

## CHAPITRE 3

# *Apprentissage*

Hugo SALOU MPI\*

Dernière mise à jour le 29 novembre 2022

---

## Table des matières

<b>0 Motivation</b>	<b>2</b>
<b>1 Vocabulaire</b>	<b>2</b>
<b>2 Apprentissage supervisé</b>	<b>2</b>
2.1 $k$ plus proches voisins . . . . .	3
2.2 Arbres $k$ -dimensionnels . . . . .	4
2.3 Algorithme $\text{ID}_3$ . . . . .	5
<b>3 Apprentissage non supervisé</b>	<b>10</b>
3.1 Algorithme $\text{HAC}$ , classification hiérarchique ascendant . . . . .	10
3.2 $k$ -moyenne . . . . .	11

---

## 0 Motivation

L'intelligence artificielle est vu comme un « objet magique » mais ce n'est pas le cas : c'est ce que nous allons étudier dans ce chapitre. Il existe plusieurs méthodes permettant l'apprentissage : descente de gradient,  $k$ -plus proches voisins, ...

La base de donnée la plus utilisée est MNIST : elle contient 60 000 images de  $28 \times 28$  pixels représentant un chiffre, et le chiffre correspondant. L'idée de l'apprentissage est de « deviner » le chiffre dessiné en connaissant l'image.

## 1 Vocabulaire

**Définition:** On appelle *signature de données* un  $n$ -uplet de paires nom, ensemble ; on le typographie

$$(\text{nom}_1 : S_1, \text{nom}_2 : S_2, \dots, \text{nom}_n : S_n).$$

EXEMPLE: 1.  $S_1 = (\text{titre} : \text{string}, \text{longueur} : \mathbb{N}, \text{date} : \mathbb{N})$ ,  
2.  $S_2 = (x : \mathbb{R}, y : \mathbb{R})$ ,  
3.  $S_3 = (R : \llbracket 0, 255 \rrbracket, G : \llbracket 0, 255 \rrbracket, B : \llbracket 0, 255 \rrbracket)$ .

**Définition:** Étant donné une signature de données  $S = (\text{nom}_1 : S_1, \dots, \text{nom}_n : S_n)$ , on appelle *donnée* un vecteur

$$\bar{v} = (v_1, v_2, \dots, v_n) \in S_1 \times S_2 \times \dots \times S_n.$$

EXEMPLE: 1. (“2001, a space odyssey”, 139, 1968) est une donnée/un vecteur de signature  $S_1$ .  
2.  $(\pi, \sqrt{2})$  est une donnée/un vecteur de signature  $S_2$ .

**Définition:** Étant donné une signature de données  $S$ , on appelle *jeu de données* un ensemble fini de vecteurs de signature  $S$ .

**Définition:** Étant donnée une signature de données  $S$  et un ensemble de classes  $\mathcal{C}$ , on appelle *jeu de données classifié* la donnée

- d'un jeu de données  $S$ ,
- d'une fonction  $f : S \rightarrow \mathcal{C}$  de classification.

## 2 Apprentissage supervisé

L'objectif de cette section est de construire des fonctions de classification, à partir d'un jeu de données classifié.

**Définition:** Étant donné une signature de données  $S$ , et un ensemble de classes  $\mathcal{C}$ , on appelle *fonction de classification* une fonction des données de signature  $S$  dans  $\mathcal{C}$ .

REMARQUE:

On discutera de la « qualité » d'une fonction de classification en fonction de ses résultats sur les données d'un jeu de données et sur des exemples de tests.

## 2.1 $k$ plus proches voisins



FIGURE 1 – Représentation de l'algorithme des  $k$  plus proches voisins

---

### Algorithme 1 $k$ -NN ( $k$ nearest neighbors)

---

**Entrée** Un jeu de données classifié  $(S, c)$ , un vecteur d'entrée  $\bar{v}$

- 1: On trie  $S$  par distance croissante de  $v$  en  $d_1, d_2, \dots, d_k, d_{k+1}, \dots$
  - 2: Soit  $D$  un dictionnaire de  $\mathcal{E}$  vers  $\mathbb{N}$  initialisé à  $0^1$
  - 3: **pour**  $j \in \llbracket 1, k \rrbracket$  **faire**
  - 4:  $D[c(d_j)] \leftarrow D[c(d_j)] + 1$
  - 5: **retourner**  $\arg \max_{d \in \mathcal{E}} D[d]$
- 

REMARQUE:

On doit avoir  $k \leq n$ , et l'espace doit être muni d'une distance. Les résultats de l'algorithme dépendent fortement du jeu de données, du paramètre  $k$  et de la distance choisie.

### Matrice de confusion

**Définition:** On appelle *matrice de confusion* d'un algorithme de prédiction  $\mathcal{A}$  sur un jeu de données classifié  $(T, c)$ , la matrice

$$\left( \text{Card}\{t \in T \mid \mathcal{A}(t) = i \text{ et } c(t) = j\} \right)_{(i,j) \in \mathcal{E}^2}.$$

Dans le cas particulier d'une classification  $(V, F)$ , on nomme

---

1. où toutes les valeurs sont initialisées à 0, pas un dictionnaire vide

	vrai	$F$	$V$
$\mathcal{A}$		vrai négatif	faux négatif
	$F$	faux positif	vrai positif

TABLE 1 – Matrice de confusion dans le cas d’une classification en  $V$  et  $F$

*Comment améliorer la performance de l’algorithme des  $k$  plus proches voisins?* En dimension 1, on peut utiliser une dichotomie. Mais, dans des dimensions plus grandes, l’ordre lexicographique, et l’ordre produit ne fonctionnent pas. Mais, on peut appliquer une “dichotomie” en changeant de dimension. Par exemple, en deux dimension, on a

À faire : Représenter le schéma

FIGURE 2 – Représentation de la “dichotomie” en dimension 2

Pour représenter cette structure de données, on utilise un arbre binaire comme montré ci-dessous. Cet arbre est appelé un arbre  $k$ -dimensionnels.

À faire : Faire l’arbre

FIGURE 3 – Arbre 2-dimensionnel représentant la “dichotomie” précédente

## 2.2 Arbres $k$ -dimensionnels

REMARQUE (Notations):

Étant donné un jeu de données  $S$ , on note pour  $v \in S$ ,

$$S^{\leq_i v} = \{u \in S \mid u_i \leq v_i\} \quad \text{et} \quad S^{>_i v} = \{u \in S \mid u_i > v_i\}.$$

**Algorithme 2 “F”** : Fabrication d’un arbre  $k$ -dimensionnel

**Entrée**  $\mathcal{V}$  un jeu de données et  $i \in \llbracket 0, n - 1 \rrbracket$ , où  $n$  est la dimension des données

- 1: **si**  $\mathcal{V} = \emptyset$  **alors**
- 2: |   **retourner** Vide
- 3: **sinon**
- 4: |   On cherche  $v \in \mathcal{V}$  tel que  $v_i$  est la médiane de  $\{u_i \mid u \in \mathcal{V}\}$
- 5: |   **retourner** Nœud $\left((v, i), \mathbf{F}(\mathcal{V} \setminus \{v\})^{\leq_i v}, i + 1 \bmod n), \mathbf{F}(\mathcal{V} \setminus \{v\})^{>_i v}, i + 1 \bmod n\right)$

---

**Algorithme 3 “R”** : Recherche du point le plus proche

---

**Entrée** Un arbre  $k$ -dimensionnel et un vecteur  $v$ 

```
1: si  $T$  est vide alors
2:   | retourner None
3: sinon
4:   |  $Nœud((u, i), G, D) \leftarrow T$ 
5:   | si  $u_i \leq v_i$  alors
6:     |  $W \leftarrow R(D, v)$ 
7:     | si  $W = \text{None}$  alors
8:       |  $W' \leftarrow R(G, v)$ 
9:       | si  $W' = \emptyset$  alors
10:        | retourner  $\text{Some}(u)$ 
11:        | sinon
12:          |  $\text{Some}(z) \leftarrow W'$ 
13:          | retourner le plus proche de  $v$  entre  $u$  et  $z$ 
14:        | sinon
15:          |  $\text{Some}(w) \leftarrow W$ 
16:          | si  $v_i - u_i \leq d(w, v)$  alors  $\triangleright d(w, v)$  représente la distance entre  $w$  et  $v$ 
17:            |  $W' \leftarrow R(G, v)$ 
18:            | si  $W' = \text{None}$  alors
19:              | retourner  $\text{Some}(\text{plus proche de } v \text{ entre } u \text{ et } w)$ 
20:              | sinon
21:                |  $\text{Some}(z) \leftarrow w$ 
22:                | retourner  $\text{Some}(\text{plus proche de } v \text{ entre } u, z, \text{ et } w)$ 
23:              | sinon
24:                | retourner  $W$ 
25:            | sinon
26:              |  $\langle \text{comme le cas précédent} \rangle$ 
```

---

### 2.3 Algorithme m3

L'algorithme des  $k$  plus proches voisins (sans arbres  $k$ -dimensionnels) n'a pas de *phase d'apprentissage* : en effet, les données ne sont pas réorganisées. Mais, par exemple, pour l'utilisation des arbres  $k$ -dimensionnels, les données sont réorganisées dans un arbre.

Ce qu'on aimerai avoir, c'est des *bordures* entre les différentes classes. Par exemple, dans l'exemple précédent, on aimerai avoir les différentes zones ci-dessous.

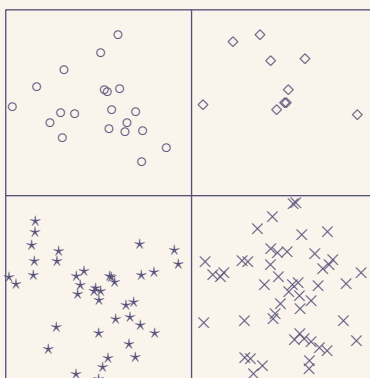


FIGURE 4 – Représentation de *bordures* entre les différentes classes

De ces zones, on peut construire un algorithme qui classe les données, que l'on représente sous forme d'arbre. Ce type d'arbre est un *arbre de décision*. Dans l'exemple précédent, on peut donc créer l'arbre ci-dessous.

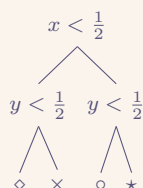


FIGURE 5 – Arbre de décision pour la classification  
 À faire : Refaire l'arbre plus proprement

Dans le reste de cette section, on s'intéresse uniquement à des données de  $\mathbb{B}^n$  (une liste de  $n$  booléens) pour un certain  $n \in \mathbb{N}$ .

On considère l'exemple dont les données ci-dessous.

Transport	Moteur	Rails	Sous-terre	$\geq 320$ km/h	Train ?
A380	✓	×	×	✓	×
TGV	✓	✓	×	✓	✓
Métro	✓	✓	✓	×	✓
Wagonnet	×	✓	✓	×	×
Draisine	×	✓	×	×	×
Tram	✓	✓	×	×	✓

TABLE 2 – Exemple de données

### Entropie

**Définition:** Étant donné un variable aléatoire finie  $X$  à valeurs dans  $E$ . On note  $p_X : E \rightarrow [0, 1]$  sa loi de probabilité :

$$\forall x \in E, \quad p_X(x) = P(X = x).$$

On définit l'entropie  $H(X)$  de cette variable aléatoire comme

$$H(X) = - \sum_{x \in E} p_X(x) \ln(p_X(x)).$$

On prolonge  $p_X(x) \ln(p_X(x))$  par continuité à la valeur 0 lorsque  $p_X(x) = 0$ .

EXEMPLE:

On considère la variable aléatoire  $X$  à valeur dans  $\{\bullet, \bullet\}$  telle que  $P(X = \bullet) = 1$  et  $P(X = \bullet) = 0$ . On a

$$H(X) = -0 \ln 0 - 1 \ln 1 = 0.$$

EXEMPLE:

On considère la variable aléatoire  $X$  à valeur dans  $\{\bullet, \bullet\}$  telle que  $P(X = \bullet) = p$  et  $P(X = \bullet) = 1 - p$ , avec  $p \in [0, 1]$ . On a alors

$$H(X) = -p \ln p - (1 - p) \ln(1 - p)$$

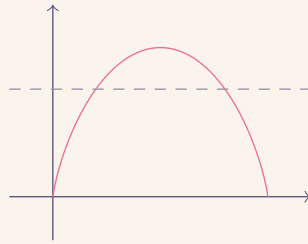


FIGURE 6 – Représentation graphique de  $H(X)$  en fonction de  $p$

**Lemme:** Si  $(p_i)_{i \in [1, n]} \in ]0, 1]^n$  sont tels que  $\sum_{i=1}^n p_i = 1$ . Soit  $(q_i)_{i \in [1, n]}$  tels que  $\forall i, q_i = \frac{1}{n}$ . On a alors

$$-\sum_{i=1}^n p_i \ln p_i \leq -\sum_{i=1}^n p_i \ln(q_i).$$

*Preuve:*

On a

$$\sum_{i=1}^n p_i \ln(q_i) - \sum_{i=1}^n p_i \ln(p_i) = \sum_{i=1}^n p_i \ln\left(\frac{q_i}{p_i}\right) \leq \ln\left(\sum_{i=1}^n p_i \frac{q_i}{p_i}\right) = 0.$$

□

**Propriété:** Soit  $n = |E|$ . L'entropie d'une variable aléatoire à valeurs dans  $E$  est maximale lorsque

$$\forall e \in E, P(X = e) = \frac{1}{n}.$$

*Preuve:*

On conclut, d'après le lemme précédent, que ...

□

**Définition:** Étant donné un jeu de données classifiés  $(S, c)$  où  $c : S \rightarrow E$ , on appelle *entropie* de ce jeu de données l'entropie de la variable aléatoire  $c(Y)$  où  $Y \sim \mathcal{U}(S)$ . On a donc

$$H((S, c)) = -\sum_{e \in E} \frac{\text{Card } c^{-1}(\{e\})}{\text{Card } S} \ln\left(\frac{\text{Card } c^{-1}(\{e\})}{\text{Card } S}\right).$$

**Définition:** Étant donné un jeu de données une partition  $\{S_1, S_2, \dots, S_p\}$  d'un jeu de données classifié  $(S, c)$ , l'*entropie* de cette partition est la moyenne (pondérée par les cardinaux et renormalisée) :

$$H(\{S_1, \dots, S_p\}, c) = \sum_{i=1}^p \frac{\text{Card } S_i}{\text{Card } S} H((S_i, c)).$$

L'entropie de  $\{w, a, t, d, r, m\}$  est  $H = -\frac{3}{6} \ln\left(\frac{3}{6}\right) - \frac{3}{6} \ln\left(\frac{3}{6}\right) = \ln 2 \simeq 0,69$ . Mais, avec le



découpage de l'arbre de décision ci-dessous, on obtient l'entropie

$$\begin{aligned}
 H &= \frac{2}{6}H(\{w; d\}) + \frac{4}{6}H(\{a, t, r, m\}) \\
 &= \frac{2}{6} \times 0 + \frac{4}{6} \times \left( -\frac{1}{4} \ln\left(\frac{1}{4}\right) - \frac{3}{4} \ln\left(\frac{3}{4}\right) \right) \\
 &\simeq 0,37.
 \end{aligned}$$

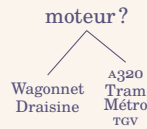


FIGURE 7 – Arbre de décision possible se basant sur le moteur

Avec un autre arbre (comme celui ci-dessous), on obtient une entropie différente :

$$\begin{aligned}
 H &= \frac{1}{6}H(\{a\}) + \frac{5}{6}H(\{w, d, t, r, m\}) \\
 &= \frac{5}{6} \left( -\frac{2}{5} \ln\left(\frac{2}{5}\right) - \frac{3}{5} \ln\left(\frac{3}{5}\right) \right) \\
 &\simeq 0,56.
 \end{aligned}$$



FIGURE 8 – Arbre de décision possible se basant sur les rails

Pour le sous-terrain, on a  $H = \ln 2$ .



FIGURE 9 – Arbre de décision possible se basant sur sous-terrain

À faire : Vérifier

À faire : Autre cas

Ainsi, on choisit de commencer avec la condition "moteur" car l'entropie est la plus faible avec cette condition. Ainsi, l'arbre de décision ressemble à celui ci-dessous.

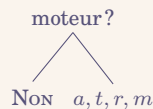


FIGURE 10 – Arbre de décision partiel

On réitère avec les autres conditions. L'entropie en se basant sur la vitesse est  $\frac{1}{2} \ln 2 \simeq 0,34$ . En effet, l'arbre de décision possible ressemble à celui ci-dessous.



FIGURE 11 – Arbre de décision possible se basant sur le moteur puis la vitesse

Mais, en se basant sur sous-terrain, on obtient une entropie de  $\frac{3}{4} (-\frac{2}{3} \ln(\frac{2}{3}) - \frac{1}{3} \ln(\frac{1}{3})) \simeq 0,48$ .

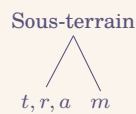


FIGURE 12 – Arbre de décision possible se basant sur le moteur puis sous-terrain

Et, en se basant sur les rails, on obtient une entropie de 0.

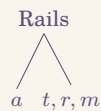


FIGURE 13 – Arbre de décision possible se basant sur le moteur puis les rails

On en déduit que l'arbre final de décision est celui ci-dessous.

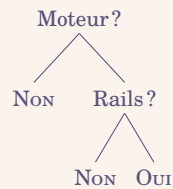


FIGURE 14 – Arbre de décision final pour la classification de trains

Les données que l'on a utilisées sont pour l'apprentissage. On teste notre arbre de décision sur les données ci-dessous.

Nom	Moteur	Rail	Sous-terre	Vitesse	Résultat de l'algorithme
Bus	✓	×	×	×	×
TER	✓	✓	×	×	✓
Cheval	×	×	×	×	×
Ascenseur spatial	✓	✓	×	×	✓

TABLE 3 – Test de l'arbre de décision créé

Attention : il ne faut pas faire de *sur-apprentissage*, comme montré sur la figure ci-dessus. À faire : Figure sur-apprentissage Aussi, il faut faire attention aux critères : par exemple, lors de la classification de photos de chats et de chiens, les photos de chiens sont en général prises

en extérieur et l'algorithme  $\text{ID}_3$  aurait donc pu choisir de baser sa décision sur l'emplacement de la photo, même si elle n'importe pas dans la différenciation chat/chien.

Autre exemple : on considère la table de données ci-dessous. Utilisons l'algorithme  $\text{ID}_3$  sur ces données, et trouvons l'arbre de décision.

$A$	$B$	$C$	$D$	Classification
✓	×	×	✓	●
✓	✓	×	✓	●
✓	×	✓	✓	●
✓	✓	✓	✓	●
×	×	✓	✓	●
×	✓	✓	×	●
×	×	×	×	●
×	✓	×	✓	●

TABLE 4 – Table de données d'exemple

Pour les conditions  $A$ ,  $B$  et  $C$ , on a  $H \simeq 0,63$  et, pour  $D$ , on a  $H = 0,65$ . Comme on prend le 1<sup>er</sup> dans l'ordre lexicographique, on choisit la condition  $A$ . De même, on construit l'arbre ci-dessous.

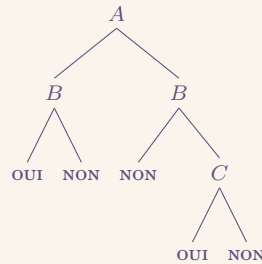


FIGURE 15 – Arbre de décision pour la table de données précédente

Le nom de  $\text{ID}_3$  vient de *iterative dichotomizer*.

### 3 Apprentissage non supervisé

On rappelle que l'*apprentissage non supervisé* utilise des données non classifiées. L'objectif est de les classifier. Plus rigoureusement, étant donné un jeu de données  $\mathcal{D}$ , on veut partitionner  $\mathcal{D}$  en  $\{S_1, S_2, \dots, S_n\}$  où les  $S_i$  sont regroupés par « similitude. »

REMARQUE:

Dans le reste de cette section,  $\mathcal{D} \subseteq \mathbb{R}^p$ , avec  $p \in \mathbb{N}^*$ , et la notion de similitude est donnée par la distance euclidienne.

#### 3.1 Algorithme HAC, classification hiérarchique ascendant

**Définition:** Étant donné deux sous-ensembles  $A$  et  $B$  de  $\mathcal{D}$  disjoints et non-vides, on définit différentes mesures de dissimilarité :

- $D(A, B) = \min\{d(a, b) \mid a \in A, b \in B\}$ ;
- $D(A, B) = \max\{d(a, b) \mid a \in A, b \in B\}$ ;

---

—  $D(A, B) = \frac{1}{|A||B|} \times \sum_{(a,b) \in A \times B} d(a, b)$ ,  
où  $d(a, b)$  représente la distance entre  $a$  et  $b$ .

---



---

**Algorithme 4** Algorithme HAC

---

- 1:  $P \leftarrow \{\{x\} \mid x \in \mathcal{D}\}$
  - 2: **tant que**  $|P| \geq 2$  **et** ?<sup>2</sup> **faire**
  - 3:     Soit  $A, B \in P$  minimisant  $D(A, B)$  avec  $A \neq B$
  - 4:      $P \leftarrow (P \setminus \{A, B\}) \cup \{A \cup B\}$
  - 5: **retourner**  $P$
- 

### 3.2 $k$ -moyenne

Au préalable, on fixe le nombre de classes  $k$  que l'on souhaite obtenir après exécution de l'algorithme.

**Propriété:** On considère la fonction

$$f : \mathbb{R}^p \longrightarrow \mathbb{R}$$

$$H \longmapsto \sum_{v \in \mathcal{D}} \|H - v\|_2^2.$$

Cette fonction atteint son minimum en le barycentre  $G$ .

*Preuve:*

Soit  $G$  le barycentre de  $\mathcal{D}$ . Soit  $H \in \mathbb{R}^p$ . Montrons que  $f(H) > f(G)$ . On calcule

$$\begin{aligned} f(H) &= f(H - G + G) \\ &= \sum_{v \in \mathcal{D}} \langle (H - G + G) - v \mid (H - G + G) - v \rangle \\ &= \sum_{v \in \mathcal{D}} \left( \langle G - v \mid G - v \rangle + 2\langle H - G \mid H - v \rangle + \langle H - G \mid H - G \rangle \right) \\ &= f(G) + \underbrace{|\mathcal{D}| \cdot \|H - G\|_2^2}_{\geq 0} + 2 \underbrace{\langle H - G \mid \sum_{v \in \mathcal{D}} (G - v) \rangle}_{= 0} \\ &\hspace{15em} \text{par définition du barycentre} \end{aligned}$$

Donc, si  $H \neq G$ , on a bien  $f(H) > f(G)$ . □

Dans la suite de cette sous-section, on s'intéresse à des partitionnement particuliers : étant donné une famille  $(m_i)_{i \in \llbracket 1, k \rrbracket}$  de vecteurs de  $\mathbb{R}^p$ , et un entier  $j \in \llbracket 1, k \rrbracket$ , on définit

$$\mathcal{C}_j^m = \left\{ v \in \mathcal{D} \mid \arg \min_{i \in \llbracket 1, k \rrbracket} \|v - m_i\|_2 = j \right\}.$$

**REMARQUE:**

En cas d'égalité ( $\|v - m_{i_1}\|_2 = \|v - m_{i_2}\|_2$ ), le "arg min" retourne le plus petit indice :  $\min(i_1, i_2)$ .

Étant donné une famille  $(m_i)_{i \in \llbracket 1, k \rrbracket}$  de vecteurs de  $\mathbb{R}^p$ , on a donc un partitionnement  $\{\mathcal{C}_j^m \mid j \in \llbracket 1, k \rrbracket\}$ .

---

2. où, pour ?, on peut choisir (1) un critère sur le nombre de classes (2) un critère sur la valeur de la plus petite dissimilarité

---

**Définition:** Étant donné une famille  $(m_i)_{i \in \llbracket 1, k \rrbracket}$  de vecteurs de  $\mathbb{R}^p$ , on définit

$$L(m) = \sum_{k=1}^n \left( \sum_{v \in \mathcal{C}_i^m} \|v - m_i\|_2^2 \right).$$

Pour minimiser localement la fonction  $L$ , on aimerait que les  $(m_i)_{i \in \llbracket 1, k \rrbracket}$  soient les barycentres des  $(\mathcal{C}_i^m)_{i \in \llbracket 1, k \rrbracket}$ . Mais, la définition des  $(\mathcal{C}_i^m)$  dépend de la position des  $(m_i)$ . Ainsi, en itérant ce procédé, en modifiant les  $(m_i)$  pour être les barycentres des  $(\mathcal{C}_i^m)$  respectifs, puis en modifiant les  $(\mathcal{C}_i^m)$ , la famille  $(m_i)_{i \in \llbracket 1, k \rrbracket}$  converge vers les barycentres des classes.

---

**Algorithme 5**  $k$ -moyenne

**Entrée**  $\mathcal{D}$  et  $k$

1:  $(m_i)_{i \in \llbracket 1, k \rrbracket} \leftarrow k$  vecteurs de  $\mathcal{D}$

2: stable  $\leftarrow \mathbf{F}$

3: **tant que**  $\neg$ stable **faire**

4:      $C \leftarrow (\mathcal{C}_i^m)_{i \in \llbracket 1, k \rrbracket}$

5:      $m' \leftarrow (\text{barycentre}(\mathcal{C}_i))_{i \in \llbracket 1, k \rrbracket}$

6:     stable  $\leftarrow m \stackrel{?}{=} m'$

7:      $m \leftarrow m'$

8: **retourner**  $C$

---