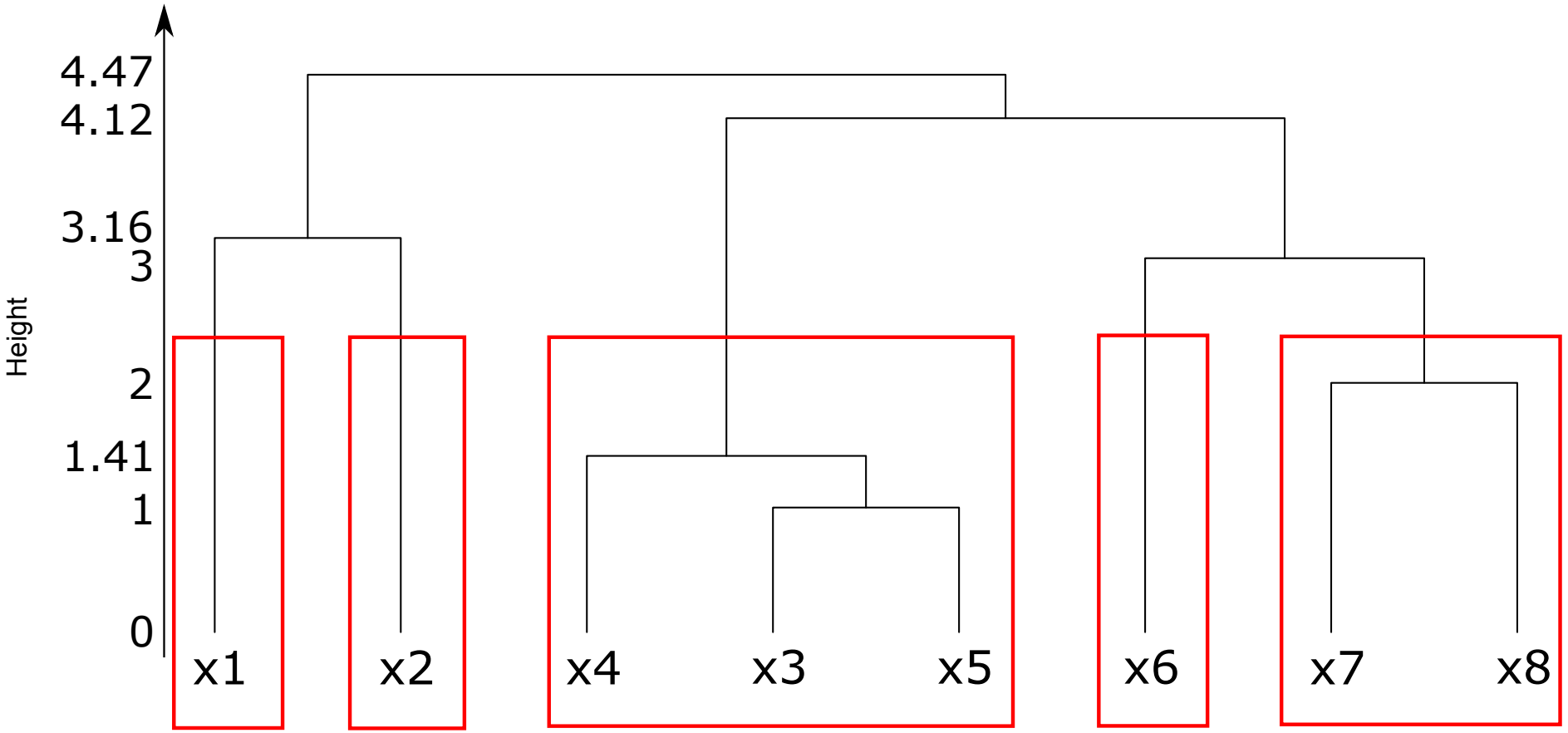
Partition issue d'une hiérarchie

Partition en 3 classes (avec critère d'agrégation du lien minimum)



Partition issue d'une hiérarchie

Partition en 5 classes (avec critère d'agrégation du lien minimum)



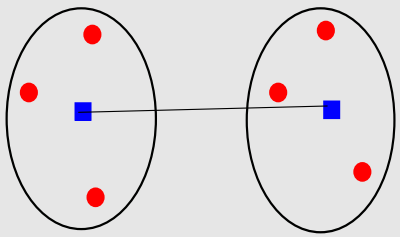
Méthode de Ward

Définition

Algorithme CAH avec le critère d'agrégation :

$$D(A,B) = \frac{n_A \cdot n_B}{n_A + n_B} d^2(g_A, g_B)$$

en supposant que d est la distance euclidienne,
et g_A et g_B sont les centres de gravité respectifs de A et B.



Formule de récurrence

$$D(A, B \cup C) = \frac{(n_A + n_B)D(A,B) + (n_A + n_C)D(A,C) - n_A D(B,C)}{n_A + n_B + n_C}$$

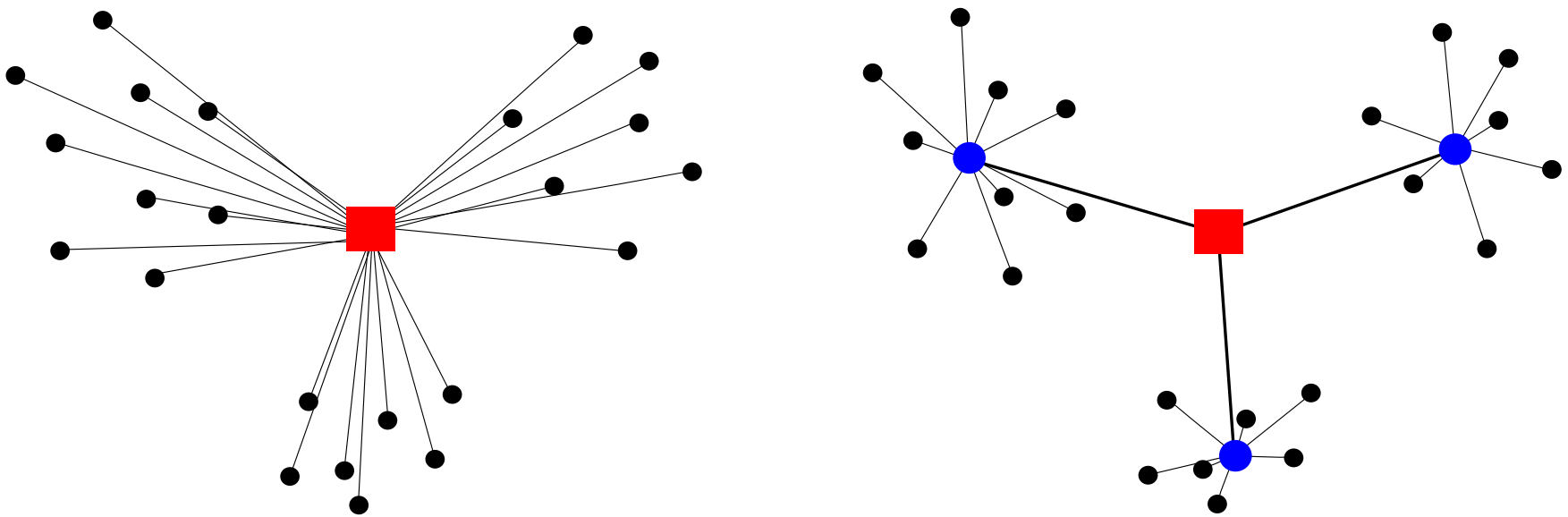
Notion d'inertie

$$I = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{g}\|^2 \qquad \text{Inertie totale}$$

$$I_B = \frac{1}{n}\sum_{k=1}^{K} n_k\|\mathbf{g}_k - \mathbf{g}\|^2 \qquad \text{Inertie inter-classe}$$

$$I_W = \frac{1}{n}\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in P_k}\|\mathbf{x}_i - \mathbf{g}_k\|^2 \qquad \text{Inertie intra-classe}$$

I = I_B + I_W



Méthode de Ward et critère d'inertie

Proposition (optimisation locale) : à chaque itération, la fusion de deux classes par le critère de Ward augmente le moins possible l'inertie intra-classe.

Preuve : soit $P = (P_1, \ldots, P_K)$ une partition et P' la partition obtenue à partir de P en fusionnant les classes P_k et P_ℓ. On peut alors montrer que

$$I_W(P') - I_W(P) = \frac{n_k n_\ell}{n_k + n_\ell} d^2(g_k, g_\ell) = D_{ward}(P_k, P_\ell)$$

où g_k et g_ℓ sont les moyennes des classes P_k et P_ℓ.

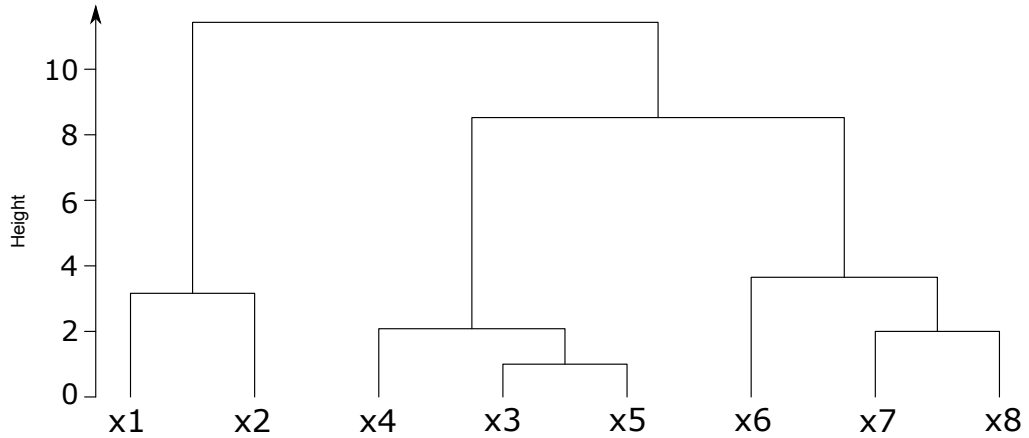
Méthode de Ward

Comportement de l'inertie au cours de l'algorithme

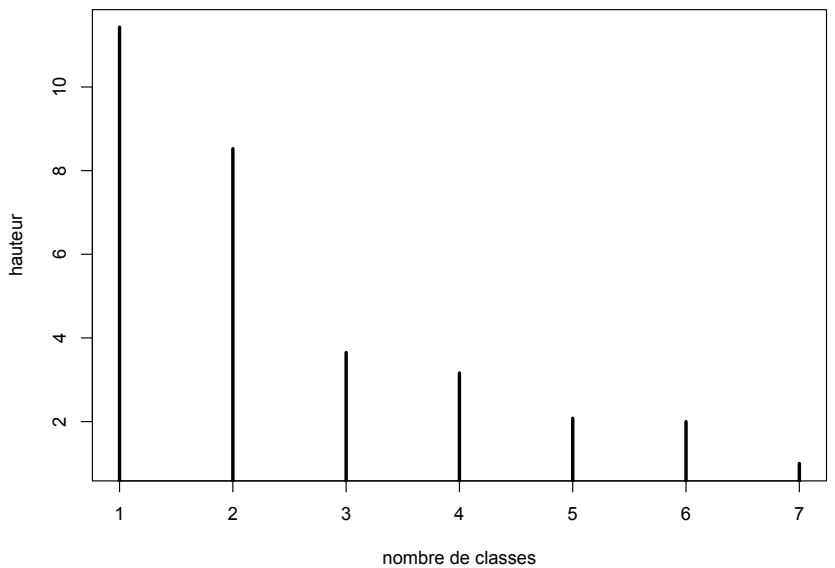
- A l'étape initiale, $I_W = 0$ et $I_B = I =$ inertie totale.
- A l'étape finale, $I_W = I$ et $I_B = 0$.
- Au fur et à mesure que l'on effectue les regroupements, I_W augmente et I_B diminue.

CAH avec le critère d'agrégation de Ward

Dendrogramme

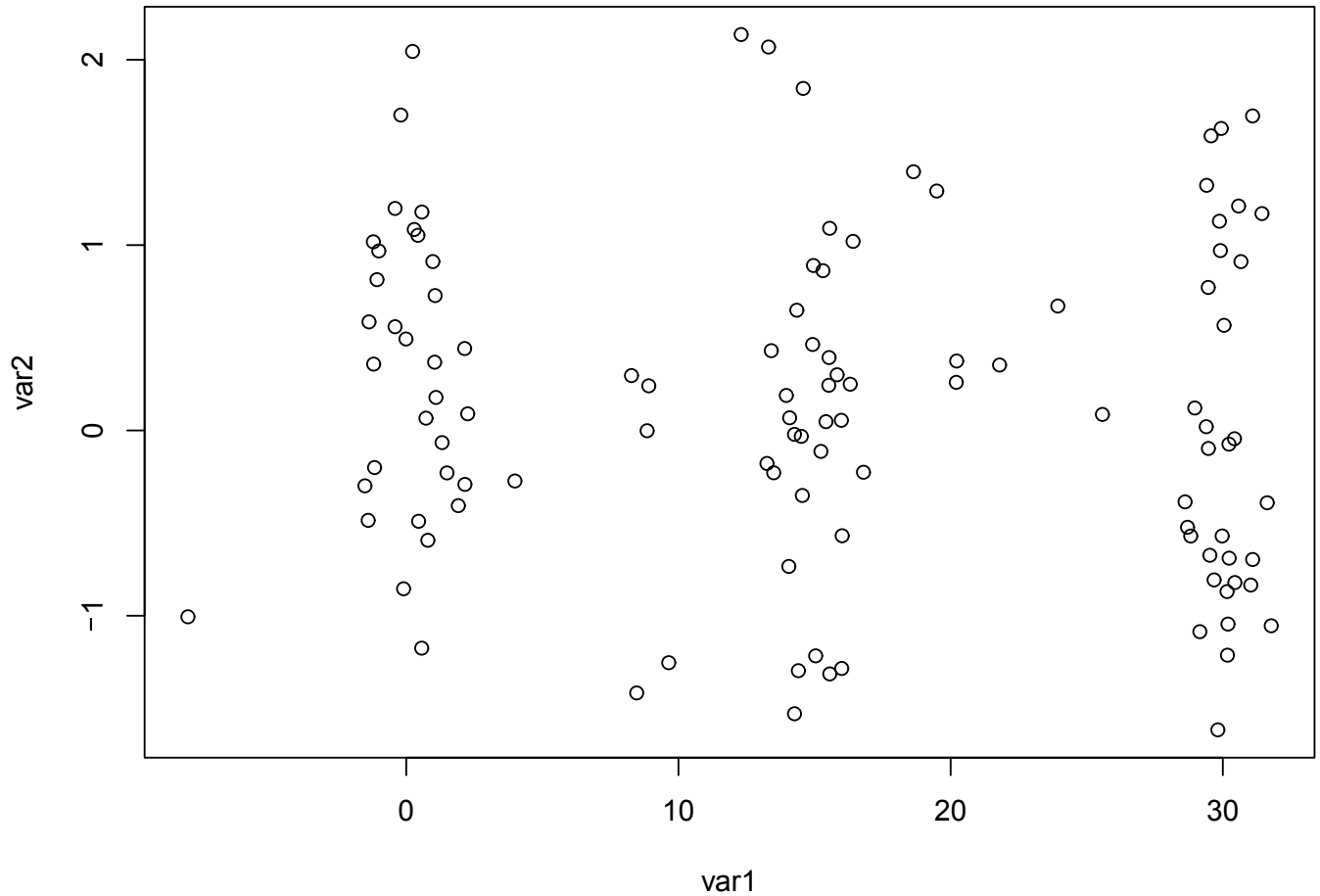


Paliers décroissants du dendrogramme



CAH sur des données simulées

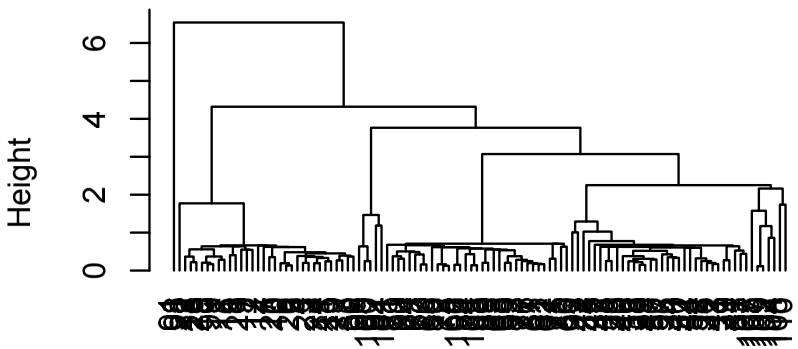
Données : trois classes gaussiennes légèrement bruitées



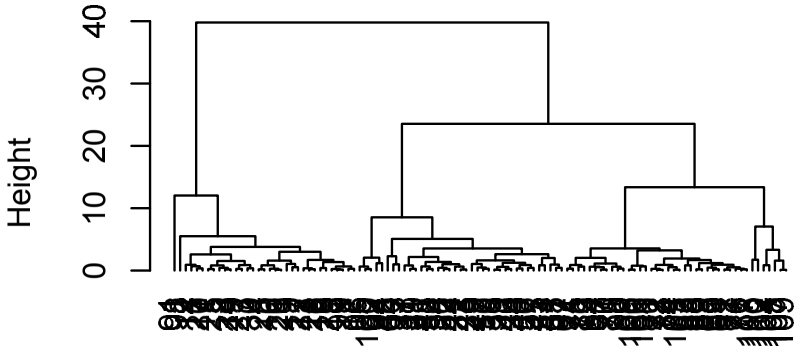
CAH sur des données simulées

Dendrogrammes

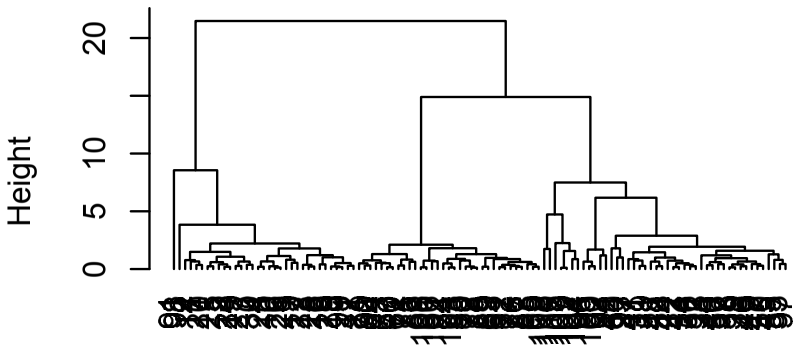
lien minimal



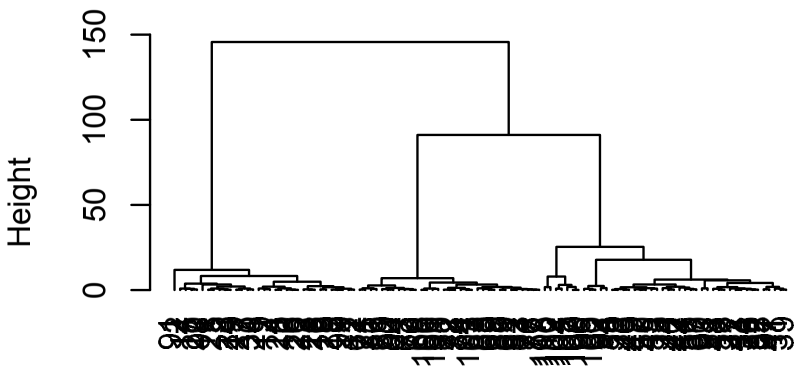
lien maximal



lien moyen



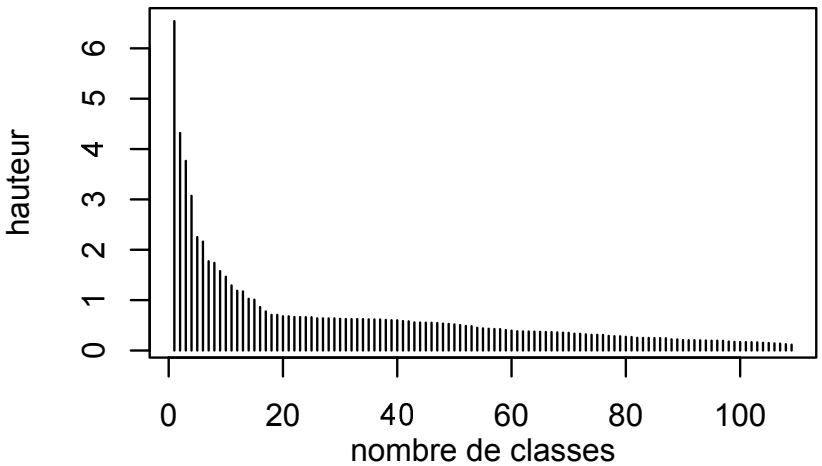
lien de Ward



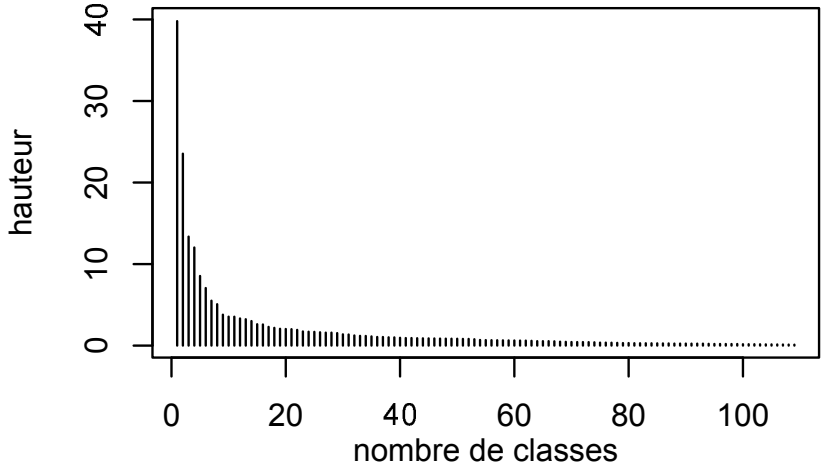
CAH sur des données simulées

Paliers décroissants des dendrogrammes

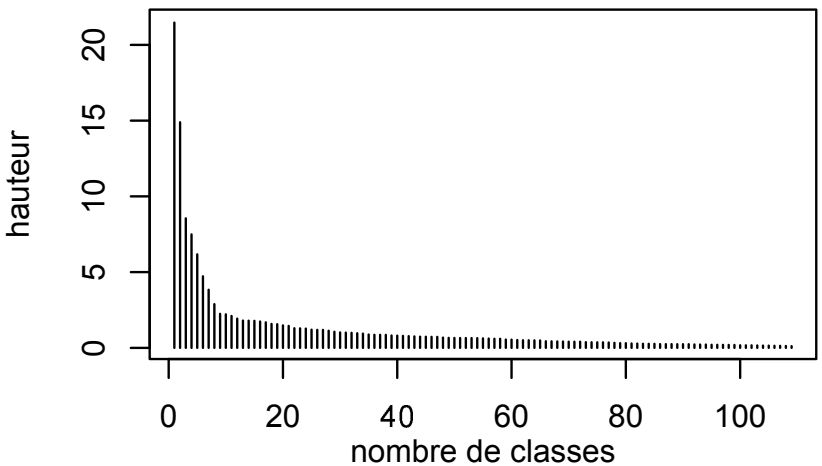
lien minimal



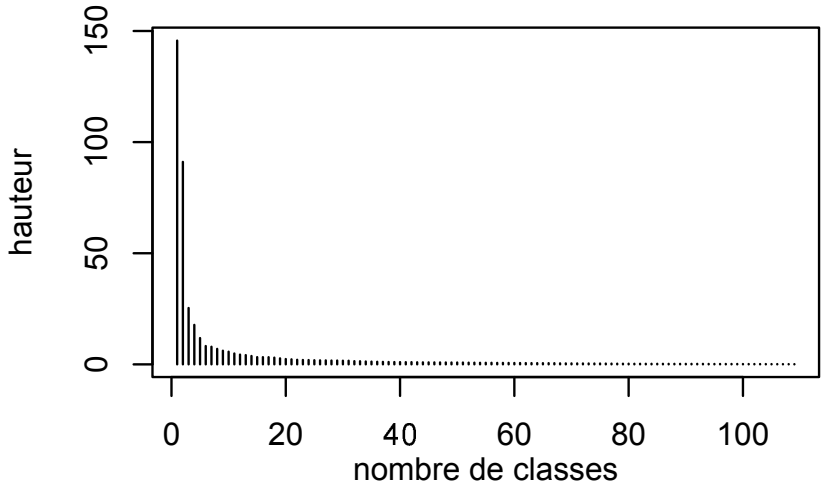
lien maximal



lien moyen

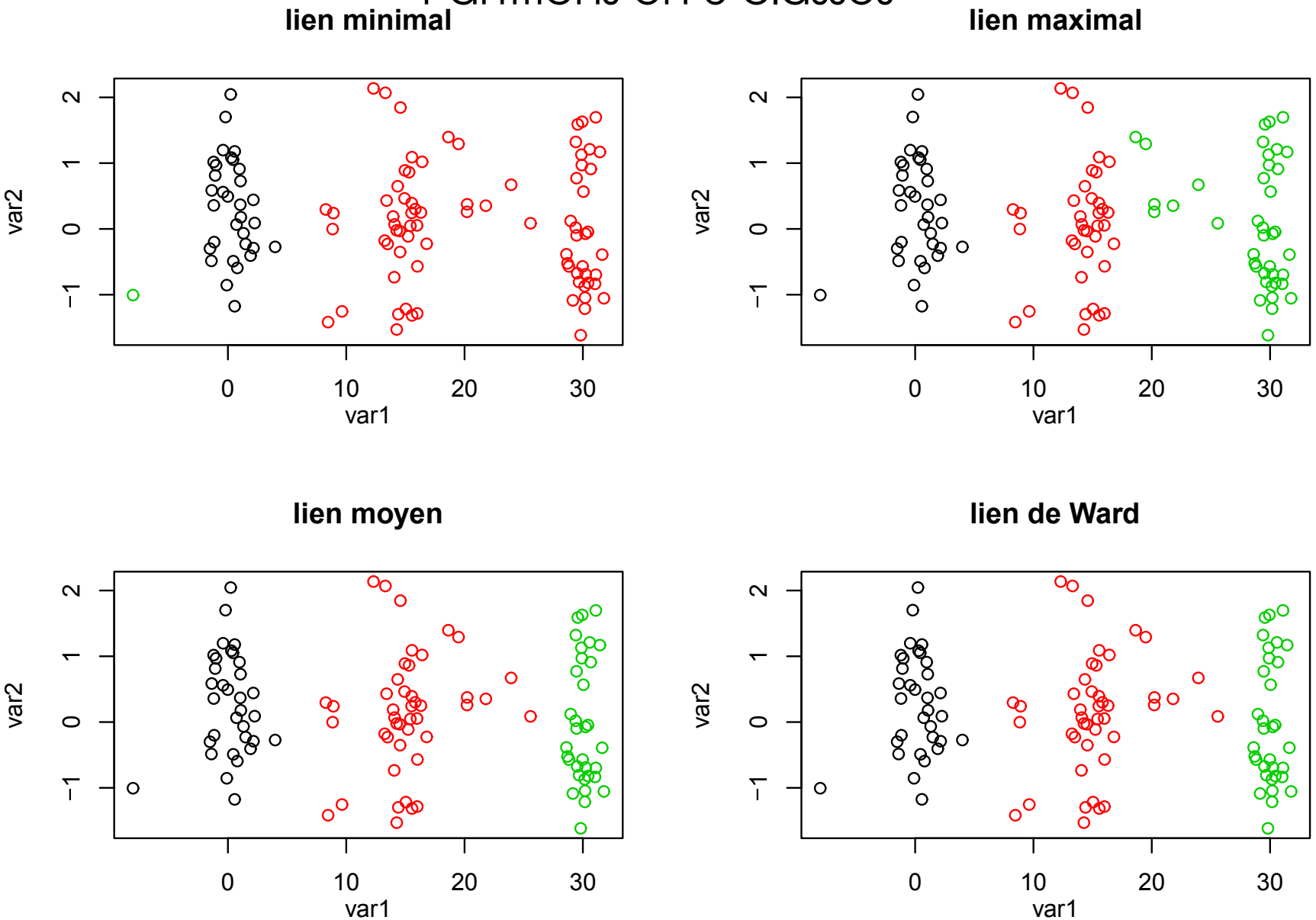


lien de Ward



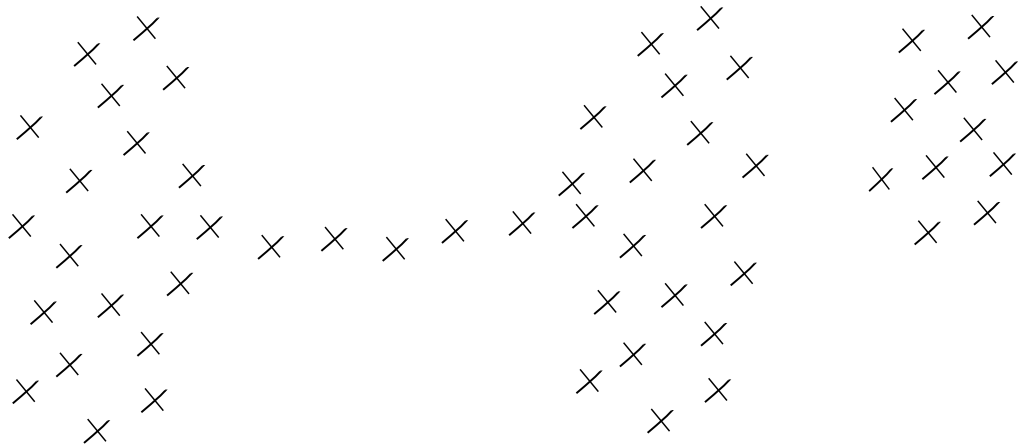
CAH sur des données simulées

Partitions en 3 classes



Remarque sur les critères d'agrégation

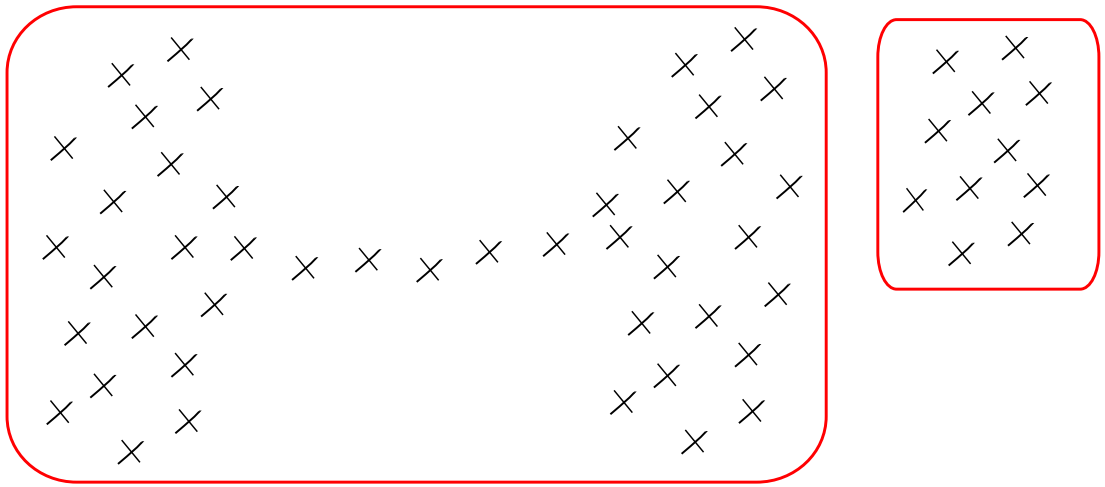
- Lien minimal : classes pouvant être déséquilibrées en volume et sensibles au bruit (« l'effet de chaîne »)



- Lien maximal : classes plus équilibrées en volume mais pouvant être très proches les unes des autres
- Lien moyen : situation intermédiaire entre le lien minimal et le lien maximal
- Ward : optimisation de l'inertie intra-classe à chaque itération

Remarque sur les critères d'agrégation

- Lien minimal : classes pouvant être déséquilibrées en volume et sensibles au bruit (« l'effet de chaîne »)



- Lien maximal : classes plus équilibrées en volume mais pouvant être très proches les unes des autres
- Lien moyen : situation intermédiaire entre le lien minimal et le lien maximal
- Ward : optimisation de l'inertie intra-classe à chaque itération

Récapitulatif sur la CAH

- La CAH nécessite de choisir une distance ou une mesure de dissimilarité et un critère d'agrégation.
- La représentation graphique généralement associée est le dendrogramme.
- Dans le cas de la distance euclidienne, on utilise généralement le critère d'agrégation de Ward qui correspond à une augmentation optimale de la variance intra-classe.
- Les différents critères d'agrégation ne conduisent pas toujours au même résultat.
- Il est possible de déterminer une partition en effectuant une coupe au premier saut d'indice jugé significatif
- Il n'est pas toujours facile de lire l'arbre hiérarchique (si n est grand).
- La CAH peut devenir très lente si n est grand.

Utilisation du logiciel R

```
# Matrice des distances
D <- dist(data, method = "euclidean")

# Mise en oeuvre de l'algorithme
H <- hclust(D, method="ward.D2")

# Représentation graphique (dendrogramme)
plot(H)

# Coupe de l'arbre pour trouver la meilleure partition en K=2 classes
classes <- cutree(H, k=2)

# Graphique dans le premier plan principal (ACP) avec classes colorées
ACP <- princomp(data)
plot(ACP$scores[,1],ACP$scores[,2],col=classes,pch=classes)
```

Utilisation du logiciel R

```
# Utilisation de la fonction agnes de la librairie cluster
library(cluster)

# Matrice des distances
D <- dist(data, method = "euclidean")

# Mise en oeuvre de l'algorithme
H <- agnes(D, method="ward")

# Représentation graphique (dendrogramme)
plot(H,which.plots=2)
# De manière plus lisible
plot(as.hclust(H),hang=-1)

# Dessin de rectangles autour des classes (sur le dendrogramme)
rect.hclust(as.hclust(H),k=2,border="red")
```