
Classification non supervisée de données fonctionnelles

A. Choury, Alykis SAS

anna.choury@alykis.com

1 Introduction

1.1 Définition de la classification

L'objectif d'une classification est la recherche d'une structure à l'intérieur de l'espace des données au travers une segmentation. Il s'agit de regrouper les individus ayant des propriétés similaires en classes, les plus homogènes possibles et ainsi obtenir une partition des données.

On distingue la classification supervisée, appelée couramment classification, qui nécessite un échantillon d'apprentissage dont le classement est connu, de la classification non supervisée, appelée clustering, dont les classes ne sont pas définies a priori.

Nous nous restreindrons au cas du clustering et répondrons à la question : "*comment, à partir d'un ensemble de données, peut-on créer une structure de classes homogènes ?*"

1.2 Exemples

1.2.1 Les Simpsons

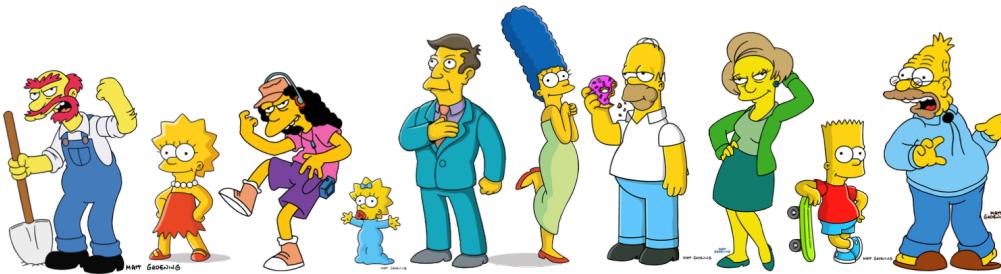
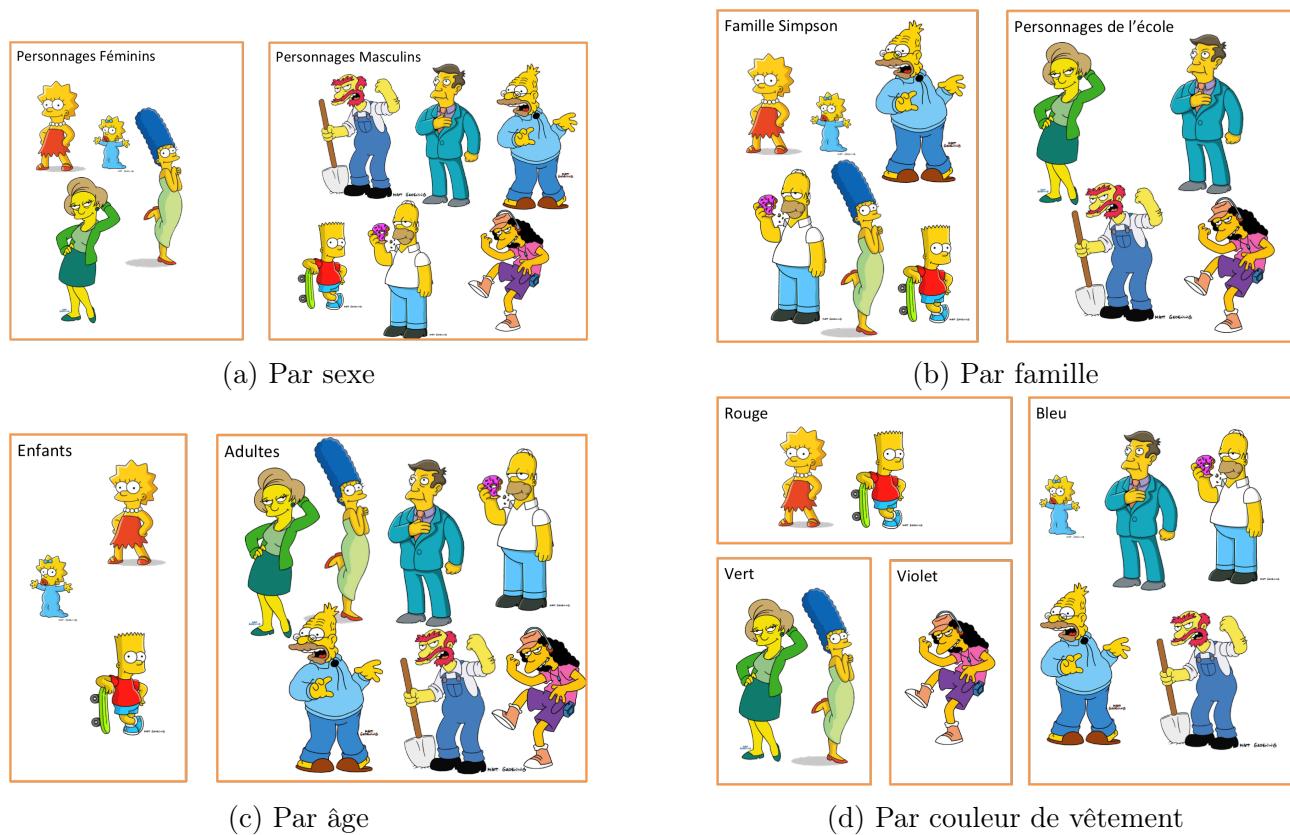


FIGURE 1 – Quelles sont les partitions possibles des personnages des *Simpsons* ?

L'exemple de la classification des personnages des Simpson introduit la notion de similarité. Sous quel(s) critère(s) peut-on regrouper les individus ?

FIGURE 2 – Propositions de clustering des personnages des *Simpsons*

1.2.2 Nuages de points



FIGURE 3 – Quelles sont les partitions possibles pour ces points ?



(a) 2 classes



(b) 4 classes



(c) 6 classes

FIGURE 4 – Propositions de clustering du nuage de points

L'exemple du clustering du nuage de points introduit la notion d'homogénéité des classes. On commence à parler de variance intra-classe à minimiser et de variance inter-classe à maximiser.

1.3 Pourquoi classifier

La classification permet donc de créer des groupes homogènes. On définit ainsi la structure interne des données. Par exemple figure 5, un clustering sur des trajectoires de taxi à San Francisco fait apparaître des zones géographiques dans la ville caractérisées par des comportements similaires.

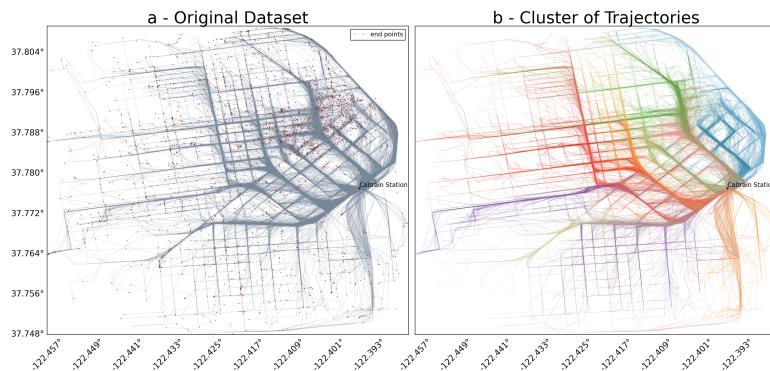


FIGURE 5 – Un clustering fait apparaître des zones géographiques

La création de ces groupes peut parfois être un but en soi, par exemple en segmentation pour le marketing. L'identification de sous-ensembles d'acheteurs partageant des besoins et des comportements d'achats similaires va permettre un ciblage plus efficace de la publicité, et donc une meilleure atteinte du client potentiel. Un exemple en est donné figure 6 avec les différentes sortes de Coca-Cola. Le public ciblé n'est pas le même pour le Coca-cola original, le Coca light et le Coca zéro.



FIGURE 6 – Exemple de segmentation marketing

Le clustering peut être une façon de réduire l'information, lorsque l'on veut résumer des données. Le but étant alors d'extraire un représentant de chaque groupe pour résumer l'information. Figure 7 les vitesses moyennes par classe sont représentées sur la droite. Chaque profil de vitesse résume l'information du jour de la semaine.

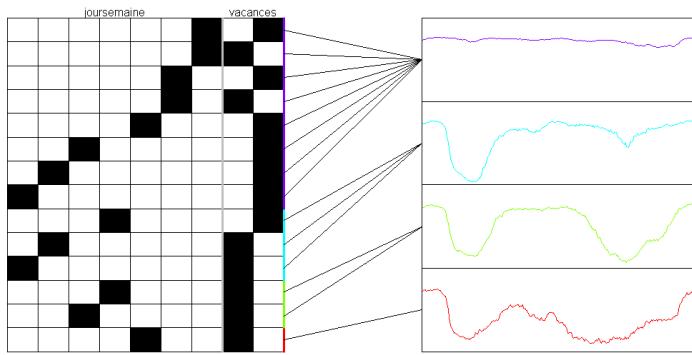


FIGURE 7 – Un représentant de chaque classe résume l’information

2 Formalisme

L’objectif de la classification non supervisée est d’identifier des sous-ensembles homogènes C_i de $\{(x_i, y_i) \subset \mathcal{X} \times \mathcal{Y}\}$ tels que

- pour $i \neq j$, C_i est bien distinct de C_j
- les éléments de C_i sont similaires

Un calcul élémentaire de combinatoire montre que le nombre de partitions possibles d’un ensemble de n éléments croît plus qu’exponentiellement avec n .

Le nombre de partitions de n éléments en k classes est le nombre de Stirling de seconde espèce :

$$S(n, k) = \left\{ \begin{matrix} n \\ k \end{matrix} \right\} = \frac{1}{k!} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} j^n \quad (1)$$

Le nombre total de partition est le nombre de Bell B_n , qui correspond à la somme des nombres de Sterling de seconde espèce :

$$B_n = \sum_{k=1}^n \left\{ \begin{matrix} n \\ k \end{matrix} \right\} = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!} \quad (2)$$

Pour $n = 20$ éléments, le nombre de partitions est de l’ordre de 10^{13} . Il n’est donc pas question de chercher à optimiser un critère sur toutes les partitions possibles. Les méthodes se limitent à l’exécution d’un algorithme itératif convergeant vers une « bonne » partition qui correspond en général à un optimum local.

L’utilisateur doit donc effectuer les choix suivants :

- distance entre les individus
- critère d’homogénéité des classes à optimiser
- la méthode de clustering

2.1 Choix de la distance

Pour regrouper les individus en classes, il est nécessaire de pouvoir définir une mesure d’éloignement entre eux. Soit $\Omega = \{x_1, \dots, x_n\}$ l’ensemble des individus à classifier, on définit sur Ω trois types de mesures d’éloignement : la similarité (2.1.1), la dissimilarité (2.1.2) et la distance (2.1.3).

2.1.1 Similarité

La similarité mesure la ressemblance entre observations. Elle est définie par une fonction $s : \Omega \times \Omega \rightarrow \mathbb{R}_+$ telle que

- $s(x_1, x_2) = s(x_2, x_1) \geq 0$
- $s(x_i, x_i) = S > 0$
- $s(x_1, x_2) \leq S$

Un indice de ressemblance normé $s^* : \Omega \times \Omega \rightarrow [0, 1]$ est donc défini par

$$s^*(x_1, x_2) = \frac{1}{S} s(x_1, x_2)$$

La valeur absolue du coefficient de corrélation est un exemple de similarité :

$$|\rho(x_1, x_2)| = \left| \frac{\sum_{i=1}^N (x_{1i} - \bar{x}_{1\bullet})(x_{2i} - \bar{x}_{2\bullet})}{\sqrt{\sum_{i=1}^N (x_{1i} - \bar{x}_{1\bullet})^2} \sqrt{\sum_{i=1}^N (x_{2i} - \bar{x}_{2\bullet})^2}} \right|$$

2.1.2 Dissimilité

A l'inverse, on définit une dissimilité comme une fonction $d : \Omega \times \Omega \rightarrow \mathbb{R}_+$ telle que

- $d(x_1, x_2) = d(x_2, x_1) \geq 0$
- $d(x_1, x_2) = 0 \Rightarrow x_1 = x_2$

La relation entre une dissimilité d et une similarité s se résume pour $(x_1, x_2) \in \Omega \times \Omega$

$$d(x_1, x_2) = S - s(x_1, x_2)$$

Remarquons qu'une distance est une dissimilité, puisque toute distance possède les deux propriétés ainsi que l'inégalité triangulaire. Toutes les distances connues, en particulier la distance euclidienne, sont donc des exemples de dissimilité.

2.1.3 Distance

Une distance sur Ω est, par définition, une dissimilité vérifiant en plus la propriété d'inégalité triangulaire. Une distance $d : \Omega \times \Omega \rightarrow \mathbb{R}_+$ vérifie

- $d(x_1, x_2) = d(x_2, x_1) \geq 0$
- $d(x_1, x_2) = 0 \Rightarrow x_1 = x_2$
- $d(x_1, x_2) \leq d(x_1, x_3) + d(x_3, x_2)$

Le choix de la distance est primordial pour la construction des classes. De ce choix va dépendre la nature des données à classifier. Nous y reviendrons section 6.

2.2 Critère d'homogénéité des classes

Il est nécessaire de formaliser les notions d'homogénéité dans les classes et de distinction entre classes de façon géométrique.

On considère n points x_1, \dots, x_n et on désigne par x_G le barycentre du nuage de ces points :

$$x_G = \frac{1}{n} \sum_{i=1}^n x_i \quad (3)$$

L'inertie totale est définie par

$$I_T = \sum_{i=1}^n d^2(x_i, x_G) \quad (4)$$

On choisit pour d la distance euclidienne : $d(x_1, x_2) = \|x_1 - x_2\|$.

On suppose qu'en réalité le nuage de points est constitué de K classes C_1, \dots, C_K . On note x_{C_k} le barycentre de la classe C_k . On reprend alors l'équation (4) :

$$I_T = \sum_{i=1}^n \|x_i - x_G\|^2 \quad (5)$$

$$= \sum_{k=1}^K \sum_{i \in C_k} \|x_i - x_G\|^2 \quad (6)$$

$$= \sum_{k=1}^K \sum_{i \in C_k} \|x_i - x_{C_k} + x_{C_k} - x_G\|^2 \quad (7)$$

En appliquant le Théorème de Hyugens sur les moments d'inertie, on obtient

$$I_T = \sum_{k=1}^K \sum_{i \in C_k} (\|x_i - x_{C_k}\|^2 + \|x_{C_k} - x_G\|^2) \quad (8)$$

$$= \underbrace{\sum_{k=1}^K \sum_{i \in C_k} d^2(x_i, x_{C_k})}_{(1)} + \underbrace{\sum_{k=1}^K n_k d^2(x_{C_k}, x_G)}_{(2)} \quad (9)$$

Où n_k est le cardinal de la classe C_k .

Le terme (1) de l'équation (9) mesure la somme des distances entre les points d'une classe et leur barycentre. Il s'agit donc de l'**inertie intra-classe**. Si les classes sont homogènes, l'inertie intra-classe est faible.

Le terme (2) de l'équation (9) mesure la distance des barycentres des classes au barycentre global. C'est une mesure d'éloignement des classes, appelée **inertie inter-classe**.

La partition optimale C_K^* des observations en K classes est définie par

$$C_K^* = \arg \min_{C \in C_K} \sum_{k=1}^K \sum_{i \in C_k} d^2(x_i, x_{C_k}) \quad (10)$$

où C_K est l'ensemble des partitions possibles des n observations en K classes.

A noter que le nombre K de classes est considéré comme *a priori* connu. La détermination du nombre de classes en fonction des données est un problème complexe que nous abordons section 3.4.

2.3 Choix de l'algorithme de clustering

Comme indiqué à l'équation (2), la complexité combinatoire du problème est trop importante pour chercher la partition optimale parmi toutes les partitions possibles. Les algorithmes de classification réduisent donc le temps de calcul en n'envisageant qu'un nombre restreint de partitions. Deux de ces algorithmes, dits heuristiques, sont présentés section 3 et 4 ci-dessous.

3 La Classification Ascendante Hiérarchique

La Classification Ascendante Hiérarchique, notée CAH, a pour but de construire une suite de partitions emboîtées des données de façon itérative.

3.1 Principe

L'initialisation de l'algorithme consiste à calculer une matrice de distance ou de dissimilarité entre les n individus à classer. L'algorithme part alors de la partition triviale de n singletons, où chaque individu constitue une classe et cherche à constituer des classes par agrégation des deux éléments les plus proches de la partition de l'étape précédente. L'algorithme s'arrête avec l'obtention d'une seule classe. Les regroupements successifs sont représentés sous la forme d'un *arbre binaire de classification* ou *dendrogramme*.

Ce dendrogramme représente une hiérarchie de partitions. Un exemple de dendrogramme est donné figure 8.

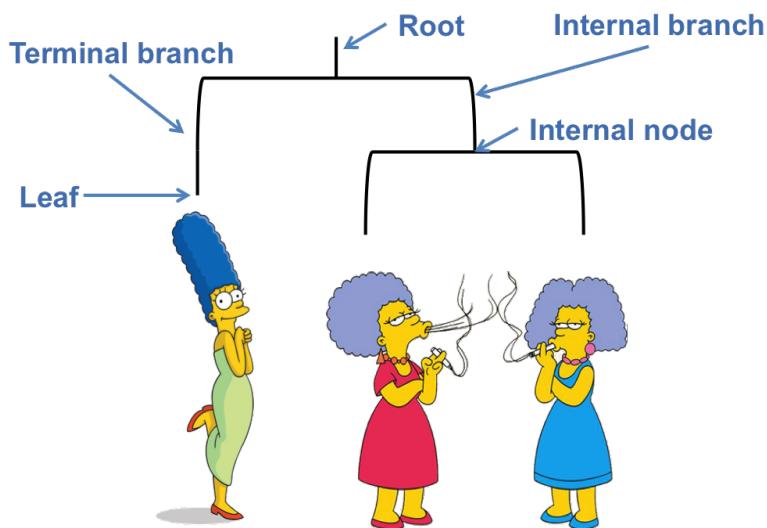


FIGURE 8 – Exemple de dendrogramme

On peut alors choisir une partition en tronquant l'arbre à un niveau donné. Par exemple Figure 9, choisir une partition à deux classes revient à considérer les branches en rouge comme étant les branches finales (*terminal branches*). On obtient alors une partition par sexe. Choisir une partition à quatre classes arrête le dendrogramme aux branches vertes. Un niveau de regroupement de plus par membres d'une même « famille » (Simpsons ou école).

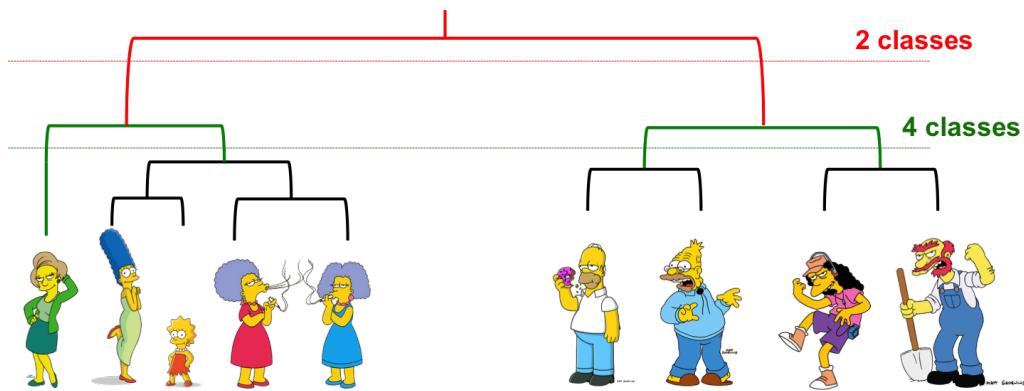


FIGURE 9 – Exemples de troncatures de dendrogramme

3.2 Distance inter-classe

A chaque étape de l'algorithme la matrice de distances (ou dissimilarités) est mise à jour. Après chaque regroupement, de deux individus, de deux classes ou d'un individu à une classe, les distances entre ce nouvel objet et les autres sont calculées et viennent remplacer, dans la matrice, les distances des objets qui viennent d'être agrégés.

Différentes stratégies peuvent être mises en place et dépendent du choix de la distance choisie et de la méthode dite de « saut » entre les classes. Notons A et B deux classes, ou éléments, d'une partition donnée, w_A et w_B leurs pondérations, et $d_{i,j}$ la distance entre deux individus quelconques i et j .

Le problème est de définir $d(A, B)$, distance entre deux éléments d'une partition de Ω .

3.2.1 cas d'une dissimilité

On présente trois stratégies de définition de distance inter-classe :

1. Le saut minimum, *single linkage* :

$$d(A, B) = \min_{i \in A, j \in B} (d_{ij}) \quad (11)$$

2. Le saut maximum ou diamètre, *complete linkage* :

$$d(A, B) = \sup_{i \in A, j \in B} (d_{ij}) \quad (12)$$

3. Le saut moyen, *group average linkage* :

$$d(A, B) = \frac{1}{\text{card}(A)\text{card}(B)} \sum_{i \in A, j \in B} d_{ij} \quad (13)$$

A noter que les stratégies ci-dessus s'appliquent également à des indices plus structures (distance) mais n'en utilisent pas toutes les propriétés.

3.2.2 cas d'une distance

La stratégie basée sur la **distance des barycentres** (*centroïd*) nécessite la connaissance de représentations euclidiennes des individus. On considère une matrice $n \times p$ des individus ou

une matrice $n \times n$ des distances euclidiennes entre les individus afin, au minimum, de pouvoir définir les barycentres notes g_A et g_B des classes. La distance inter-classe s'exprime alors par :

$$d(A, B) = d(g_A, g_B) \quad (14)$$

Enfin, en cas de distance euclidienne ou non-euclidienne, le **saut de Ward** a un rôle prépondérant. Considérons l'évolution de l'inertie intra-classe au fur et à mesure de la classification. A l'initialisation, toutes les classes sont composées d'une unique observation. Chaque classe est donc parfaitement homogène, et l'inertie intra-classe est nulle. A 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. Il est important de comprendre qu'à chaque étape de la classification, l'inertie intra-classe augmente alors que l'inertie inter-classe diminue. L'objectif est d'aboutir à une partition en K classes d'inertie intra-classe minimum et d'inertie inter-classe maximum. La stratégie va donc consister à regrouper les deux classes dont la fusion entraîne le plus faible perte d'inertie inter-classe.

A partir de l'élément (2) de l'équation (9) on reprend la définition de l'inertie inter-classe avant la fusion (I_{inter}^K) et après la fusion (I_{inter}^{K-1}) :

$$I_{inter}^K = w_A d^2(x_A, x_G) + w_B d^2(x_B, x_G) + \sum_{k \in \{1, \dots, K\} \setminus \{A, B\}} w_k d^2(x_{C_k} - x_G) \quad (15)$$

$$I_{inter}^{K-1} = (w_A + w_B) d^2(x_{AB}, x_G) + \sum_{k \in \{1, \dots, K\} \setminus \{A, B\}} w_k d^2(x_{C_k} - x_G) \quad (16)$$

avec x_A le barycentre de la classe A , x_B le barycentre de la classe B et x_{AB} le barycentre de la nouvelle classe. Par définition,

$$x_{AB} = \frac{w_A x_A + w_B x_B}{w_A + w_B} \quad (17)$$

En considérant pour d la distance euclidienne, la perte d'inertie inter-classe devient donc :

$$\begin{aligned} I_{inter}^{K-1} - I_{inter}^K &= w_A \|x_A - x_G\|^2 + w_B \|x_B - x_G\|^2 - (w_A + w_B) \left\| \frac{w_A x_A + w_B x_B}{w_A + w_B} - x_G \right\|^2 \\ &= w_A \|x_A - x_G\|^2 + w_B \|x_B - x_G\|^2 - \frac{1}{w_A + w_B} \|w_A x_A + w_B x_B - w_A x_G - w_B x_G\|^2 \\ &= w_A \|x_A - x_G\|^2 + w_B \|x_B - x_G\|^2 - \frac{1}{w_A + w_B} \|w_A(x_A - x_G) + w_B(x_B - x_G)\|^2 \\ &= w_A \|x_A - x_G\|^2 + w_B \|x_B - x_G\|^2 - \frac{w_A^2}{w_A + w_B} \|x_A - x_G\|^2 - \frac{w_B^2}{w_A + w_B} \|x_B - x_G\|^2 \\ &\quad - \frac{2w_A w_B}{w_A + w_B} \langle w_A - x_G, x_B - x_G \rangle \\ &= \frac{w_A w_B}{w_A + w_B} \|x_A - x_G\|^2 + \frac{w_A w_B}{w_A + w_B} \|x_B - x_G\|^2 - \frac{2w_A w_B}{w_A + w_B} \langle w_A - x_G, x_B - x_G \rangle \\ &= \frac{w_A w_B}{w_A + w_B} (\|x_A - x_G\|^2 + \|x_B - x_G\|^2 - 2 \langle w_A - x_G, x_B - x_G \rangle) \\ &= \frac{w_A w_B}{w_A + w_B} \|(x_A - x_G) - (x_B - x_G)\|^2 \\ &= \frac{w_A w_B}{w_A + w_B} \|x_A - x_B\|^2 \end{aligned}$$

La distance du saut de Ward est donc défini par cette mesure :

$$D_W^2(A, B) = \frac{w_A w_B}{w_A + w_B} d(x_A, x_B) \quad (18)$$

Ainsi ce critère induit, à chaque étape de regroupement, une minimisation de la décroissance de la variance inter-classe. C'est sur ce critère que reposera le choix le plus optimal possible du nombre de classes.

3.3 Algorithme

ALGORITHME 1 :

classification ascendante hierarchique

- **Initialisation** *Les classes initiales sont les singletons. Calculer la matrice de leurs distances deux à deux.*
- **Iterer** *les deux étapes suivantes jusqu'à l'aggregation en une seule classe :*
 1. *regrouper les deux classes les plus proches au sens de la “distance” entre classes choisie,*
 2. *mettre à jour le tableau de distances en remplaçant les deux classes regroupées par la nouvelle et en calculant sa “distance” avec chacune des autres classes.*

3.4 Choix du nombre de classes

Comme représenté figure 9, le dendrogramme créé par la CAH est un outil d'aide au choix du nombre de classes. Toutefois, le choix du nombre de classes est une décision délicate difficile à généraliser.

Une possibilité est d'observer la décroissance de la variance inter-classe dans le cas du saut de Ward et de refaire le raisonnement à l'inverse de la CAH : on part d'une seule classe et on arrête d'ajouter des classes lorsque cela ne diminue pas significativement la croissance inter-classe.

Tibshirani et al. ([19]) proposent de rationaliser ce raisonnement en introduisant la *statistique du gap* :

1. Soit D_r la somme de toutes les distances prises entre les observations deux à deux au sein d'une même classe $r = 1, \dots, K$.
2. Soit W_k la moyenne pondérée (par la taille de la classe) de ces sommes de distances. Si la distance initiale est euclidienne, W est (à un facteur 2 près) la norme carrée de la matrice de variance intra-classe.
3. On peut alors comparer le graphe de $\log(W_k)$ par rapport à celui d'une distribution de référence obtenue par simulation selon une loi uniforme. Le plus grand écart (*gap*) indique le nombre de classes optimal.

Enfin, dans le contexte de mélanges supposés gaussiens le choix du nombre de classes s'apparente à une sélection de modèle par des critères AIC, BIC, spécifiques.

3.5 Exemple

Le jeu de données USArrest disponible sous R est bien adapté à un exemple de Classification Ascendante Hiérarchique. Les individus statistiques sont les 50 Etats des Etats-Unis. Pour chaque Etat sont renseignés les quantités d'arrestations pour meurtre, agression et viol et le pourcentage de population urbaine. La table 1 montre les deux premières lignes du jeu de données.

| | Murder | Assault | UrbanPop | Rape |
|---------|--------|---------|----------|-------|
| Alabama | 13.20 | 236 | 58 | 21.20 |
| Alaska | 10.00 | 263 | 48 | 44.50 |

TABLE 1 – USArrest data set

En utilisant la distance euclidienne pour une CAH on obtient le dendrogramme de la figure 10. La variance inter-classe est représentée figure 11. D'après le dendrogramme et l'allure de la décroissance de la variance inter-classe, il semblerait que $K = 5$ soit le nombre de classes optimal pour cet exemple.

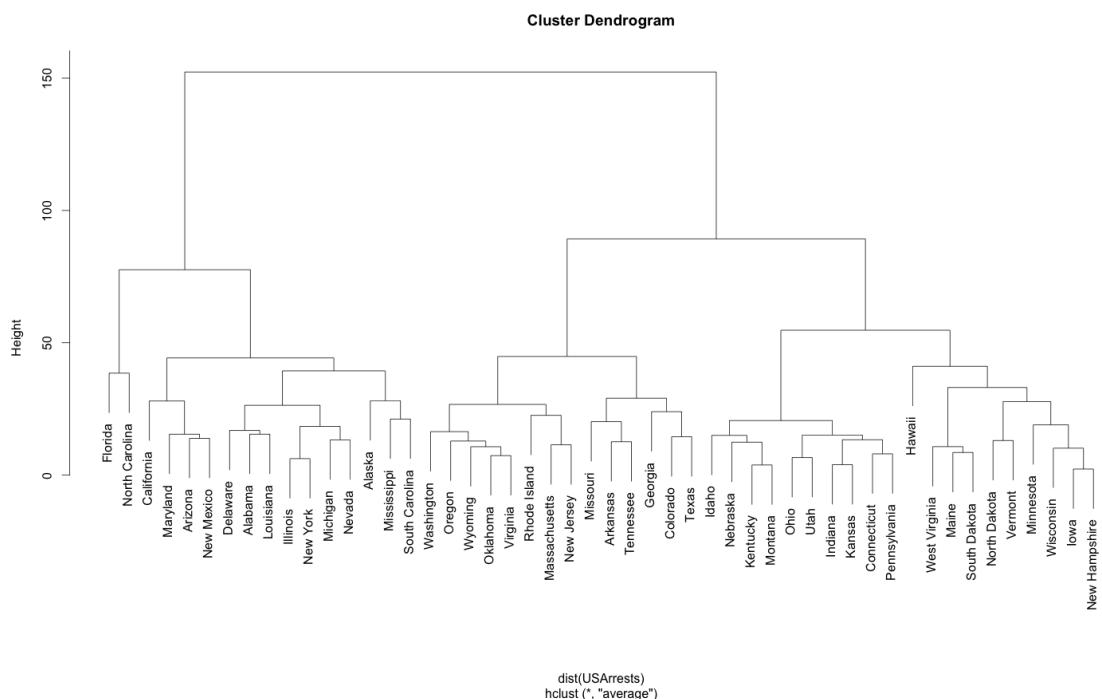


FIGURE 10 – Dendrogramme USArrest

Une fois le nombre de classe sélectionné l'arbre est coupé et fournit dans chaque sous-arbre la répartition des individus en classes. Ces classes peuvent ensuite être représentées dans les axes d'une analyse factorielle, comme par exemple figure 12 dans les coordonnées d'un MDS. Les classes sont représentées par des couleurs et permettent une visualisation de l'agrégation des individus.

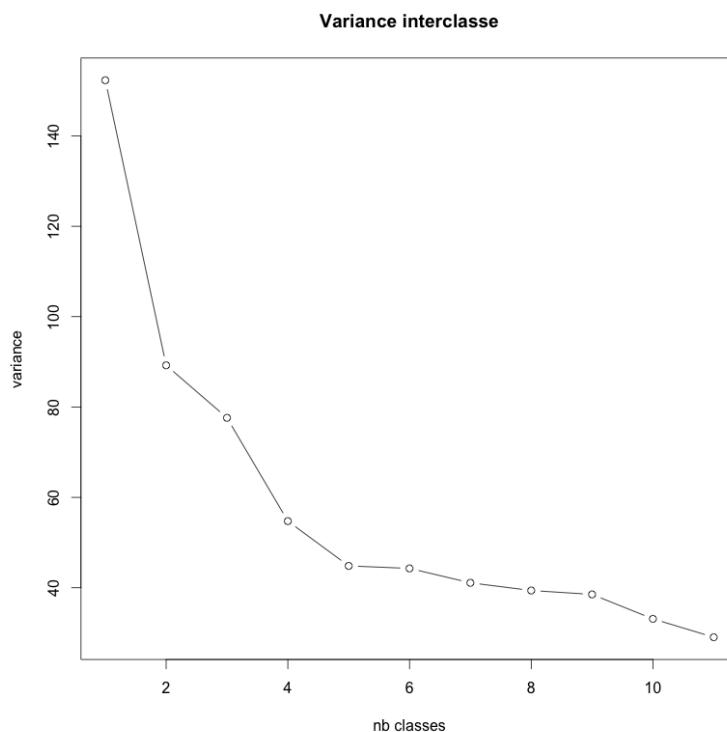


FIGURE 11 – Décroissance de la variance inter-classe pour USArrest

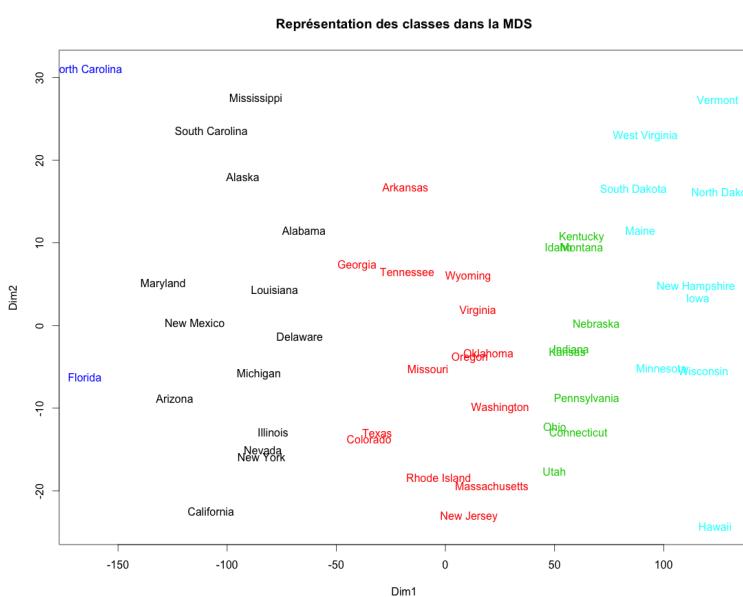


FIGURE 12 – Représentation des classes dans les coordonnées du MDS pour USArrest

4 Agrégation autour de centres mobiles

Contrairement à la classification hiérarchique, l'agrégation autour de centres mobiles est un partitionnement. Il n'y a pas de notion de hiérarchie dans la classe. La figure 13 illustre cette différence.

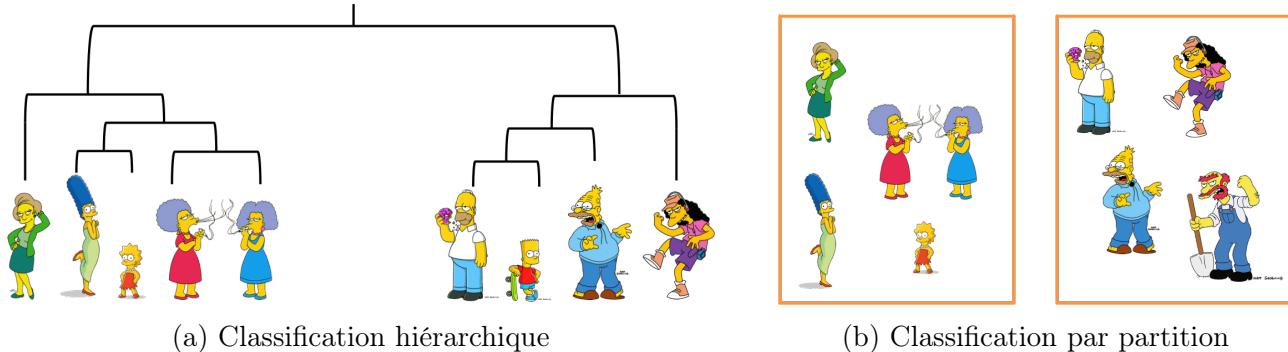


FIGURE 13 – Illustration des deux méthodes de classification pour l'exemple des *Simpsons*

La principale méthode est le *kmeans*, proposée en 1965 par Forgy [18].

4.1 Principe

On suppose qu'on observe X_1, \dots, X_n appartenant à un espace normé $(\mathcal{X}, \|\cdot\|)$. On recherche une partition des données en K groupes homogènes, K fixé *a priori*. Pour cela, on va rechercher k points $(c_1, \dots, c_k) \in \mathcal{X}^k$ qui seront les *centres* de ces groupes.

A l'initialisation, ces centres sont désignés aléatoirement. L'algorithme répète alors les deux opérations suivantes, de manière itérative, jusqu'à la convergence d'un critère :

1. chaque individu est assigné à la classe dont le centre est le plus proche au sens d'une métrique choisie
2. les centres des classes sont mis à jour

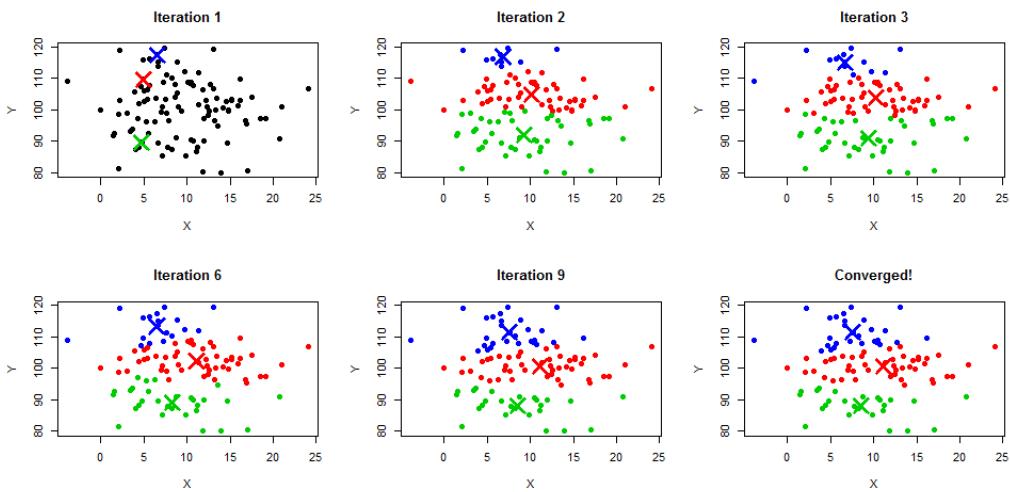
Une illustration du processus est donné figure 14. Une interface de clustering dynamique est proposée à l'adresse suivante : <http://shiny.rstudio.com/gallery/dynamic-clustering.html>.

4.2 Algorithme

ALGORITHME 2 :

Kmeans

- **Initialisation** Tirer au hasard, ou sélectionner pour des raisons extérieures à la méthode, K points dans l'espace des individus, appelés centres ou noyaux.
- **Itérer** les deux étapes suivantes, jusqu'à ce que le critère de variance inter-classe ne croisse plus de manière significative, ce qui signifie la stabilisation des classes.
 1. Allouer chaque individu au centre (donc à la classe) le plus proche au sens de la norme choisie ; on obtient ainsi, à chaque étape, une classification en K classes (ou moins si, finalement, une des classes devient vide).
 2. Calculer le centre de gravité de chaque classe : il devient le nouveau noyau. Si une classe s'est vidée, on peut éventuellement tirer aléatoirement un nouveau noyau complémentaire.

FIGURE 14 – Evolution des centres de classes lors des itérations d'un *kmeans*

4.3 Propriétés

4.3.1 Vitesse de convergence

L'étude de la méthode des plus proches voisins repose sur un outil fondamental de la théorie du clustering : la **distance de Wasserstein**, définie de la façon suivante :

Soient deux probabilités μ_1 et μ_2 sur \mathcal{X} ayant des moments d'ordre 2. On définit la distance de Wasserstein par :

$$W^2(\mu_1, \mu_2) = \inf_{X \sim \mu_1, Y \sim \mu_2} \mathbb{E}\|X - Y\|^2.$$

L'erreur commise lorsqu'on résume l'information contenue dans ces données par les seuls centres est définie par

$$E(c_1, \dots, c_k) = \frac{1}{n} \sum_{i=1}^n \min_{j=1, \dots, K} \|X_i - c_j\|^2. \quad (19)$$

Chercher la meilleure classification au sens donné par la norme $\|\cdot\|$ revient dès lors à minimiser l'erreur de classification (19) pour toutes les configurations de centres possibles. Ainsi choisir une classification revient à sélectionner un k -uplet de centres.

Les observations sont modélisées par des réalisations de variables aléatoires X_1, \dots, X_n de distribution μ . Définissons la mesure empirique des observations :

$$\mu_n := \frac{1}{n} \sum_{i=1}^n \delta_{X_i} \quad (20)$$

telle que pour tout ensemble mesurable $A \subset \mathcal{X}$,

$$\mu_n(A) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}_{X_i \in A}.$$

L'erreur de clustering des données dépend clairement de la loi des observations. Ainsi si on note $\hat{\mathbf{c}} = (\hat{c}_1, \dots, \hat{c}_K)$, on écrira

$$W(\hat{\mathbf{c}}, \mu_n) = \frac{1}{n} \sum_{i=1}^n \min_{j=1, \dots, K} \|X_i - \hat{c}_j\|^2,$$

l'erreur de clustering empirique.

A partir de maintenant on se restreint aux distributions de probabilité d'ordre, c'est-à-dire telles que

$$\int \|x\|^2 d\mu(x) < +\infty.$$

Dès lors, l'erreur moyenne de clustering s'écrira

$$W(\mathbf{c}, \mu) = \mathbb{E} \min_{j=1, \dots, K} \|X - c_j\|^2 = \int_{\mathcal{X}} \min_{j=1, \dots, K} \|x - c_j\|^2 d\mu(x). \quad (21)$$

Le risque optimal, appelé **oracle**, est alors défini comme réalisant la plus petite erreur moyenne, c'est-à-dire

$$W^*(\mu) = \inf_{\mathbf{c} \in \mathcal{X}^k} W(\mathbf{c}, \mu).$$

A partir des observations on cherche à construire une classification. i.e à sélectionner des centres minimisant l'erreur empirique

$$\mathbf{c}_n \in \arg \min_{\mathbf{c} \in \mathcal{X}^k} W(\mathbf{c}, \mu_n).$$

S'il existe un k -centre qui minimise cette erreur, noté $\mathbf{c}_n = (c_{n,1}, \dots, c_{n,k})$, on notera la partition de Voronoi associée $\mathcal{C}_n = (C_{n,1}, \dots, C_{n,K})$ définie par

$$C_{n,1} = \{x \in \mathcal{X}, \|x - c_{n,1}\| \leq \|x - c_j\|, j = 1, \dots, K\}$$

$$C_{n,l} = \{x \in \mathcal{X}, \|x - c_{n,l}\| \leq \|x - c_j\|, j = 1, \dots, K\} - \cup_{m=1}^{l-1} C_{n,m}.$$

Les clusters sont donnés, pour le l -ème par $C_{n,l} \cap \{X_1, \dots, X_n\}$.

La classification est dite consistante si l'erreur empirique converge vers l'erreur optimale au sens où

$$W(\mathbf{c}_n, \mu) \xrightarrow{n \rightarrow +\infty} W^*(\mu) \quad \text{p.s.}$$

La vitesse de convergence est alors définie comme la suite $r_n(k)$ décroissante vers 0 vérifiant

$$EW(\mathbf{c}_n, \mu) - W^*(\mu) \leq Cr_n(k),$$

pour C une constante indépendante de k et de n .

4.3.2 Optimum local

La solution obtenue est un optimum local : la répartition en classes obtenue après stabilisation du critère, n'est peut-être pas la meilleure. Tout particulièrement, la solution dépend du choix initial des noyaux.

Une possibilité est de relancer l'algorithme avec des centres de départ différents afin de détecter ces problèmes d'optima locaux et de trouver les classes présentes de manière stable dans la plupart des partitions obtenues. En pratique, on choisit comme centres initiaux les centres des classes obtenues par une CAH.

4.4 Variantes

4.4.1 Variante du *kmeans*

McQueen [8] propose que les centres des classes soient recalculés à chaque allocation d'un individu dans une classe. L'algorithme est plus efficace mais la solution dépend alors de l'ordre des individus dans le jeu de données.

Une variante [5] [9] consiste à remplacer chaque centre de classe par un noyau constitué d'éléments représentatifs de cette classe. Cela permet de corriger l'influence des valeurs extrêmes sur le calcul du barycentre.

4.4.2 Partitioning Around Medoids

L'algorithme PAM [14] permet de classifier les données de façon plus robuste. Les centres de classes sont moins influencés par les éventuels outliers.

Le noyau d'une classe, au lieu d'en être le barycentre, est un médoïde, c'est à dire l'observation qui minimise la moyenne des distances aux autres observations de la classe. Une différence notable avec *kmeans* est que le centre de classe n'est plus artificiel mais fait partie des données. Le PAM permet donc en particulier de classifier des matrices de dissimilarités.

En revanche, cet algorithme est limité par le nombre d'observations, puisqu'il y a un grand nombre de matrices de dissimilarités à stocker, et en temps de calcul ($\mathcal{O}(n^2)$). En effet, à chaque itération, un médoïde est mis en concurrence avec un autre individu tiré aléatoirement. Si l'échange améliore le critère, cet individu devient le nouveau médoïde.

5 Extension aux données fonctionnelles

Les données sont dites *fonctionnelles* lorsque les observations proviennent de la discréétisation d'une fonction f observée en une succession de valeurs $f(t_1), f(t_2), \dots, f(t_p)$.

On suppose que les fonctions observées sont de moyenne nulle. Si ce n'est pas le cas on les centre en retranchant la moyenne empirique

$$\bar{f} = \frac{1}{p} \sum_{j=1}^p f(t_j)$$

Considérer ces observations comme un simple vecteur de \mathbb{R}^p revient à perdre l'information que ces observations proviennent d'une seule et même fonction f .

On va donc chercher à projeter les fonctions sur des bases orthonormées afin de travailler non plus sur les observations d'origine mais sur leurs coefficients dans les bases de projection. Soit ϕ_λ une base orthonormée de $\mathbb{L}^2(\mathbb{E}; \langle \cdot, \cdot \rangle)$. La fonction f s'écrit alors :

$$f_n = \sum_{|\lambda| \leq n} a_\lambda \phi_\lambda \tag{22}$$

où $a_\lambda = \langle f; \phi_\lambda \rangle$.

Démonstration. —

$$\begin{aligned} \langle f; \phi_{\lambda_0} \rangle &= \left\langle \sum_{\lambda} a_{\lambda} \phi_{\lambda}; \phi_{\lambda_0} \right\rangle \\ &= \sum_{\lambda} a_{\lambda} \langle \phi_{\lambda}; \phi_{\lambda_0} \rangle \\ &= a_{\lambda_0} \end{aligned}$$

■

A noter que

$$\begin{aligned} a_{\lambda} &= \langle f; \phi_{\lambda} \rangle \\ &= \int f(t) \phi_{\lambda}(t) dt \end{aligned}$$

une estimation de a_{λ} est donnée par :

$$\hat{a}_{\lambda} = \sum_{i=1}^p f(t_i) \phi_{\lambda}(t_i) \left(\frac{t_i - t_{i-1}}{p} \right) \quad (23)$$

pour p grand.

Démonstration. — On cherche à démontrer que $\hat{a}_{\lambda} \xrightarrow[p \rightarrow +\infty]{} a_{\lambda}$.

Soit $\hat{f}_{m;\lambda} = \sum_{|\lambda| \leq m} \hat{a}_{\lambda} \phi_{\lambda}$. Exprimons l'erreur des moindres carrés $MSE = \mathbb{E} \left\| \hat{f}_{m;\lambda} - f \right\|^2$:

$$\mathbb{E} \left\| \hat{f}_{m;\lambda} - f \right\|^2 = \left\| \mathbb{E} \hat{f}_{m;\lambda} - f \right\|^2 + \mathbb{E} \left\| \hat{f}_{m;\lambda} - \mathbb{E} \hat{f}_{m;\lambda} \right\|^2 \quad (24)$$

$$= \underbrace{\left\| \sum_{|\lambda| \leq m} (\mathbb{E} \hat{a}_{\lambda} - a_{\lambda}) \phi_{\lambda} + \sum_{|\lambda| > m} a_{\lambda} \phi_{\lambda} \right\|^2}_{(1)} + \underbrace{\mathbb{E} \left\| \sum_{|\lambda| > m} (\hat{a}_{\lambda} - \mathbb{E} \hat{a}_{\lambda,m}) \phi_{\lambda} \right\|^2}_{(2)} \quad (25)$$

Puisque ϕ_{λ} est une base orthonormée de $\mathbb{L}^2(\mathbb{E}; \langle \cdot, \cdot \rangle)$ et que les a_{λ} sont i.i.d., la partie (2) de l'expression (25) s'écrit

$$\mathbb{E} \left\| \sum_{|\lambda| > m} (\hat{a}_{\lambda} - \mathbb{E} \hat{a}_{\lambda,m}) \phi_{\lambda} \right\|^2 = \mathbb{E} \sum_{|\lambda| > m} |\hat{a}_{\lambda} - \mathbb{E} \hat{a}_{\lambda,m}|^2 \quad (26)$$

$$= \sum_{|\lambda| > m} \mathbb{E} |\hat{a}_{\lambda} - \mathbb{E} \hat{a}_{\lambda,m}|^2 \quad (27)$$

$$= mVar(\hat{a}_{\lambda}) \quad (28)$$

De même, la partie (1) de l'expression (25) s'écrit :

$$\left\| \sum_{|\lambda| \leq m} (\mathbb{E} \hat{a}_{\lambda} - a_{\lambda}) \phi_{\lambda} + \sum_{|\lambda| > m} a_{\lambda} \phi_{\lambda} \right\|^2 = \underbrace{\sum_{|\lambda| \leq m} (\mathbb{E} \hat{a}_{\lambda} - a_{\lambda})^2}_{(1)} + \underbrace{\sum_{|\lambda| > m} a_{\lambda}^2}_{(2)} \quad (29)$$

Dès lors, l'erreur des moindres carrés s'exprime par un terme de variance (28), un biais d'estimation (29).(1) et un biais de projection (29).(2) :

$$MSE = mVar(\hat{a}_{\lambda}) + |m| \text{biais}_{(\text{estimation } \hat{a}_{\lambda} \rightarrow a_{\lambda})}^2 + \sum_{|\lambda| > m} a_{\lambda}^2 \quad (30)$$

Si f est une densité, e.G. $X_1, \dots, X_n \sim f$ alors

$$\hat{a}_\lambda = \frac{1}{n} \sum_{i=1}^n \phi_\lambda(X_i)$$

Donc

$$\begin{aligned}\mathbb{E}\hat{a}_\lambda &= \mathbb{E}\phi_\lambda(X) \\ &= \int \phi_\lambda(x)f(x)dx \\ &= \langle \phi_\lambda; f \rangle \\ &= a_\lambda\end{aligned}$$

Sinon les observations s'écrivent

$$Y_i = f(t_i) + \epsilon_i$$

où $\mathbb{E}\epsilon = 0$. Dans ce cas,

$$\hat{a}_\lambda = \frac{1}{n} \sum_{i=1}^n Y_i \phi_\lambda(t_i)$$

Donc

$$\mathbb{E}\hat{a}_\lambda = \frac{1}{n} \sum_{i=1}^n f(t_i) \phi_\lambda(t_i) \xrightarrow[n \rightarrow +\infty]{} a_\lambda$$

Le biais d'estimation (29).(1) est donc un biais déterministe, il est en pratique souvent ignoré.

Le biais de projection $\sum_{|\lambda|>m} a_\lambda^2$ (29).(2) quant à lui est un reste de série convergente. En effet, puisque $f \in \mathbb{L}^2$ alors

$$\|f\|^2 = \sum_\lambda a_\lambda^2 < +\infty$$

Donc

$$\sum_{|\lambda|>m} a_\lambda^2 \xrightarrow[m \rightarrow +\infty]{} 0$$

L'erreur des moindres carrés MSE (30) est donc composé d'un terme de variance qui croît avec m et d'un terme de biais qui décroît avec m . ■

6 Problématique de la moyenne et choix de la distance

6.1 Une distance correspondant à la structure géométrique

La définition d'une distance entre deux fonctions est une étape complexe mais fondamentale du clustering de courbes. Une illustration de la différence de résultats est donnée figure 15. Les données brutes sont représentées fig.15a. La moyenne empirique est donnée fig.15b par $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$. Remarquons que cette moyenne ne semble pas refléter la structure des données. Une moyenne basée sur la structure, représentée en rouge fig.15c semble plus représentative.

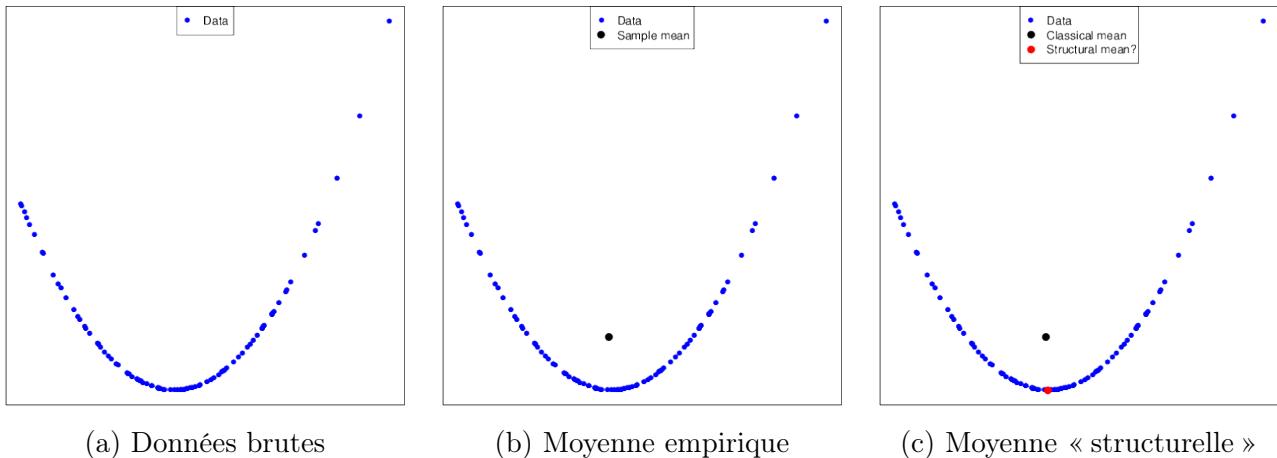


FIGURE 15 – Illustration de la différence entre la moyenne empirique et la moyenne en terme de structure

La moyenne empirique s'écrit :

$$\bar{X} = \arg \min_{\mu \in \mathbb{R}^p} \sum_{i=1}^n d^2(X_i, \mu) \quad (31)$$

avec d la distance euclidienne. Dans cet exemple, il serait probablement plus judicieux d'utiliser une distance de type géodésique, comme représenté figure 16

$$\bar{X}^\alpha = \arg \min_{\mu \in \mathcal{M}} \sum_{i=1}^n \delta^\alpha(X_i, \mu) \quad (32)$$

La première étape de la plupart des méthodes de ce type est de construire un graphe de voisinage en reliant chaque point de données à un nombre fixe de ses plus proches voisins ou à tous les points dans un certain rayon du point donné.

Les méthodes locales (LLE [20], Hessian LLE [4], LEM [15]) cherchent à préserver les relations locales entre les points en apprenant un ensemble de poids entre chaque point et ses voisins.

Les méthodes globales(Isomap [10] plongement semi-défini [13],plongement préservant la structure [3]) cherchent à préserver les relations locales et globales entre tous les points de données.

Les deux catégories de procédés cherchent une représentation des données en faible dimension à partir des vecteurs propres d'une matrice liée au poids d'apprentissage entre les paires de points.

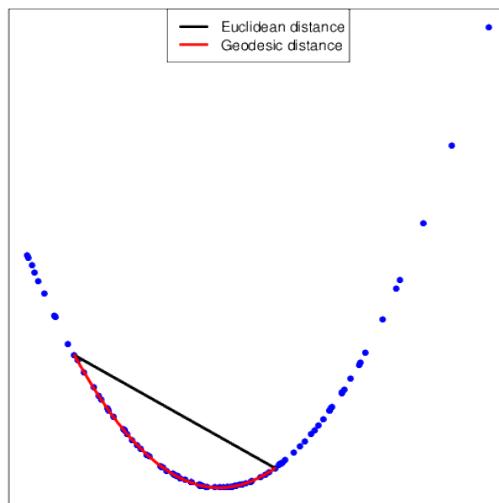


FIGURE 16 – Distance Euclidienne VS Géodésique

6.2 Décalages de phase et d'amplitude

Une problématique récurrente de l'analyse de données fonctionnelle est la présence de décalages de phase (sur l'axe des abscisses) et d'amplitude (sur l'axe des ordonnées) entre les observations. Là encore, une distance mal choisie peut masquer la structure des données.

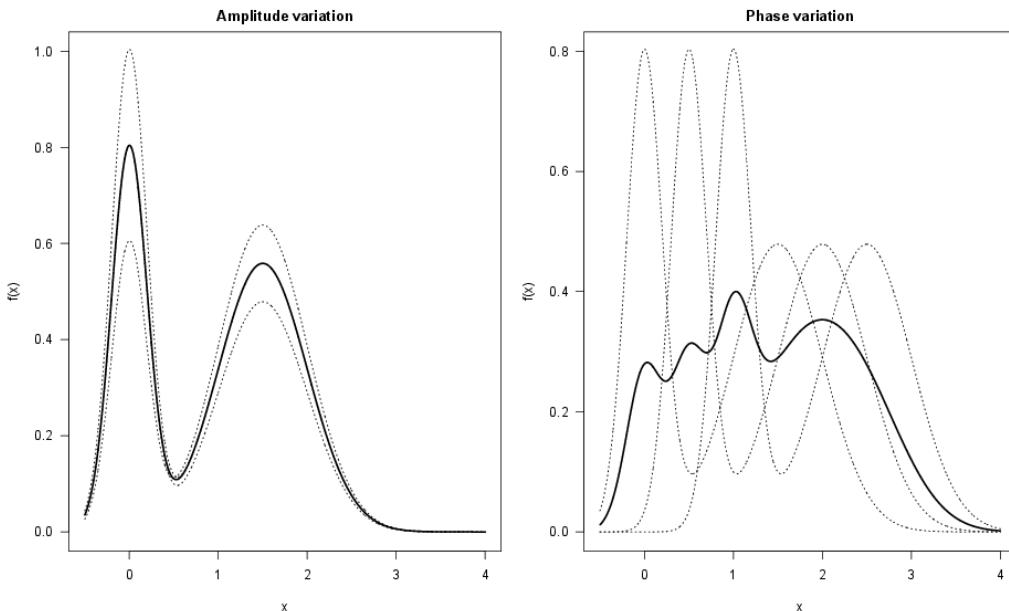


FIGURE 17 – Décalages d'amplitude et de phase et impact sur la moyenne (en trait plein)

Les décalages de phase sont ceux qui masquent le plus la structure des données. un exemple est donné figure 17; dans le cas d'une variation d'amplitude, la moyenne est représentative de la forme des données mais en cas de variation de phase, moyenner les données fait perdre de l'information sur la structure des données.

Un exemple concret commun est le cas du trafic routier, illustré figure 18. Cet exemple est issu de la thèse de G. Allain [6] co-conduite par l'[Université Toulouse III](#) et la société [Media-mobile](#).

Sur ce graphe sont représentées des vitesses observées sur une portion de périphérique le mardi.

On observe un mélange de décalages en phase et en amplitude. La moyenne, représentée en trait rouge plein, fait ressortir un ralentissement autour de 9h le matin, un léger ralentissement vers 12h30 et un ralentissement autour de 1h, moins important que celui de 9h. Si ces informations semblent cohérentes, elles ne sont pourtant pas représentatives de la réalité du phénomène physique sous-jacent.

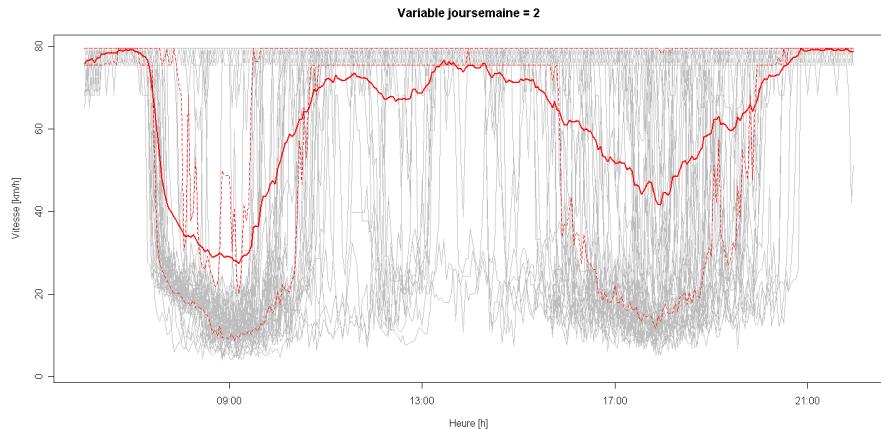


FIGURE 18 – Vitesses sur un parcours : la moyenne est-elle représentative du phénomène ?

Le premier point est que la moyenne dégrade fortement la structure d'un bouchon. En effet, une analyse plus fine, par exemple en appliquant un recalage aux données, permet de voir qu'un bouchon correspond à une baisse beaucoup plus brutale de la vitesse, une vitesse constante assez basse puis une augmentation tout aussi rapide de la vitesse, comme illustré figure 19.

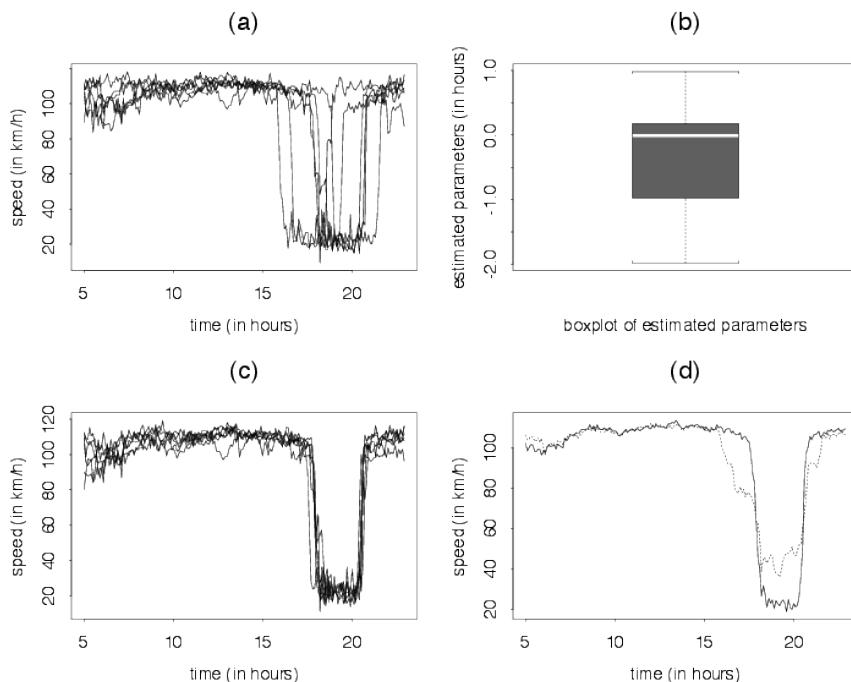


FIGURE 19 – Structure d'un bouchon VS vitesse moyenne

D'autre part, la moyenne, en sur-lissant les phénomènes, ne permet pas une bonne estimation du temps de parcours, ni une bonne compréhension des phénomènes en jeu. Un clustering

des données fonctionnelles va faire apparaître des informations sur le trafic en fonction du jour de la semaine, comme illustré figure 20.

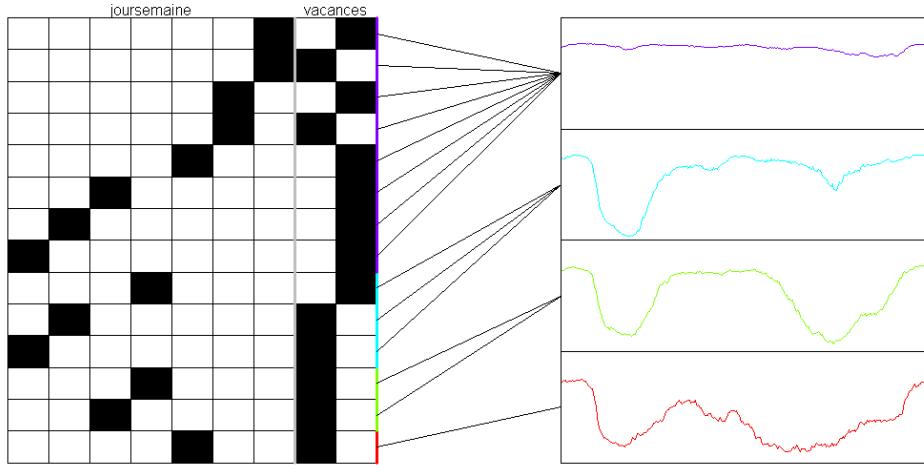


FIGURE 20 – Représentants de classes et information

7 Solutions

7.1 Dynamic Time Warping

Puisque les données présentent des variations de phase et d'amplitude, une possibilité est bien sûr de trouver une fonction de recalage de type

$$g(x) = \alpha_1 f(\alpha_2 x + \alpha_3) + \alpha_4 \quad (33)$$

Cette fonction, délicate à obtenir, présuppose que le décalage entre les courbes soit constant en tout point, ce qui est rarement le cas.

La déformation temporelle dynamique, ou *Dynamic Time Warping* (DTW) permet de recalier une courbe sur une autre en déformant l'espace de temps. Autrement dit, le décalage n'est pas considéré comme constant et plusieurs points d'une courbe peuvent être appairés au même point de la courbe objectif. Un exemple est donné figure 21.

Formellement, soient deux séquences $X = (x_1, \dots, x_N)$ et $Y = (y_1, \dots, y_M)$ que l'on souhaite aligner.

La première étape est de créer une matrice de distances locales (*warping matrix* ou *cost matrix*) $\mathcal{M} \in \mathbb{R}^{n \times m}$ telle que

$$\mathcal{M}_{ij} = d(x_i, y_j) \quad (34)$$

où $d : \Omega \times \Omega \rightarrow \mathbb{R}_+$ est en général une fonction de dissimilarité.

Pour trouver le meilleur recalage entre ces deux séquences, on définit un chemin dans la matrice qui minimise la distance totale cumulée entre X et Y . : un chemin de recalage (*warping path*) est une séquence $p = (p_1, \dots, p_L)$ où $p_L = (n_l, m_l)$ tels que :

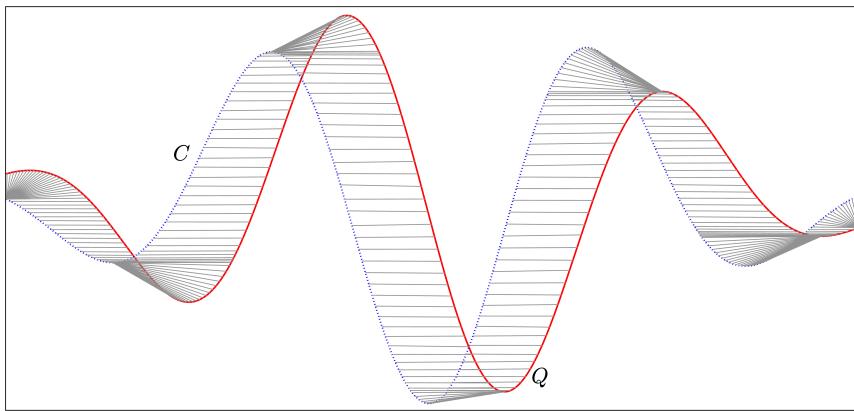


FIGURE 21 – Recalage par Dynamic Time Warping

1. $p_1 = (1, 1)$ et $p_L = (N, M)$
2. $n_1 \leq n_2 \leq \dots \leq n_L$ et $m_1 \leq m_2 \leq \dots \leq m_M$
3. $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}, \forall l \in [1, L - 1]$

Ce chemin, représenté en exemple figure 22, assigne chaque élément x_{n_l} de X à un élément y_{m_l} de Y . Le coût total du chemin est donné par la somme des éléments de la matrice (34) :

$$c_p(X, Y) = \sum_{l=1}^L \mathcal{M}_{n_l, m_l} \quad (35)$$

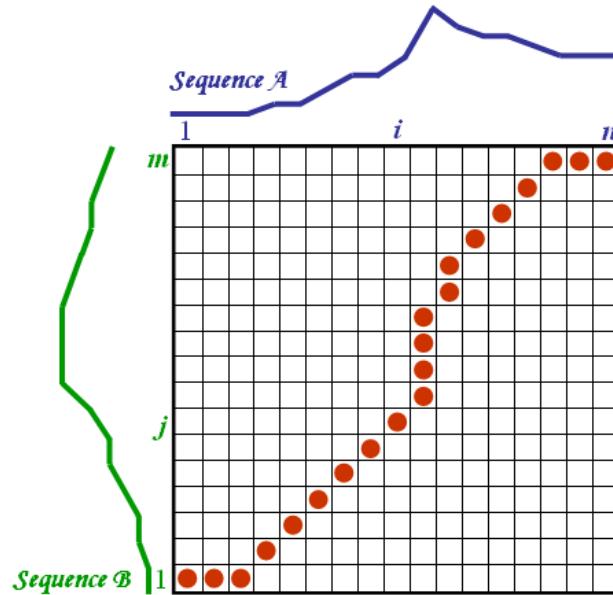


FIGURE 22 – Warping path

Le chemin optimal est donc donné par

$$p^* = DTW(X, Y) = \min \left(\sqrt{\sum_{l=1}^L d(x_{n_l}, y_{m_l})} \right) \quad (36)$$

Calculer tous les chemins possibles dans la matrice \mathcal{M} aurait une complexité exponentielle en N et en M . Un algorithme de programmation dynamique permet de réduire la complexité en $O(NM)$. En effet, considérons la matrice des coûts cumulés D telle que

$$D(n, m) = DTW((x_1, \dots, x_n), (y_1, \dots, y_m)) \quad (37)$$

Alors il est facile de prouver que :

$$D(N, M) = DTM(X, Y) \quad (38)$$

$$D(n, 1) = \sum_{k=1}^n d(x_k, y_1) \quad (39)$$

$$D(1, m) = \sum_{k=1}^m d(x_1, y_k) \quad (40)$$

$$D(n, m) = \min\{D(n - 1, m - 1), D(n - 1, m), D(n, m - 1)\} + d(x_n, y_m) \quad (41)$$

La matrice D se calcule donc de façon récursive. On lui définit une ligne et une colonne de plus pour l'initialisation de l'algorithme, de sorte que :

$$D(0, 0) = 0 \quad (42)$$

$$D(n, 0) = D(0, m) = +\infty \quad (43)$$

D se calcule alors par récurrence en commençant par $pL = (N, M)$

ALGORITHME 3 : Calcul optimal du meilleur chemin de recalage

- Initialisation : $pL = (N, M)$
- Récurrence : p_l une fois calculé, on définit p_{l-1} par (41) :

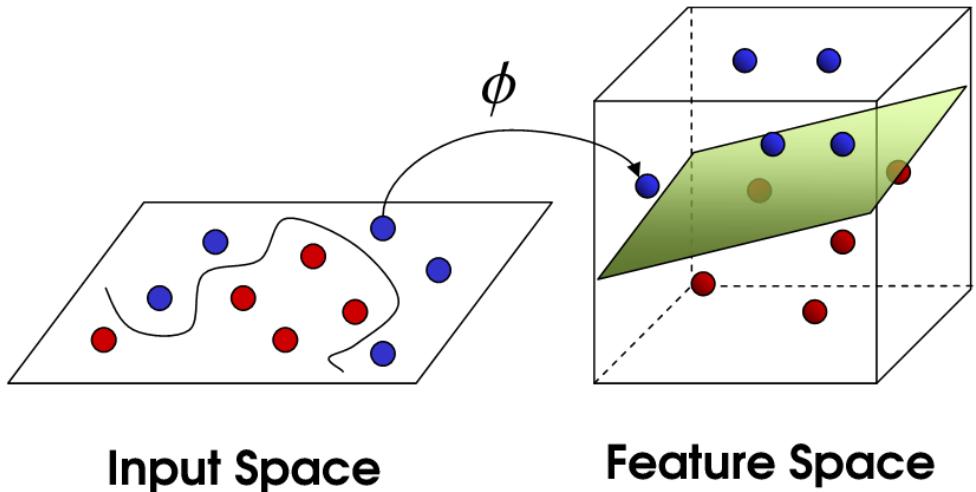
$$p_{l-1} = \begin{cases} (1, m - 1) & \text{si } n = 1 \\ (n - 1, 1) & \text{si } m = 1 \\ \arg \min(D(n - 1, m - 1), D(n - 1, m), D(n, m - 1)) & \text{sinon} \end{cases} \quad (44)$$

7.2 Astuce du noyau ou *kernel trick*

Dans les configurations où un partitionnement avec un classifieur linéaire est impossible, la définition d'un classifieur non-linéaire s'avère très complexe dans l'espace initial de représentation des données. L'astuce du noyau ou *kernel trick* consiste à utiliser une fonction à noyaux pour transformer l'espace initial en un espace de dimension supérieur, dans lequel un classifieur linéaire peut être utilisé. La figure 23 en illustre le fonctionnement : un classifieur linéaire dans l'espace de grande dimension est équivalent à un classifieur non-linéaire dans l'espace d'origine.

7.3 *Kmeans* à noyaux

La méthode du *kmeans* à noyaux (*kernel kmeans*) consiste à appliquer le *kernel trick* avant de lancer un clustering par *kmeans*. Formellement, la méthode des *kmeans* à noyaux est une généralisation de la méthode des *kmeans* qui consiste tout d'abord à utiliser une application non linéaire qui envoie les données dans un espace de forme (*feature space*), puis à classifier les données par une méthodologie *kmeans* usuelle dans ce nouvel espace, mieux adapté à la classification.

FIGURE 23 – Illustration du *kernel trick*

On considère $X_1, \dots, X_n \in \mathcal{X}^n$ avec $X_i \in \mathbb{R}^d$. Choisissons

$$\phi : \mathcal{X} \rightarrow \mathcal{Y} \quad (45)$$

une application non linéaire qui envoie les données dans un espace de dimension supérieure et définissons pour noyau :

$$K_{i,j} = K(X_i, X_j) = \phi(X_i)^T \phi(X_j) \quad (46)$$

Soit une partition $\{A_1, \dots, A_k\}$ associée aux centres $\mathbf{c} = \{c_1, \dots, c_k\} \in \mathcal{X}^k$ obtenue à l'étape k . A l'étape $k+1$, on remplace les centres par les barycentres des classes obtenues à l'étape k . L'objectif du *kmeans* s'écrit donc :

$$\hat{c}_j = \arg \min_{\{x \in \mathcal{X}\}} \sum_{i=1}^n \|X_i - x\|^2 \mathbb{I}_{X_i \in A_j}. \quad (47)$$

Cette minimisation admet pour solution

$$\hat{c}_j = \frac{1}{n_j} \sum_{\{i, X_i \in A_j\}} X_i, \quad (48)$$

avec $n_j = |A_j|$.

Dans l'espace \mathcal{Y} , l'erreur de classification pour une partition de \mathcal{X} en $\{C_1, \dots, C_k\}$ s'écrit

$$E(m_1, \dots, m_k) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k \|\phi(X_i) - m_j\|^2 \mathbb{I}_{X_i \in C_k}, \quad (49)$$

avec

$$m_j = \frac{\sum_{i=1}^n \phi(X_i) \mathbb{I}_{X_i \in C_j}}{\sum_{i=1}^n \mathbb{I}_{X_i \in C_j}} \quad (50)$$

Cette expression est solution du problème de minimisation qui généralise l'expression (48) au cadre d'un espace à noyaux. Posons

$$N_j = \sum_{i=1}^n \mathbb{I}_{X_i \in C_j} \quad (51)$$

L'algorithme des *kmeans* à noyaux s'écrit alors

ALGORITHME 4 :

Kernel Kmeans

- Initialisation : K clusters C_1, \dots, C_K
- Pour tous les points X_i , $i = 1, \dots, n$, pour tous les clusters $j = 1, \dots, K$, on calcule

$$\|\phi(X_i) - m_j\|^2$$

et on trouve l'allocation de chaque observation :

$$\hat{q}(X_i) = \arg \min_{j=1, \dots, k} \|\phi(X_i) - m_j\|^2$$

- On change la partition en modifiant tous les C_j , $j = 1, \dots, K$ en

$$C_j := \{x_i, \hat{q}(X_i) = j\}.$$

- Si les clusters n'évoluent plus, la solution est obtenue par les C_j ainsi obtenus et les barycentres m_j , sinon on recommence à l'étape 1.

On remarque que le calcul de l'erreur (49) s'écrit

$$\begin{aligned} \|\phi(X_i) - m_j\|^2 &= K_{n,n} - 2 \frac{\sum_{i=1}^n K_{n,j} \mathbb{I}_{X_i \in C_j}}{\sum_{i=1}^n \mathbb{I}_{X_i \in C_j}} \\ &\quad + \frac{\sum_{i_1=1}^n \sum_{i_2=1}^n \mathbb{I}_{X_{i_1} \in C_j} \mathbb{I}_{X_{i_2} \in C_j} K_{i_1, i_2}}{\sum_{i_1=1}^n \sum_{i_2=1}^n \mathbb{I}_{X_{i_1} \in C_j} \mathbb{I}_{X_{i_2} \in C_j}}. \end{aligned}$$

Ainsi seule la connaissance des valeurs $K_{i,j}$ pour $(i, j) \in (1, n)^2$ est nécessaire, comme dans toutes les méthodes à noyaux.

L'algorithme converge dès que le noyau est strictement semi-défini positif. La preuve est laissée à la curiosité du lecteur, qui s'inspirera des références [7] et [2]. La complexité de l'algorithme est en $O(n^2\tau)$ où τ est le nombre d'itérations pour obtenir la convergence.

En pratique, on s'intéressera aux noyaux suivants :

- Noyau polynomial :

$$K(x, y) = (\text{scale} \cdot \langle x, y \rangle + \text{offset})^{\text{degree}} \quad (52)$$

- Noyau Gaussien :

$$K(x, y) = e^{-\sigma \|x-y\|^2} \quad (53)$$

- Noyau Sigmoïde :

$$K(x, y) = \tanh(\text{scale} \cdot \langle x, y \rangle + \text{offset}) \quad (54)$$

- Noyau ANOVA :

$$K(x, y) = \left(\sum_{i=1}^n e^{-\sigma (x^i - y^i)^2} \right)^d \quad (55)$$

où x^i est la i-ème composante de x .

7.4 Spectral clustering

Soit X la matrice $m \times n$ des observations à classifier : $X = (x_1, \dots, x_n)$, $x_i \in \mathbb{R}^m \forall i = 1, \dots, n$.

La première étape consiste à créer une matrice de similarité ou matrice d'adjacence $A \in \mathbb{R}^{n \times n}$ en choisissant un noyau de similarité k tel que

$$\begin{aligned} A_{ij} &= k(x_i, x_j) \geq 0 \\ A_{ij} &= A_{ji} \end{aligned}$$

L'espace spectral est alors construit de la manière suivante : soit $D \in \mathbb{R}^{k \times k}$ la matrice diagonale telle que

$$D_{ii} = \sqrt{\sum_{i=1}^n A_{ii}}$$

On définit la matrice Laplacienne L par

$$L = DAD^T \quad (56)$$

On définit la matrice V dont les colonnes constituent les k premiers vecteurs propres orthogonaux de L :

$$V = (v_1, \dots, v_k) \in \mathbb{R}^{n \times k}$$

On normalise la matrice V :

$$W_{ij} = \frac{V_{ij}}{\sqrt{\sum_{j=1}^k V_{ij}^2}} \quad (57)$$

Chaque ligne de W est un point de \mathbb{R}^k . On applique aux lignes de W une méthodes de clustering, par exemple un *kmeans*.

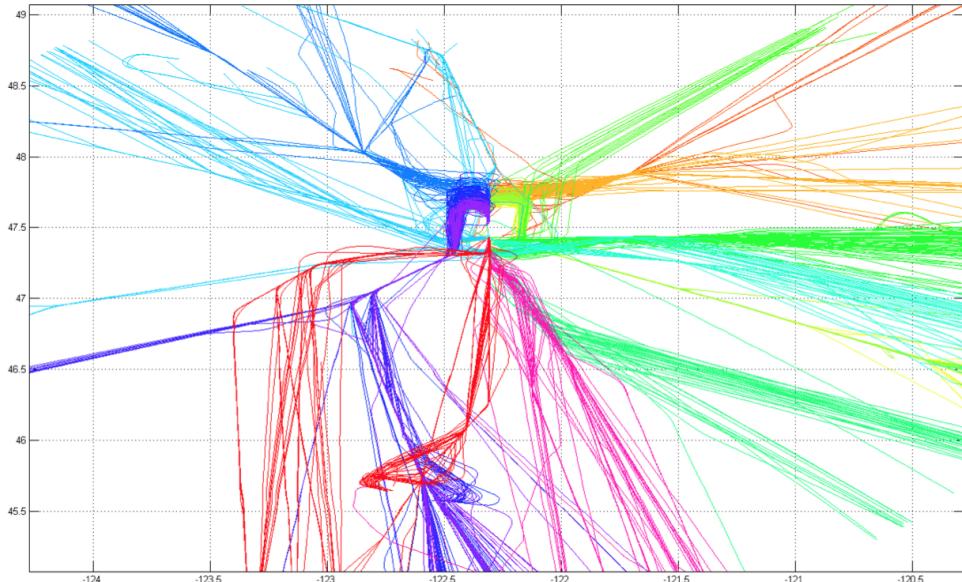


FIGURE 24 – Exemple d'utilisation du clustering spectral sur des flux de vols d'avion [16]

Références

- [1] Ng A.Y., Jordan M.I., and Weiss Y. *On spectral Clustreing : Analysis and an algorithm.* cs.berkeley.edu, 2001.
- [2] Scholkopf B., Smola A.J., and Muller K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10 :1299–1319, 1998.
- [3] Shaw B. and Jebara T. Structure preserving embedding. *International Conference on Machine Learning*, 2009.
- [4] Donoho D. and Grimes C. Hessian eigenmaps : Locally linear embedding techniques for highdimensional data. *National Academy of Sciences*, 100(10) :5591–5596, 2003.
- [5] Diday E. The dynamic clusters method in non-hierarchical clustering. *International Journal of Computer and Information Sciences*, 1 :61–88, 1973.
- [6] Allain G. *Prévision et analyse du trafic routier par des méthodes statistiques*. PhD thesis, Université Toulouse III, 2008.
- [7] Dhillon I.S., Guan Y., and Kulis B. Kernel k-means, spectral clustering and normalized cuts. *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, pages 551–556, 2004.
- [8] McQueen J. Some methods for classification and analysis of multivariate observations. *5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [9] Hartigan J.A. and Wong M.A. Algorithm as 136 : a k-means clustering algorithm. *Applied Statistics*, 28 :100–108, 1979.
- [10] Tenenbaum J.B., De Silva V., and Langford J.C. A global geometric framework for nonlinear dimensionality reduction. *Science*, page 585–591, 2002.
- [11] Ramsay J.O. and Silverman B.W. *Functional Data Analysis*. Springer Series in Statistics, 2005.
- [12] Ramsay J.O. and LI X. Curve registration. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 60 :351–363, 1998.
- [13] Weinberger K.Q. and Saul L. Unsupervised learning of image manifolds by semidefinite programming. *IEEE Conference on Computer Vision and Pattern Recognition*, page 988–955, 2004.
- [14] Kaufman L. and Rousseeuw P.J. Finding groups in data – an introduction to cluster analysis. *John Wiley and Sons*, 1990.
- [15] Belkin M. and Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Neural Information Processing Systems*, 290(5500) :2319–2323, 2000.
- [16] Enriquez M. and Kurcz C. A simple and robust flow detection algorithm based on spectral clustering. *In ICRAT Conference*, 2012.
- [17] Müller M. *Information Retrieval for Music and Motion*. Springer, 2007.
- [18] Forgy R. Cluster analysis of multivariate data : Efficiency versus interpretability of classification. *Biometrics*, 21 :768–769, 1965.
- [19] Tibshirani R., Walther G., and Hastie T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 2 :411–423, 2001.
- [20] Roweis S. and Saul L. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500) :2323–2326, 2000.
- [21] Vantini S. On the definition of phase and amplitude variability in functional data analysis. *TEST*, 21(4) :676–696, 2011.

- [22] Huang Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov.*, 3 :283–304, 1998.