UNIVERSITÉ DU QUÉBEC À MONTRÉAL

ASSOCIATION GÉNÉTIQUE MESURÉE À L'AIDE D'UNE MÉTHODE DE CLASSIFICATION DE DONNÉES FONCTIONNELLES

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN MATHÉMATIQUES

PAR

MOM RAVY IENG

DÉCEMBRE 2021

REMERCIEMENTS

J'aimerais tout d'abord remercier ma directrice de recherche Marie-Hélène Descary et mon co-directeur de recherche Fabrice Larribe, pour leur support, leur patience, leur générosité, leurs conseils et encouragements malgré les difficultés rencontrés lors de la rédaction de ce mémoire. En particulier Marie-Hélène, pour ses nombreuses relectures et ses commentaires. J'ai beaucoup appris de vous deux.

Je voudrais adresser un remerciement à ma famille et mes amis, tout particulier à mes parents. Sans leur support et leur amour inconditionnel, je n'aurais jamais été capable de me rendre aussi loin.

Je tient également à remercier mon frère et ma soeur pour leurs encouragements.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

RÉSUMÉ

Ce mémoire propose une nouvelle méthode d'association génétique quand le caractère génétique d'intérêt est une fonction. À cette fin, nous présentons une méthode non-paramétrique de classification supervisée pour des données fonctionnelles, développée par Ferraty et Vieu (2006). Dans un premier temps, nous présentons quelques concepts de l'analyse de données fonctionnelles et passons en revue le problème de classification dans le cas fonctionnel, la méthode des K plus proches voisins et les fonctions noyaux. Nous introduisons trois familles de semi-métriques qui agissent comme une mesure de proximité entre deux fonctions. Par la suite, nous étudions la performance de la méthode non-paramétrique de classification supervisée à l'aide d'une étude de simulation. Finalement, nous présentons l'approche consistant à utiliser la méthode de classification non-paramétrique aux données génétiques d'une plante à fleurs (appelée arabette des dames) de Moore *et al.* (2013) afin d'étudier l'association entre le phénotype et le génotype. On compare les résultats obtenus avec les résultats de Kwak *et al.* (2016) obtenus avec une approche par vraisemblance. On constate que la méthode proposée donne des résultats différents de l'approche par vraisemblance.

Mots-clés : données fonctionnelles, semi-métriques, classification supervisée, SNP

INTRODUCTION

Grâce aux développements technologiques, on a maintenant accès à des instruments de mesure puissants qui permettent quasiment d'enregistrer des données en continu sur un intervalle de temps. On peut donc s'intéresser à des observations qui prennent par exemple, la forme de courbes, d'images ou de surfaces ; l'analyse de données fonctionnelles est une branche de la statistique permettant d'analyser de telles données.

L'objectif principal de ce mémoire est de proposer l'utilisation d'une méthode de classification supervisée comme outil d'association génétique. La méthode de classification que nous allons utiliser est une méthode non-paramétrique qui est basée sur l'estimateur de Nadaraya-Watson et developpée par Ferraty et Vieu (2006). Dans ce mémoire, nous allons expliquer comment le taux de classification à différents marqueurs génétiques peut être utilisé pour mesurer la force d'une association génétique, puis nous allons appliquer la méthode aux données génétiques d'une plante à fleurs de Moore *et al.* (2013) pour pouvoir décrire l'association entre le phénotype et le génotype. Ces études d'association ont pour but d'identifier les gènes sur plusieurs chromosomes influençant des caractères d'intérêt.

Le premier chapitre de ce mémoire contient une introduction des concepts de base en analyse de données fonctionnelles qui seront utilisés dans les chapitres subséquents. Le deuxième chapitre introduit les concepts de classification, de semi-métriques, l'estimateur des K plus proches voisins, l'estimation par noyau ainsi que la méthode de classification non-paramétrique pour des données fonctionnelles. Au chapitre trois, nous présentons une étude de simulation afin d'évaluer

la performance de la méthode non-paramétrique de classification supervisé. Nous terminons ce mémoire, avec le quatrième chapitre, par la présentation de notre nouvelle méthode, puis nous présentons les résultats obtenus à l'aide de cette méthode, enfin nous les comparons à ceux de Kwak *et al.* (2016).

CHAPITRE I

INTRODUCTION À L'ANALYSE DE DONNÉES FONCTIONNELLES

1.1 Que sont les données fonctionnelles ?

Les données fonctionnelles apparaissent naturellement dans de nombreux domaines des sciences appliquées (médecine, économétrie, biométrie, etc.) dans lesquels les données collectées sont des courbes. Considérons une situation où une variable aléatoire peut être observée à plusieurs moments différents dans un intervalle de temps (t_{min}, t_{max}). Cette variable peut être exprimée par la famille aléatoire $\left\{X(t_m)\right\}_{m=1,...,M}$. Puisque la grille temporelle $t_1, ..., t_M$ peut être très fine, c'est-à-dire que les instants consécutifs auxquels une variable aléatoire est observée sont très proches les uns des autres, il est possible de considérer une variable comme la famille continue $\mathcal{X} = \Big\{X(t), t \in (t_{min}, t_{max})\Big\}$. L'analyse de données fonctionnelles consiste en l'analyse d'un ensemble de réalisations $x_1, ..., x_n$ d'un échantillon aléatoire $\mathcal{X}_1, ..., \mathcal{X}_n$, où $\mathcal{X}_i$ est distribuée comme une variable aléatoire fonctionnelle $\mathcal{X}$. Dans Ferraty et Vieu (2006), on définit une variable fonctionnelle comme étant une variable aléatoire $\mathcal{X}$ prenant des valeurs dans un espace fonctionnel de dimension infinie. Soit $T = [a, b] \subset \mathbb{R}$; nous allons travailler avec des variables fonctionnelles qui sont des éléments de $L^2(T) = \Big\{f : T \to \mathbb{R} \mid \int_T f(t)^2 dt < \infty \Big\}$. De façon intuitive, une donnée fonctionnelle est une version lisse d'une séquence d'observations discrètes échantillonnées à partir d'un processus continu. Le domaine est

généralement le temps, mais il pourrait être autre chose comme l'espace. Prenons le temps comme domaine, les données fonctionnelles sont alors enregistrées sur une grille de temps ordonnés et la grille de temps peut différer pour chaque variable observée. Dépendamment de l'instrument utilisé pour la mesure du temps, la grille temporelle peut être régulière (intervalle de temps équidistant) ou pas. En pratique, il n'est pas possible d'observer une fonction $\mathcal{X}$ dans son ensemble. En effet, pour cela, on devrait disposer d'instruments de mesure avec une vitesse d'enregistrement infinie, et être en mesure de sauvegarder un nombre infini de valeurs. Ainsi, ce n'est pas possible d'observer $\{\mathcal{X}(t), t \in T\}$, mais on peut observer $\big(\mathcal{X}(t_1), ..., \mathcal{X}(t_n)\big)$. En analyse de données fonctionnelles, on suppose que les courbes $x_1, ..., x_n$ sont lisses. Il est donc possible d'estimer les données fonctionnelles à partir d'un ensemble de données discrétisées par des méthodes de lissage. Une des approches la plus couramment utilisée pour estimer une donnée fonctionnelle est la méthode des splines de lissage ; notons qu'on peut considérer le lissage des données comme une étape de prétraitement des données. Nous allons présenter des méthodes de lissage de données fonctionnelles à la section 1.2. Les courbes de croissance et les courbes de température sont de bons exemples de données fonctionnelles. La figure 1.1(a) illustre l'évolution des tailles de 10 garçons entre les âges de 1 à 18 ans extrait du jeu de données de l'étude de la croissance de Berkeley tiré du paquet *fda* de R. Ce jeu de données contient l'évolution des tailles (en cm) de 39 garçons et de 54 filles mesurées à 31 âges, entre 1 et 18 ans. On remarque sur la figure que l'on est en présence d'une grille temporelle non équidistante de $m = 31$ points. La figure 1.1(b) illustre les moyennes des températures quotidiennes mesurées sur 30 ans dans 35 stations météorologiques au Canada. Les données proviennent du jeu de données du cycle météorologique annuel au Canada tiré du paquet *fda* de R. Les 35 courbes sont observées sur une grille équidistante composée de $m = 365$ points.

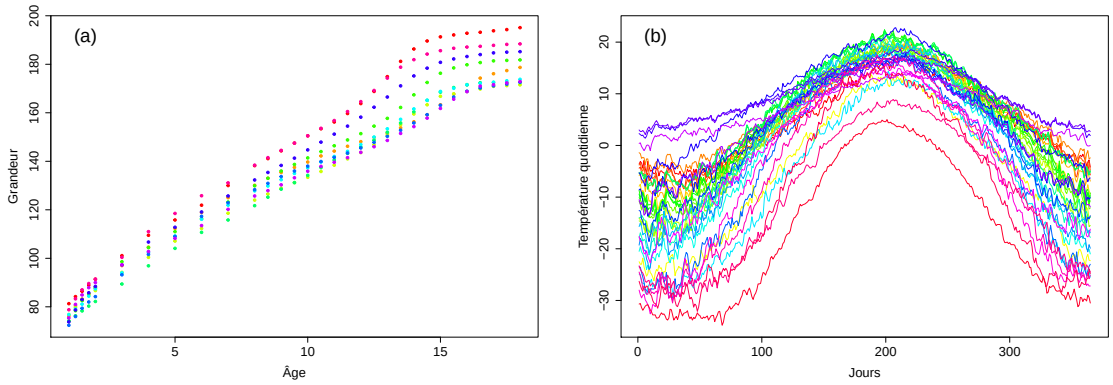

Figure 1.1: (a) Illustration de l'évolution de la taille de 10 garçons entre les âges de 1 à 18 ans du jeu de données de l'étude de croissance de Berkeley et (b) illustration des moyennes quotidiennes des températures mesurées sur 30 ans dans 35 stations météorologiques au Canada.

1.1.1 Caractéristiques d'une variable aléatoire fonctionnelle

Dans cette section, nous allons introduire quelques concepts de base, tels que l'espérance et la covariance pour une variable aléatoire fonctionnelle $\mathcal{X}$. Nous allons de plus présenter des statistiques descriptives pour un ensemble de données fonctionnelles $x_1, ..., x_n$. Les statistiques descriptives des données fonctionnelles sont similaires à celles des données univariées ; la différence est que dans le cas fonctionnel, chaque valeur descriptive est une fonction. Par exemple, l'équivalent de la moyenne d'un échantillon de variables univariées est, dans le cas fonctionnel, une fonction que nous appellerons fonction moyenne.

L'espérance de $\mathcal{X}$ est la fonction non aléatoire, élément de $L^2(\mathrm{T})$, définie pour chaque $t \in T$ par :

$$\mu(t) = E\big[(\mathcal{X}(t)\big].$$

La moyenne d'un échantillon $x_1, ...x_n$ est quant à elle définie par :

$$\bar{x}(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t),\ \forall t \in T. \tag{1.1}$$

On définit la variance de $\mathcal{X}$ comme suit :

$$Var(\mathcal{X}(t)) = E(\mathcal{X}(t) - \mu(t))^2,\ \forall t \in T.$$

La fonction d'écart-type est la racine carrée de la fonction de variance et la fonction de variance empirique $Var_x(t)$ est définie par :

$$Var_x(t) = \frac{1}{n-1}\sum_{i=1}^{n}[x_i(t) - \bar{x}(t)]^2,\ \forall t \in T.$$

Pour tout $s, t \in T$, la structure de dépendance sous-jacente de $\mathcal{X}$ peut être caractérisée par la fonction de covariance :

$$v(s,t) = E\big[\,\{\mathcal{X}(t) - \mu(t)\}\,\{\mathcal{X}(s) - \mu(s)\}\,\big].$$

La fonction de covariance empirique est calculée pour tout $s, t \in T$ par :

$$\hat{v}(s,t) = \frac{1}{n-1}\sum_{i=1}^{n}\{x_i(s) - \bar{x}(s))\}\,\{x_i(t) - \bar{x}(t)\}\,. \tag{1.2}$$

Dans le cas multivarié, la covariance est représentée sous une forme de matrice tandis que dans le cas fonctionnel, la covariance est un opérateur intégrale. Rappelons ce qu'est un opérateur intégrale. Une transformée intégrale (appelée aussi opérateur intégrale) est un opérateur linéaire défini sur certains espaces fonctionnels à l'aide d'une intégrale. La forme générale d'une transformée intégrale V de fonction noyau K est :

$$V\Big(f(s)\Big) = \int K(t,s)f(t)dt,$$

où f est une fonction et désigne l'entrée de cette transformée alors que la fonction $K(s,t)$ désigne le noyau de l'opérateur. Lorsque le noyau de l'opérateur est la

fonction de covariance $v(s,t)$, la transformée intégrale V est appelée l'opérateur de covariance :

$$V\Big(f(s)\Big) = \int_T v(s,t)f(t)dt,\ s \in T,\ f \in L^2(T). \tag{1.3}$$

Dans le cas fonctionnel, la fonction de corrélation empirique est donnée par :

$$\hat{\rho}(s,t) = \frac{\hat{v}(s,t)}{Var_x(s)Var_x(t)}.$$

La fonction moyenne et la fonction de covariance empirique des données des températures quotidiennes canadiennes sont illustrées à la figure 1.2. Notons que les données ont été préalablement lissées afin d'obtenir des courbes sur $T = [0, 365]$. La technique de lissage utilisée est présentée à la section 1.2.3.

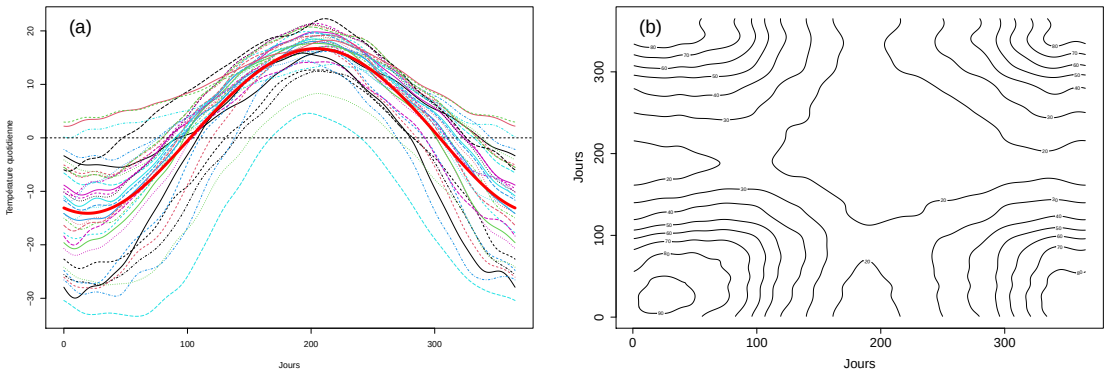

Figure 1.2: Illustration de la fonction moyenne en rouge (a) et de la fonction de covariance à l'aide de courbes de niveau (b) pour les données des températures canadiennes.

1.2 Transformation des données discrètes en données fonctionelles

Tel que nous l'avons mentionné à la section 1.1, en pratique on observe chaque courbe d'un échantillon $x_1, ..., x_n$ de façon discrète, i.e. que chaque courbe x_i,

$1 \leq i \leq n$, est observée aux temps $t_{i1}, ..., t_{im_i} \in T$ avec possiblement une erreur de mesure. Dénotons le vecteur contenant les m_i observations de la courbe x_i par $\mathbf{y_i} = (y_{i1}, ..., y_{im_i})$. La première étape d'une analyse de données fonctionnelles consiste donc à retrouver pour chaque courbe x_i, $1 \leq i \leq n$, sa forme fonctionnelle $\{x_i(t),\ t \in T\}$ à partir de sa forme discrétisée $\{(t_{ij}, y_{ij}) : 1 \leq j \leq m_i\}$. Puisque ceci est fait de façon indépendante pour chaque courbe, nous allons alléger la notation dans cette section en omettant l'indice i et en considérant ainsi qu'une seule courbe x. Dans le cas où les erreurs de mesure sont négligeables, nous avons :

$$y_j = x(t_j),\ 1 \leq j \leq m.$$

Dans ce cas, on peut utiliser une technique d'interpolation entre les paires de coordonnées (t_j, y_j) afin de trouver x. Par contre, en pratique, il est courant de considérer que nos observations ont des erreurs de mesure. Dans une telle situation, les valeurs observées sont :

$$y_j = x(t_j) + \epsilon_j,\ 1 \leq j \leq m,$$

où on suppose que les erreurs de mesure $\epsilon_1, ..., \epsilon_m$ sont des réalisations d'une variable aléatoire centrée $\mathcal{E}$ ($E[\mathcal{E}_j] = 0$) et de variance inconnue σ^2. Nous allons maintenant présenter des méthodes de lissage permettant de filtrer les erreurs de mesure dans le but de retrouver les courbes lisses $x_1, ..., x_n$. Autrement dit, nous allons trouver un estimateur de x à partir de données discrètes. Après avoir appliqué la méthode de lissage, nous pourrons procéder à l'étape de l'analyse de données fonctionnelles.

1.2.1 Combinaison linéaire de fonctions de base

Nous cherchons à modéliser la fonction x et à l'estimer à l'aide des observations $(t_1, y_1), ..., (t_m, y_m)$. Nous ne nous attendons pas à ce que x soit une fonction

linéaire de t, donc la régression linéaire simple n'est pas un bon choix et nous devons modéliser x comme une fonction non linéaire de t. Un modèle flexible pour x est de la représenter sous la forme d'une combinaison linéaire de fonctions de base. Notons que chaque fonction $x \in L^2(T)$ peut être représentée comme une combinaison linéaire de fonctions de base de $L^2(T)$, tout comme chaque vecteur d'un espace vectoriel peut être représenté comme une combinaison linéaire de vecteurs de base pour cet espace. Soit $\{\phi_k\}_{k=1}^{\infty}$ un ensemble de fonctions de base de $L^2(T)$, alors on peut écrire

$$x(t) = \sum_{k=1}^{\infty} c_k \phi_k(t),\ t \in T.$$

Il est donc possible d'approximer $x(t)$:

$$x(t) \approx \sum_{k=1}^{K} c_k \phi_k(t).$$

Le nombre de fonctions de base K peut être prédéterminé ou bien choisi par validation croisée, une méthode que nous verrons en détail à la section 1.2.2. Il existe plusieurs fonctions de base, nous allons nous concentrer sur les fonctions de base les plus connues. Parfois, le temps t est considéré de manière cyclique, par exemple lorsque t est le moment de l'année, nous serons alors en présence de données périodiques. Dans le cas où nous devons analyser ce type de données, les bases de Fourier sont les plus utilisées, tandis qu'en présence de données non-périodiques, les splines sont souvent utilisées. Commençons par voir les bases de Fourier. La base la plus connue est fournie par la série de Fourier :

$$x(t) = c_0 + c_1 \sin(\omega t) + c_2 \cos(\omega t) + c_3 \sin(2\omega t) + c_4 \cos(2\omega t) + ...$$

Les fonctions de base sont définies par les fonctions de sinus et de cosinus d'une fréquence croissante :

$$\phi_1(t) = 1,$$

et pour $k > 1$

$$\phi_k(t) = \begin{cases} \sin(\frac{k}{2}\omega t) & \text{pour } k \text{ pair,} \\ \cos(\frac{k-1}{2}\omega t) & \text{pour } k \text{ impair.} \end{cases}$$

La constante ω définit la période d'oscillation de la première paire de sinus et cosinus. Elle est égale à $2\pi/P$, où P est la période. On utilise en général $K = 2m+1$ fonctions de base où m est le plus grand nombre d'oscillations nécessaires dans une période de longueur P.

Si nous voulons modéliser des données non périodiques, les splines sont le système d'approximation le plus utilisé pour ce type de données. Une spline d'ordre $D+1$ est une fonction définie par morceaux par des polynômes de degré D, et dont les $(D-1)^e$ premières dérivées sont continues. Les points où les morceaux se rencontrent sont appelés noeuds. La propriété clé d'une spline est qu'elle est flexible tout en étant lisse. Rappelons que lorsqu'on fait une régression, nous écrivons la fonction de régression comme une combinaison linéaire des variables explicatives ; dans notre cas, la variable explicative est le temps. On rappelle que la fonction de régression polynomiale de degré D s'écrit :

$$x(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + ... + \beta_D t^D = \sum_{i=0}^{D} \beta_i g_i(t),$$

où $g_i(t) = t^i$. L'ensemble $\{g_i\}_{i=0}^{D}$ forme une base pour les polynômes de degré D. L'inconvénient de la régression polynomiale est qu'elle impose une structure globale sur la relation entre t et $x(t)$ tandis que les splines sont beaucoup plus flexibles puisqu'elles sont définies par morceaux. Une spline d'ordre 4 ($D = 3$) est appelée une spline cubique. Nous allons voir comment une spline cubique peut être construite à partir de polynômes par morceaux et nous allons voir qu'une spline peut être représentée par des fonctions de base. Une spline est obtenue en divisant le domaine T en plusieurs intervalles contigus et la spline peut être représentée par un polynôme dans chaque intervalle. Une spline d'ordre $D+1$,

associée à une séquence $\xi_1 < ... < \xi_S \in \mathbb{R}$ de S noeuds fixes, peut s'écrire comme une combinaison linéaire de fonctions de base :

$$x(t) = \sum_{l=1}^{S+D+1} \beta_l h_l(t),$$

où $\{h_l\}_{l=1}^{S+D+1}$ est l'ensemble des fonctions de base pour les splines que l'on va maintenant définir. La figure 1.3(a) illustre 3 polynômes cubiques par morceaux (délimités par 2 noeuds $\xi_1 = 10/3$ et $\xi_2 = 20/3$). Nous avons généré 200 observations (t_j, y_j) où les valeurs $t_1 < \ldots < t_{200}$ sont uniformément réparties entre 0 et 10 et $y_j = cos(1.4 * t_j) + \epsilon_j$ pour $\epsilon_1, \ldots, \epsilon_{200}$ des réalisations indépendantes d'une loi $N(0, 1/4)$. Les fonctions de base utilisées pour la construction des polynômes illustrés à la figure 1.3(a) sont :

$$\begin{array}{lll}
h_1(t) = \mathbb{1}(t < \xi_1), & h_2(t) = \mathbb{1}(\xi_1 \leq t < \xi_2), & h_3(t) = \mathbb{1}(t \geq \xi_2), \\
h_4(t) = t \cdot \mathbb{1}(t < \xi_1), & h_5(t) = t \cdot \mathbb{1}(\xi_1 \leq t < \xi_2), & h_6(t) = t \cdot \mathbb{1}(t \geq \xi_2), \\
h_7(t) = t^2 \cdot \mathbb{1}(t < \xi_1), & h_8(t) = t^2 \cdot \mathbb{1}(\xi_1 \leq t < \xi_2), & h_9(t) = t^2 \cdot \mathbb{1}(t \geq \xi_2), \\
h_{10}(t) = t^3 \cdot \mathbb{1}(t < \xi_1), & h_{11}(t) = t^3 \cdot \mathbb{1}(\xi_1 \leq t < \xi_2), & h_{12}(t) = t^3 \cdot \mathbb{1}(t \geq \xi_2).
\end{array}$$

Le nombre de paramètres utilisés dans ce modèle est 12, car il y a 4 paramètres pour chacune des 3 régions. Les polynômes par morceaux, même ceux continus aux noeuds, ont tendance à ne pas être lisses car il y a souvent des changements brusques de pente aux noeuds tels qu'on peut le voir à la figure 1.3(b) où on a illustré l'ajout de la contrainte qui exige que les polynômes soient continus aux noeuds ξ_1 et ξ_2 (mais les dérivées aux noeuds ne le sont pas). Pour éviter cela, il suffit d'imposer qu'en plus d'être continue, la fonction x ait un certain nombre de dérivées continues. Une spline cubique définie avec deux noeuds, $\xi_1 < \xi_2$, peut être représentée avec les fonctions de base suivantes :

$$h_1(t) = 1, h_2(t) = t, h_3(t) = t^2,$$

$$h_4(t) = t^3, h_5(t) = (t - \xi_1)_+^3, h_6(t) = (t - \xi_2)_+^3,$$

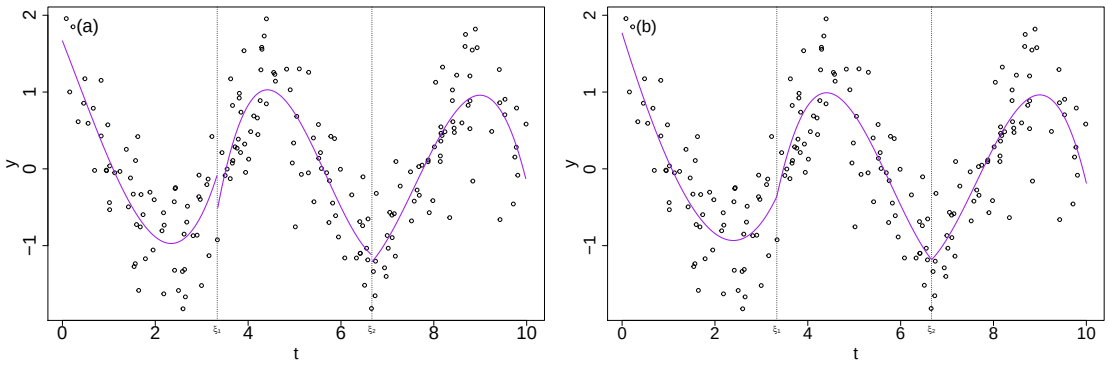

Figure 1.3: Illustration de 200 observations qui ont été simulées avec la fonction $cos(1.4 * t_j)$ avec des erreurs de mesure qui sont des réalisations indépendantes d'une loi $N(0, 1/4)$. (a) 3 polynômes cubiques par morceaux et (b) 3 polynômes cubiques continues par morceaux.

où t_+ désigne la partie positive, i.e. $t_+ = \max(t, 0)$, tandis qu'une spline de degré D, avec S noeuds $\xi_1 < ... < \xi_S$ peut être représentée par avec les fonctions de base suivantes :

$$h_s(t) = t^{s-1}, \ s = 1, ..., D + 1,$$

$$h_{s+D}(t) = (t - \xi_s)_+^D, \ s = 1, ..., S.$$

On a donc besoin d'un modèle à $S + 4$ paramètres pour représenter une spline cubique. En effet, on a besoin de quatre paramètres par région, pour un total de $4 \times (S + 1)$ paramètres. De plus, l'ajout d'une contrainte de continuité à chaque noeud fait en sorte qu'il y aura $3 \times S$ paramètres en moins, on obtient donc un total de $4 \times (S + 1) - 3 \times S = S + 4$ paramètres. Le nombre de paramètres utilisés dans l'exemple illustré à la figure 1.3(b) est : (3 régions $\times$ 4 paramètres par région) - (2 noeuds $\times$ 3 contraintes par noeud), donc 6 paramètres au total. Plus généralement, une spline d'ordre $D + 1$ aura $S + D + 1$ paramètres. La figure 1.4(a) montre la spline cubique lisse obtenue en imposant que les deux premières dérivées soient continues et la figure 1.4(b) illustre une spline d'ordre

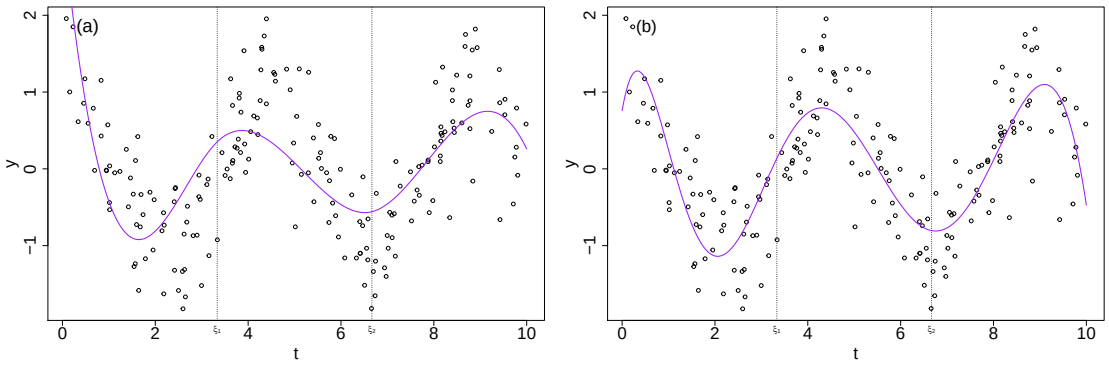

Figure 1.4: Illustration de 200 observations qui ont été simulées avec la fonction $cos(1.4 * t_j)$ avec des bruits qui sont des réalisations indépendantes d'une loi $N(0, 1/4)$. (a) Une spline cubique et (b) une spline d'ordre 6.

6. Une des difficultés avec les splines est de choisir le nombre de noeuds ainsi que leur emplacement. On peut essayer plusieurs choix de nombre de noeuds ou bien utiliser une approche par validation croisée pour déterminer le meilleur nombre de noeuds. Les noeuds sont souvent placés où il y a une forte courbure (ou variabilité), et on réduit le nombre de noeuds où il y a moins de fluctuations. Plus nous augmentons le nombre de noeuds et mieux la courbe s'adaptera aux données. Si nous posons le nombre de noeuds égal au nombre total d'observations ($S = m$), la spline s'adaptera exactement aux données. Comme les données sont sujettes à des erreurs de mesure, cela nous mènera à un modèle surajusté.

1.2.2 Estimation à l'aide de la méthode des moindres carrés

Nous venons de voir comment représenter une fonction par une combinaison linéaire de fonctions de base et nous allons maintenant présenter une méthode afin

d'estimer les coefficients. Rappelons que notre modèle est :

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t).$$

Comme nous l'avons vu dans la section 1.2.1, nous pouvons choisir les fonctions de base en fonction du type de nos données (périodiques ou non), nous allons donc supposer que les fonctions de base $\phi_k, k = 1, ..., K$, sont connues. Une méthode possible afin d'estimer les coefficients $c_1, ..., c_k$ est la méthode des moindres carrés. Cette méthode consiste à trouver les coefficients $c_1, ..., c_k$ qui minimise la somme des carrés des résidus (SCR) :

$$SCR(c_1, ..., c_k) = \sum_{j=1}^{m} (y_j - x(t_j))^2 = \sum_{j=1}^{m} \left(y_j - \sum_{k=1}^{K} c_k \phi_k(t_j) \right)^2 .$$

Définissons $\mathbf{y}$ comme étant le vecteur $(y_1, ..., y_m)^T \in \mathbb{R}^m$, $\mathbf{c}$ comme étant le vecteur des coefficients $(c_1, ..., c_k)^T$ et $\mathbf{\Phi} \in \mathbb{R}^{m \times K}$ la matrice de fonctions de base évaluées sur la grille $t_1, ..., t_m$, c'est-à-dire :

$$\mathbf{\Phi} = \begin{bmatrix} \phi_1(t_1) & \phi_2(t_1) & \cdots & \phi_K(t_1) \\ \phi_1(t_2) & \phi_2(t_1) & \cdots & \phi_K(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(t_m) & \phi_2(t_m) & \cdots & \phi_K(t_m) \end{bmatrix}.$$

Afin de simplifier les expressions, la fonction $SCR(c_1, ..., c_k)$ peut être exprimée sous forme matricielle :

$$SCR(\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})^T (\mathbf{y} - \mathbf{\Phi c}). \tag{1.4}$$

La dérivée de la fonction $SCR(\mathbf{c})$ par rapport à $\mathbf{c}$ est :

$$2\mathbf{\Phi}\mathbf{\Phi}^T\mathbf{c} - 2\mathbf{\Phi}^T\mathbf{y},$$

et en posant l'expression précédente égale à 0, nous obtenons l'estimateur $\hat{\mathbf{c}}$ qui minimise la somme des carrés des résidus :

$$\hat{\mathbf{c}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}.$$

On obtient donc l'estimation de la fonction $x(t)$:

$$\hat{x}_K(t) = \sum_{k=1}^{K} \hat{c}_k \phi_k(t).$$

Notons qu'un grand nombre K de fonctions de base ajoute de la flexibilité à l'estimation $\hat{x}_K(t)$, mais peut nous mener à un surajustement. Au contraire, l'utilisation de peu de fonctions de base pourrait faire en sorte que nous ne pouvons pas capturer les caractéristiques intéressantes des courbes que l'on souhaite modéliser. Cela implique peu de flexibilité, mais des fonctions plus lisses. On souhaite exprimer ce compromis en termes du biais de l'estimateur de $x(t)$:

$$Biais[\hat{x}_K(t)] = x(t) - E[\hat{x}_K(t)],$$

et de sa variance

$$Var[\hat{x}_K(t)] = E\left[\{\hat{x}_K(t) - E[\hat{x}_K(t)]\}^2\right].$$

L'utilisation de trop de fonctions de base risque d'engendrer un petit biais mais une grande variance, tandis que celle de trop peu de fonctions de base risque d'engendrer une petite variance mais un biais important. Habituellement, afin d'obtenir un équilibre entre le biais et la variance, on tente de minimiser l'erreur quadratique moyenne (EQM) :

$$EQM[\hat{x}_K(t)] = E\left[\{\hat{x}_K(t) - x(t)\}^2\right] = Biais^2[\hat{x}_K(t)] + Var[\hat{x}_K(t)].$$

Généralement, l'EQM est inconnue, mais nous pouvons l'estimer par validation croisée. Pour choisir le nombre de fonctions de base dans le cas fonctionnel, une approche que nous allons utiliser est un cas particulier de la validation croisée appelée en anglais "leave-one-out" (LOO). Prenons une courbe $x(t)$ et on considère sa forme discrétisée $\{(t_j, y_j) : 1 \leq j \leq m\}$ comme notre ensemble d'apprentissage auquelle on retire une observation (t_j, y_j) et cette dernière est considérée comme notre ensemble de validation. Autrement dit, à chaque itération, nous calculons

l'erreur associée à notre modèle sur une seule observation (t_j, y_j) et l'apprentissage se fera à partir des autres données. Dans notre cas, comme nous avons m observations, nous allons donc entraîner et tester notre modèle pour m fois.

Pour une itération j et une valeur K_l du paramètre K, on ajuste le modèle à partir de l'ensemble d'apprentissage (toutes les observations sauf la j$^{\text{ème}}$) et on calcule la distance $(y_j - \hat{x}_{K_l}^{-j}(t_j))^2$. On répète cette procédure pour un ensemble de valeurs $K \in K_1, ..., K_p$ du paramètre K et jusqu'à ce que toutes les observations (t_j, y_j) de la courbe $x(t_j)$ aient été utilisées comme un ensemble de validation. Nous allons obtenir une matrice $m \times p$ qui contient les m distances pour p différentes valeurs de K. Pour chacun des $K_1, ..., K_p$, on fait la somme des m distances obtenues comme suit :

$$LOO^{K_l}(\hat{x}) = \sum_{j=1}^{m} (y_j - \hat{x}_{K_l}^{-j}(t_j))^2.$$

Parmi $K_1, ..., K_p$, on choisit la valeur qui minimise le critère LOO^{K_l}. Il est à noter que cette approche lisse une courbe à la fois. Nous présentons une application de la validation croisée LOO pour le jeu de données des températures canadiennes à la figure 1.5. L'erreur totale au carré (LOO) en fonction du nombre de fonctions de base pour une courbe de température moyenne quotidienne pour la ville de Resolute est présentée en (a) alors que l'erreur totale au carré en fonction du nombre de fonctions de base pour la ville de Victoria est présentée en (b). On peut voir que le nombre de fonctions de base choisi pour la ville de Resolute est $K = 7$ et $K = 13$ pour la ville de Victoria.

1.2.3 Estimation à l'aide de splines de lissage

Nous allons voir une méthode non-paramétrique pour faire le lissage des données fonctionnelles. Les splines de lissage évitent d'avoir à choisir le nombre de fonctions de base K. L'objectif est de trouver une fonction x qui s'adapte bien et tout en

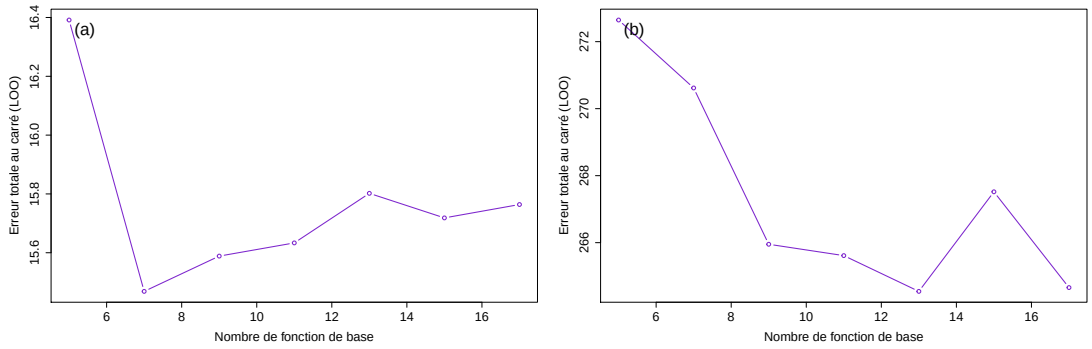

Figure 1.5: Illustration de l'erreur totale au carré en fonction du nombre de fonction de base pour la ville de Resolute (a) et la ville de Victoria (b) du jeu de données des températures canadiennes.

étant lisse. On veut donc trouver la fonction $x \in C^2$ qui minimise l'expression :

$$\sum_{j=1}^{m}(y_j - x(t_j))^2 + \lambda \cdot PEN_2(x), \tag{1.5}$$

où $PEN_2(x) = \int [x''(t)]^2 dt$ mesure la rugosité de x et $\lambda > 0$ est le paramètre de lissage qui contrôle le compromis entre la qualité de l'ajustement et le lissage. À mesure que λ augmente, la rugosité est de plus en plus pénalisée et x sera de plus en plus lisse. À l'opposé, à mesure que λ diminue, la rugosité est de moins en moins pénalisée et x s'ajustera de mieux en mieux aux données. Par un théorème de Boor (2001), on sait que la fonction qui minimise l'expression 1.5 est en fait une spline cubique avec un noeud à chaque t_j, $j = 1, ..., m$. Bien qu'il y ait un noeud à chaque observation de notre ensemble de données, la courbe résultante sera lisse grâce au régularisateur $PEN_2(x)$. En posant encore une fois :

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t) = \mathbf{c}^T \Phi(t),$$

où $K = m + 4$ et en utilisant les splines cubiques comme fonctions de base, on estime les coefficients c_k en minimisant une version modifiée du critère d'ajuste-

ment des moindres carrés présenté à la section 1.2.2 (équation 1.4). Ce nouveau critère est nommé la somme des carrés des résidus pénalisés ($SCRPEN$) et est défini sous forme matricielle comme suit :

$$SCRPEN(\mathbf{y} \mid \mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})^T(\mathbf{y} - \mathbf{\Phi c}) + \lambda \times PEN_2(x).$$

Alors

$$PEN_2(x) = \int [D^2x(t)]^2 dt = \int \mathbf{c}^T[D^2\Phi(t)][D^2\Phi(t)^T]\mathbf{c}\ dt = \mathbf{c}^T\mathbf{R}_2\mathbf{c},$$

où $\mathbf{R}_2 = \int [D^2\Phi(t)][D^2\Phi(t)^T]dt = \int [D^2\mathbf{\Phi}][D^2\mathbf{\Phi}^T]$ est la matrice de pénalité. La dérivée de la fonction $SCRPEN(\mathbf{y} \mid \mathbf{c})$ par rapport à $\mathbf{c}$ est :

$$-\mathbf{\Phi}^T(2\mathbf{y} - \mathbf{\Phi c}) + \lambda\mathbf{R}_2\mathbf{c}$$

et en posant l'expression précédente égale à 0, on obtient :

$$-2\mathbf{\Phi}^T\mathbf{y} + (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{R}_2)\mathbf{c} = 0.$$

L'estimateur de $\mathbf{c}$ est donc

$$\hat{\mathbf{c}}_{PEN} = (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{R}_2)^{-1}\mathbf{\Phi}^T\mathbf{y},$$

et on obtient l'estimation de la fonction $x(t)$:

$$\hat{x}(t) = \sum_{k=1}^{K} \hat{\mathbf{c}}_{PEN}\phi_k(t),$$

qui s'écrit sous forme matricielle comme : $\hat{\mathbf{x}} = \mathbf{S}_\lambda\mathbf{y}$, où $\mathbf{S}_\lambda = (\mathbf{\Phi}^T\mathbf{\Phi} + \lambda\mathbf{R}_2)^{-1}\mathbf{\Phi}^T$ est appelée la matrice de lissage. Les degrés de liberté (dl) de l'ajustement correspondent au nombre de paramètres du modèle qui doivent être estimés, et il est possible de montrer que ce nombre égale à la trace de la matrice $\mathbf{S}_\lambda$. Il existe de nombreuses façons de choisir le paramètre de lissage λ ; par exemple, la validation croisée "leave-one-out" (LOO), la validation croisée généralisée (VCG) et l'estimation par vraisemblance maximale restreinte. La fonction LOO de la validation

croisée LOO qui a été présentée à la section 1.2.2 peut en fait être écrite de la façcon suivante :

$$LOO^{\lambda}(\hat{x}_{\lambda}) = \frac{1}{m}\left(\sum_{j=1}^{m}(y_j - \hat{x}_{\lambda}^{-j}(t_j)\right)^2 = \frac{1}{m}\sum_{j=1}^{m}\frac{(y_j - \hat{x}_{\lambda}(t_j))^2}{(1 - s_{\lambda,jj})^2},$$

où les $s_{\lambda,jj}, j = 1, ..., m$ sont les éléments diagonaux de la matrice de lissage s_{λ}. On remarque qu'il est possible de calculer VCO^{λ} en une seule itération (plutôt qu'en m itérations), car la fonction $\hat{x}_{\lambda}$ est obtenue en utilisant toutes les données. Finalement, on choisit la valeur λ qui minimise $VCO^{\lambda}(\hat{x}_{\lambda})$. Une autre mesure qui est populaire dans la littérature sur les splines de lissage est la mesure de validation croisée généralisée. Elle a tendance à donner des résultats plus lisses que la VCO. La mesure VGC est définie comme suit :

$$VCG^{\lambda}(\hat{x}) = \left(\frac{m}{m - dl(\lambda)}\right)\left(\frac{\sum_{j=1}^{m}(y_j - \hat{x}_{\lambda}(t_j))^2}{m - dl(\lambda)}\right),$$

où $dl(\lambda) = trace(\mathbf{S}_{\lambda})$. Nous allons maintenant présenter une application de la validation croisée généralisée sur le jeu de données de l'étude de croissance des garçons de Berkeley et sur le jeu de données des températures canadiennes vus à la section 1.1. La figure 1.6 présente les valeurs du critère de VCG pour différentes valeurs de λ. La figure 1.6(a) montre un minimum à $\log_{10}(\lambda)$ = -4.25 et la figure 1.6(b) montre un minimum à $\log_{10}(\lambda) = 6$. Les figures 1.7(a) et (b) illustre le lissage des jeux de données des courbes de croissances des garçons et des courbes des températures canadiennes en utilisant les paramètres de lissage $\lambda = 10^{-4.25}$ et $\lambda = 10^{6}$ respectivement.

1.3 Analyse en composantes principales fonctionnelles

Une fois que le lissage des données fonctionnelles a été effectué, il est courant de procéder à une étape de réduction de la dimension des données à l'aide d'une analyse en composantes principales (ACP). Dans cette section, nous allons tout

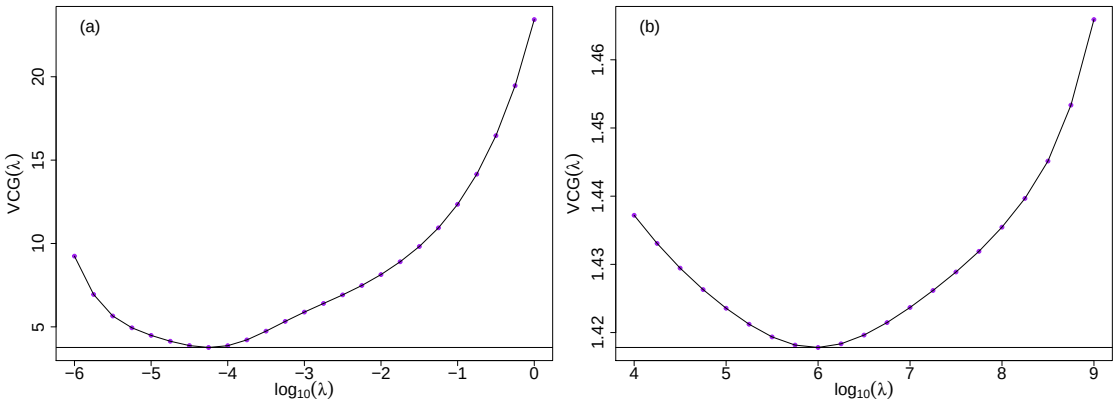

Figure 1.6: Illustration des valeurs du critère de la validation croisée généralisée pour différentes valeurs de lissage λ. (a) Les valeurs du critères VCG en fonction de λ pour les courbes de croissance des garçons de l'étude de Berkeley et (b) les valeurs du critères VCG en fonction de λ pour le jeux de données des températures canadiennes.

d'abord faire un rappel de l'ACP dans le cas multivarié et nous allons ensuite la présenter dans le cas fonctionnel.

1.3.1 Analyse en composantes principales dans le cas multivarié

De nos jours, il est courant d'avoir des données où il y a beaucoup plus de variables explicatives que d'observations ; c'est souvent le cas, par exemple en génétique. Ces données sont souvent difficiles à analyser en raison de leur grande dimension. L'analyse en composantes principales est une méthode qui permet de réduire la dimension d'un ensemble de données tout en minimisant la perte d'information. À l'aide d'une transformation orthogonale, l'ACP a pour but de créer à partir des variables originales, de nouvelles variables non corrélées qui maximisent la variance des données. Ces nouvelles variables, définies par le jeu de données en question, sont appelées composantes principales. Rappelons que l'orthogonalité,

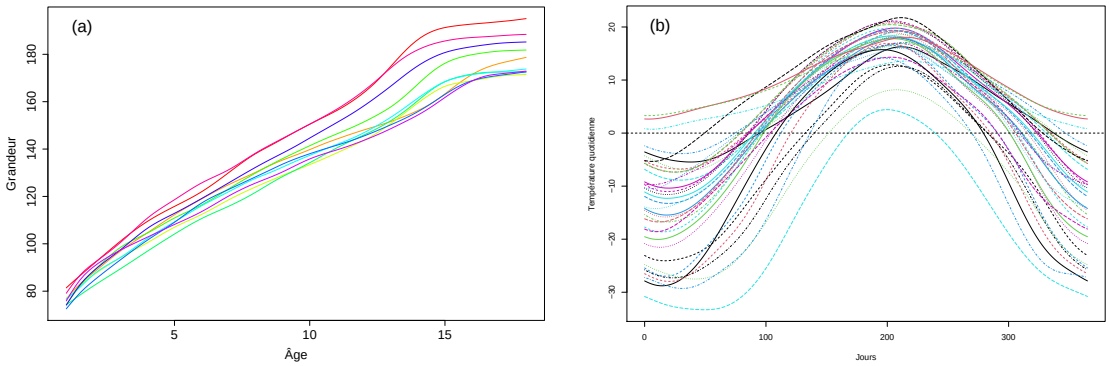

Figure 1.7: (a) Illustration des données lisses du jeu de données des 10 premiers garçons de l'étude de Berkeley avec $\lambda = 1e^{-4.25}$ et (b) du jeu de données des températures canadiennes avec $\lambda = 1e^{6}$.

dans notre contexte, préserve les longueurs des vecteurs et les angles entre les vecteurs. Considérons un ensemble de données avec n observations sur d variables. Ces données définissent d vecteurs $\mathbf{x}_1, ..., \mathbf{x}_d$ de dimension n. On obtient donc une matrice de données $\mathbf{X}_{n\times d}$ qui forme un nuage de n points dans un espace à d dimensions dont la $j^{\text{ième}}$ colonne est le vecteur $\mathbf{x}_j$ des observations sur la $j^{\text{ième}}$ variable. L'analyse en composantes principales a pour but de trouver de nouvelles directions (composantes principales) qui sont basées sur la matrice de variance-covariance des variables originales, de telle sorte qu'une variable avec un écart-type élevé aura un poids plus élevé pour le calcul de la direction qu'une variable avec un écart-type faible. Puisque l'écart-type d'une variable n'est pas invariant à l'échelle dans laquelle est exprimée la variable, les variables doivent être préalablement standardisées. L'ACP permet de trouver une combinaison linéaire des colonnes de la matrice $\mathbf{X}$ ayant la plus grande variance possible. Ces combinaisons linéaires sont données par $\sum_{j=1}^{d} a_j\mathbf{x}_j = \mathbf{Xa}$, où $\mathbf{a} = (a_1, a_2, ..., a_d)^T$ est un vecteur de constantes. La matrice de variance-covariance de la combinaison linéaire $\mathbf{Xa}$ est donnée par $\text{Var}(\mathbf{X}a) = \mathbf{a}^T\mathbf{Sa}$, où $\mathbf{S} = \mathbf{X}^T\mathbf{X}$. Obtenir la combinaison linéaire

avec la variance maximale est équivalent à trouver un vecteur $\mathbf{a}$ de dimension d qui maximise la forme quadratique $\mathbf{a}^T\mathbf{S}\mathbf{a}$ sous la contrainte que $\mathbf{a}^T\mathbf{a} = 1$. Notons que si nous considérons des vecteurs qui ne sont pas de norme unitaire dans le problème de maximisation, on n'obtiendrait pas une solution appropriée, car la variance de la projection pourrait devenir arbitrairement grande en augmentant la norme du vecteur. Nous voulons donc maximiser $\mathbf{a}^T\mathbf{S}\mathbf{a} - \lambda(\mathbf{a}^T\mathbf{a} - 1)$, où λ est un multiplicateur de Lagrange. Pour cela, il suffit de différencier par rapport au vecteur $\mathbf{a}$ et de le mettre égal au vecteur nul, ce qui nous donne :

$$(\mathbf{S} - \lambda\mathbf{I})\mathbf{a} = \mathbf{0} \iff \mathbf{S}\mathbf{a} = \lambda\mathbf{a}. \tag{1.6}$$

Nous pouvons donc résoudre cette équation en utilisant la décomposition spectrale de la matrice de variance-covariance $\mathbf{S}$. Rappelons tout d'abord ce qu'est la décomposition spectrale d'une matrice. Soit $\mathbf{C}$ une matrice carrée symétrique $d \times d$ possédant des valeurs propres $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$ dont les vecteurs propres orthonormés correspondants sont $\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_d} \in \mathbb{R}^d$. Alors $\mathbf{C}$ peut s'écrire sous la forme :

$$\mathbf{C} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T = \begin{bmatrix} | & | & & | \\ \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix} \begin{bmatrix} — & \mathbf{w}_1 & — \\ — & \mathbf{w}_2 & — \\ & \vdots & \\ — & \mathbf{w}_d & — \end{bmatrix},$$

où Λ est une matrice diagonale dont les éléments diagonaux sont les valeurs propres de $\mathbf{C}$. Puisque la matrice de variance-covariance $\mathbf{S}$ est symétrique, elle admet donc une décomposition spectrale définie par :

$$(\lambda_1, \mathbf{a}_1), (\lambda_2, \mathbf{a}_2), (\lambda_3, \mathbf{a}_3), ..., (\lambda_d, \mathbf{a}_d).$$

Chaque couple $(\lambda_j, \mathbf{a}_j), j = 1, ..., d$ est une solution de l'équation 1.6 mais $\mathbf{a}_1$ est le vecteur propre associé à la plus grande valeur propre ainsi $\mathbf{a}^T\mathbf{S}\mathbf{a}$ atteint sa valeur maximale $\mathbf{a}_1^T\mathbf{S}\mathbf{a}_1 = \lambda_1$ en $\mathbf{a}_1$. Une fois que la direction $\mathbf{a}_1$ a été choisie, il suffit

de trouver une direction $\mathbf{a}_2$ qui maximise $\mathbf{a}^T\mathbf{S}\mathbf{a}$ et qui est orthogonale à $\mathbf{a}_1$. Cette direction est donnée par le deuxième vecteur propre qui est tel que $\mathbf{a}_2^T\mathbf{S}\mathbf{a}_2 = \lambda_2$. On peut répéter cette procédure jusqu'à ce qu'on obtienne la dernière composante principale $\mathbf{a}_d$ qui est associé à la plus petite valeur propre λ_d.

Remarquons que l'ACP préserve les $m < d$ premières valeurs propres de la matrice de variance-covariance $\mathbf{S}$ lorsqu'on projette les observations d'un espace de dimension d sur un espace de dimension inférieure m. Considérons la matrice de variance-covariance $\mathbf{S} = \mathbf{X}^T\mathbf{X}$ et sa décomposition spectrale : $\mathbf{A}\mathbf{\Lambda}\mathbf{A}^T$. Soit $\mathbf{T} = \mathbf{X}\mathbf{U}$, les observations projetées sur m composantes principales, où

$$\mathbf{U} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_m \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{d\times m}.$$

Alors la matrice de variance-covariance des données projetées dans l'espace de dimension réduite m est :

$$\begin{aligned} Cov(\mathbf{T}) &= (\mathbf{X}\mathbf{U})^T(\mathbf{X}\mathbf{U}) \\ &= \mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U} \\ &= \mathbf{U}^T(\mathbf{X}^T\mathbf{X})\mathbf{U} \\ &= \mathbf{U}^T\mathbf{S}\mathbf{U} \\ &= \mathbf{U}^T\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\mathbf{U} = \mathbf{\Lambda}. \end{aligned}$$

La variance totale de $\mathbf{X}\mathbf{U}$ est mesurée par la trace de la matrice $\mathbf{\Lambda}$. La variance totale expliquée lorsqu'on considère les d composantes principales est donc donnée par :

$$\sum_{i=1}^{d} \lambda_i.$$

La proportion de variation totale expliquée par les m premières composantes prin-

cipales est donnée par :

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{d} \lambda_i}.$$

Généralement, nous retenons que les composantes les plus importantes, une façon de le faire est de choisir m tel que la proportion de variation totale expliquée soit d'au moins 0.90.

Considérons l'exemple illustré à la figure 1.8. Le graphique (a) illustre un jeu de données standardisés où $d = 2$ alors que les axes illustrés en mauve et en vert représentent respectivement les axes de la première et seconde composante pricipale du jeu de données. Si nous voulons projeter les observations sur les deux composantes principales, l'ACP ne fait qu'une rotation des axes des composantes principales tel qu'illustré en (b). Si nous voulons projeter nos données dans un espace de dimension $m = 1$ en gardant le plus d'information possible, il suffit de projeter les données sur la $1^{\text{ère}}$ composante principle tel qu'illustré en (c), car la première composante représente la direction pour laquelle la variance est maximale dans le jeu de donnée. Chaque observation peut maintenant être projetée sur cette axe afin d'obtenir les coordonnées le long de la direction de la première composante principale. Ces nouvelles coordonnées sont également connues sous le nom de score. On obtient un nouveau jeu de données qui est représenté en (d). Un jeu de données standardisées de dimension $d = 3$ est illustré en (e). Nous pouvons obtenir la représentation du jeu de données en $m = 2$ dimensions en traçant les scores associés aux deux premières composantes principales ; un tel graphique est appelé graphique des scores. La figure (f) représente le graphique des scores basé sur les deux premières composantes principales du jeu données en (e).

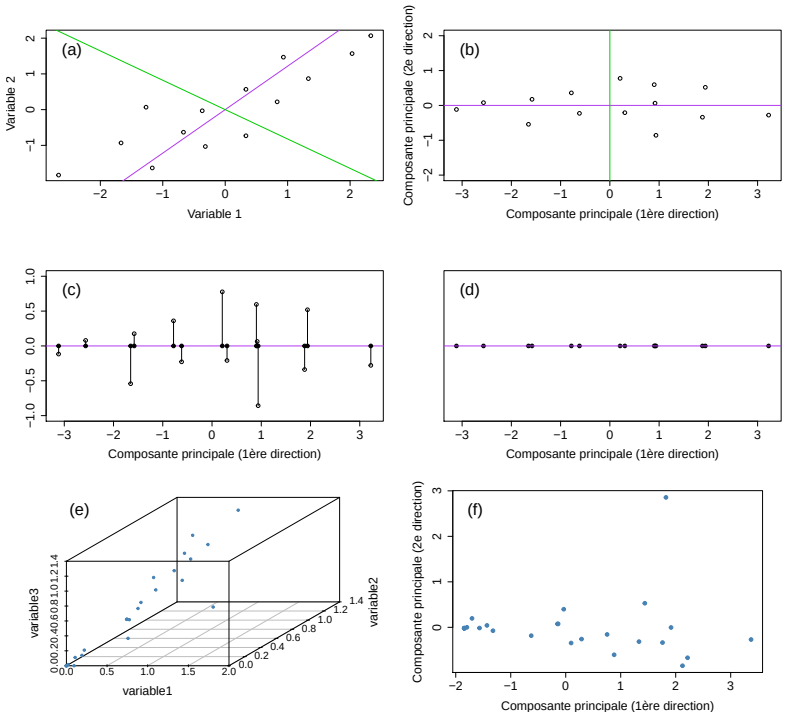

Figure 1.8: Illustration de la méthode d'ACP appliquée sur deux jeux de données. La figure (a) représente un nuage de points composé de 14 données standardisées représentées en 2 dimensions, le graphique (b) montre la rotation des axes des 2 composantes principales, la figure (c) illustre la projection des observations sur la $1^{\text{ère}}$ composantes principales, le graphique (d) illustre le sous-espace réduit par l'ACP en dimension m = 1, la figure (f) illustre le graphique des scores d'un jeu de données composées de 26 observations en dimension d = 3 tel qu'illustré en (e).

1.3.2 Analyse en composantes principales dans le cas fonctionnel

Une généralisation de l'ACP multivariée au cas fonctionnel est possible. Nous allons voir que l'analyse en composantes principales fonctionnelles (ACPF), peut être utilisée pour réduire la dimension des données fonctionnelles. Similairement

à l'ACP multivariée, l'objectif de l'ACPF est de projeter les données dans un sous-espace vectoriel de $L^2(T)$ de dimension finie tout en conservant le maximum d'information des données fonctionnelles. Soit $\{X_i(t) : t \in T\}_{i=1}^n$ un ensemble de de n réalisations indépendantes et identiquement distribuées à une variable aléatoire $X \in L^2(T)$ centrée, c'est-à-dire de moyenne nulle et soit $\{x_i(t) : t \in T\}_{i=1}^n$ une observation d'un tel ensemble. Comme pour l'ACP multivariée, trouver les composantes principales fonctionnelles revient à trouver la décomposition spectrale de l'opérateur de covariance, i.e. à résoudre l'équation :

$$V\Big(\xi(s)\Big) = \int v(s,t)\xi(t)dt = \lambda\xi(s), s \in T, \tag{1.7}$$

où λ est une valeur propre et $\xi(s)$ est une fonction propre de l'opérateur V.

	ACP multivariée	**ACP fonctionnelle**
Variable aléatoire	$\mathbf{X} \in \mathbb{R}^d$ $\mathbf{X} = (\mathbf{X}_1, ..., \mathbf{X}_d)^T$	$\mathcal{X} \in L^2(T)$ $\mathcal{X} = \{X(t), t \in T\}$
Dimension	$d < \infty$	∞
Moyenne	$\mu = \mathrm{E}(\mathbf{X})$	$\mu(t) = \mathrm{E}(X(t))$
Covariance	$\Sigma = Var(\mathbf{X})$	$v(s,t) = \mathrm{Cov}(X(s), X(t))$
Valeur(s) propre(s)	$\lambda_1, \lambda_2, ..., \lambda_d$	$\lambda_1, \lambda_2, ...$
Produit scalaire	$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{j=1}^d \mathbf{X}_j y_j$	$\langle \mathcal{X}, \mathcal{Y} \rangle = \int_T X(t)Y(t)dt$
Direction de l'ACP	Vecteur(s) propre(s) : $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_d$	Fonctions propres : $\xi_1(t), \xi_2(t), ...$
Scores	$d_j = \sum_{j=1}^d \mathbf{X}_j a_j$	$f_j = \int_T \xi_j(t)X(t)dt$

Tableau 1.1: Comparaison de divers éléments de l'ACP multivariée et l'ACP fonctionnelle.

La solution de cette équation consiste en une séquence de couples (λ_j, ξ_j) telle que λ ; est la valeur propre associée à la fonction propre ξ_j. Le tableau 1.1 résume la

distinction entre l'ACP multivariée et l'ACP fonctionnelle. Comme indiqué dans le tableau 1.1, dans le cas multivarié, il existe d valeurs propres et d vecteurs propres associés à un ensemble de données de dimension finie d. Théoriquement, il existe une infinité de paires de valeurs propres et de fonctions propres pour un espace de dimension infinie ; l'idée est de trouver les m premières composantes principales fonctionnelles $\xi_1(t), \xi_2(t), ..., \xi_m(t)$, car elles résument les principales sources de variance parmi les courbes. Le score d'une observation i pour la j$^{\text{ème}}$ valeur composante principale fonctionnelle est défini comme $f_{ij} = \int_T \xi_j(t) X_i(t) dt = \langle \xi_i, X_j \rangle$. Les scores $f_{i1,...,f_{im}}$ sont en fait les coordonées de la projection de la courbe $x_i(t)$ sur les fonctions propres $\xi_1, ..., \xi_m$. Notons que les scores sont maximisés sous la contrainte $\int \xi_j(t)^2 dt = \|\boldsymbol{\xi}_j\|^2 = 1$ et que les composantes principales fonctionnelles sont orthogonales, c'est-à-dire :

$$\int \xi_i(t)\xi_j(t)dt = 0, \ \forall i \neq j.$$

De plus, les scores sont des variables aléatoires centrées et non corrélées telles que

$$\frac{1}{n}\sum_{j=1}^{n} f_{ik} = 0, \ \sum_{j=1}^{n} f_{ik} f_{il} = 0, \ \sum_{j=1}^{n} f_{ik}^2 = \lambda_i, \ \forall \ k \neq l.$$

Nous allons voir deux méthodes afin d'obtenir les fonctions propres et les valeurs propres en pratique. Les deux méthodes permettent de résoudre la version empirique de l'équation 1.7, i.e :

$$\hat{V}\Big(\hat{\xi}(s)\Big) = \int \hat{v}(s,t)\hat{\xi}(t)dt = \hat{\lambda}\hat{\xi}(s), s \in T, \tag{1.8}$$

où $\hat{v}(s,t)$ est la fonction de covariance empirique définie en 1.2, $\hat{V}$ l'opérateur de covariance associé à $\hat{v}$ et $(\hat{\lambda}, \hat{\xi})$ une valeur et une fonction propre de $\hat{V}$. La première méthode consiste à discrétiser l'équation 1.8, i.e., à utiliser les données discrétisées selon une grille qui est suffisamment fine $\mathbf{t} = [t_1, ..., t_K]$. On a donc une matrice $\mathbf{X}$ de taille $n \times K$ définie par $x_i(t_1), ..., x_i(t_K)$, $i \in \{1, ..., n\}$ et une

matrice de covariance empirique $\mathbf{S} = n^{-1}\mathbf{X}^T\mathbf{X}$ de taille $K \times K$. Alors, il est possible d'utiliser la décomposition spectrale vue à la section 1.3.1 dans le cas de l'ACP multivariée pour obtenir les paires de valeurs propres et de vecteurs propres $(\lambda_j, \mathbf{a}_j), j = 1, ..., K$. Notons qu'avec cette méthode, le nombre maximal de vecteurs propres et de valeurs propres est de K.

La deuxième méthode repose sur la représentation des données par une combinaison linéaire de fonctions de base et elle est beaucoup plus utilisée. Supposons que chaque fonction $x_i(t), i = 1, ..., n$ est exprimée avec le même ensemble de fonctions de base $\{\phi_k(t)\}_{k=1}^K$, alors on peut écrire $x_i(t)$ comme la combinaison linéaire :

$$x_i(t) = \sum_{k=1}^{K} c_{ik}\phi_k(t).$$

Si on note $\mathbf{x}$ le vecteur dont les composantes sont les fonctions $x_1(t), ..., x_n(t)$ et $\boldsymbol{\phi}$ celui dont les composantes sont $\phi_1(t), ..., \phi_K(t)$, on peut exprimer $\mathbf{x}$ d'une manière plus compacte :

$$\mathbf{x} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1K} \\ c_{21} & c_{22} & \cdots & c_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nK} \end{bmatrix} \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \vdots \\ \phi_K(t) \end{bmatrix} = \mathbf{C}\boldsymbol{\phi},$$

où la matrice de coefficients $\mathbf{C}$ est de dimension $n \times K$. Supposons que

$$\hat{\xi} = \sum_{j=1}^{K} \phi_j(t)b_j = \mathbf{b}^T\boldsymbol{\phi}$$

est la fonction propre de l'opérateur de covariance $\hat{V}$ associée à la valeur propre $\hat{\lambda}$. Notons que le nombre maximal de fonctions propres que l'on peut obtenir avec cette méthode est K. En substituant ces combinaisons linéaires de fonctions de base dans l'équation 1.8, on obtient :

$$\hat{v}(s,t) = n^{-1}\boldsymbol{\phi}(s)^T\mathbf{C}^T\mathbf{C}\boldsymbol{\phi}(t). \tag{1.9}$$

Définissons la matrice $\mathbf{W}$ de dimension $K \times K$ comme suit :

$$\mathbf{W} = \int \boldsymbol{\phi}\boldsymbol{\phi}^T. \tag{1.10}$$

Nous cherchons à résoudre l'équation 1.8 sous la contrainte :

$$\|\hat{\boldsymbol{\xi}}\|^2 = \int [\hat{\xi}(t)]^2 dt = \mathbf{b}^T \int \boldsymbol{\phi}(t)^T \boldsymbol{\phi}(t) dt \mathbf{b} = \mathbf{b}^T \mathbf{W} \mathbf{b} = 1. \tag{1.11}$$

En substituant les équations 1.9, 1.10 et 1.11 dans l'équation 1.8, celle-ci devient :

$$\begin{aligned}
\int \hat{v}(s,t)\hat{\xi}(t)dt &= \int n^{-1}\boldsymbol{\phi}(s)^T \mathbf{C}^T \mathbf{C} \boldsymbol{\phi}(t)\boldsymbol{\phi}(t)^T \mathbf{b} dt \\
&= n^{-1}\boldsymbol{\phi}(s)^T \mathbf{C}^T \mathbf{C} \int \boldsymbol{\phi}(t)\boldsymbol{\phi}(t)^T \mathbf{b} dt \\
&= n^{-1}\boldsymbol{\phi}(s)^T \mathbf{C}^T \mathbf{C} \mathbf{W} \mathbf{b} \\
&= \hat{\lambda}\boldsymbol{\phi}(s)^T \mathbf{b}.
\end{aligned}$$

Cette équation doit être vraie pour toutes les valeurs $s \in T$, et par conséquent

$$n^{-1}\mathbf{C}^T \mathbf{C} \mathbf{W} \mathbf{b} = \hat{\lambda}\mathbf{b}.$$

Si nous définissons $\mathbf{u} = \mathbf{W}^{\frac{1}{2}}\mathbf{b}$, alors nous avons l'équation symétrique

$$n^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{C}^T \mathbf{C} \mathbf{W}^{\frac{1}{2}} \mathbf{W}^{\frac{1}{2}} \mathbf{b} = \hat{\lambda}\mathbf{W}^{\frac{1}{2}}\mathbf{b}$$

$$n^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{C}^T \mathbf{C} \mathbf{W}^{\frac{1}{2}} \mathbf{u} = \hat{\lambda}\mathbf{u},$$

sous la contrainte $\mathbf{u}^T\mathbf{u} = 1$. On voit que $\mathbf{u}$ est un vecteur propre de la matrice $n^{-1}\mathbf{W}^{\frac{1}{2}}\mathbf{C}^T \mathbf{C} \mathbf{W}^{\frac{1}{2}}$. Une fois que nous avons trouvé $\mathbf{u}$, nous pouvons poser $\mathbf{b} = \mathbf{W}^{-\frac{1}{2}}\mathbf{u}$ pour pouvoir ensuite obtenir les fonctions propres $\hat{\xi}_1 ..., \hat{\xi}_K$ de l'opérateur de covariance $\hat{V}$, ordonnées par les valeurs propres $\hat{\lambda}_1 > ... > \hat{\lambda}_K$.

1.3.3 Représentation graphique des températures canadiennes en utilisant l'analyse en composantes fonctionnelles

Comme dans le cas de l'ACP multivarié, on considère souvent que les premières fonctions propres en ACPF. Encore une fois, considérons l'exemple du jeu de

données des températures canadiennes. La figure 1.9(a) représente la variance cumulative exprimée en pourcentage en fonction du nombre de fonctions propres (composantes principales). Nous pouvons voir qu'avec seulement deux fonctions propres, la variance cumulative dépasse déjà le seuil de 90 % illustré par la ligne horizontale sur le graphique. Les quatre premières fonctions propres sont illustrées à la figure 1.9(b). La courbe en noire (FP1) représente la première fonction propre

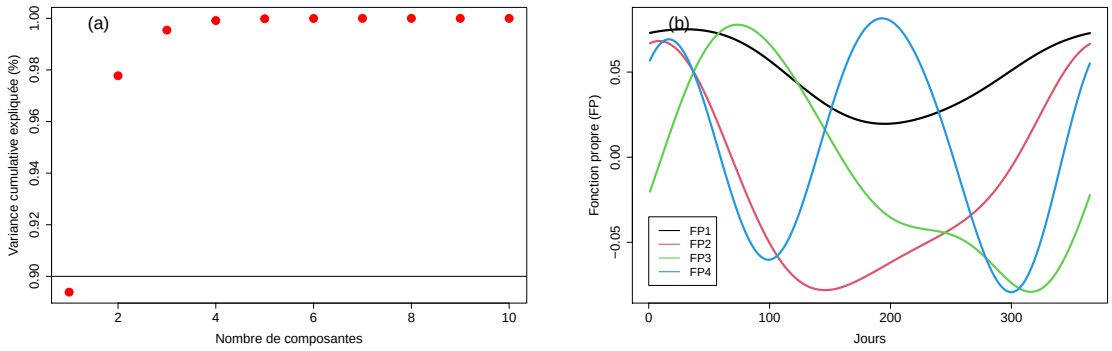

Figure 1.9: (a) La variance cumulative expliquée en fonction du nombre de composantes et (b) les 4 premières fonctions propres résumant 99.68 % de la variance totale du jeux de données des températures canadiennes quotidiennes.

$\xi_1(t)$ et elle capture 89 % de la variation totale du jeu de données. On peut constater qu'elle est positive durant toute l'année et les poids placés aux stations d'hiver sont beaucoup plus élevés que les poids placés aux stations d'été. Ceci indique que les températures à travers les villes canadiennes sont plus variables pendant la saison hivernale, c'est-à-dire qu'il y a une plus forte variation entre les températures durant les mois d'hiver que durant les mois d'été. La figure 1.10 illustre les scores $(\langle \xi_1, x_i \rangle, \langle \xi_2, x_i \rangle)$ pour chaque courbe $x_i, i = 1, ..., 35$ de nos données. Les villes dont le score $\langle \xi_1, x_i \rangle$ est grand ont des hivers plus chauds que la moyenne tandis que celles dont le score est petit auront des hivers beaucoup plus froids que la moyenne. Par exemple, on voit que la ville de Vancouver (située

sur la côte du Pacifique) a un score élevé pour la première composante principale, tandis que la ville de Resolute (située au Nunavut) a un score très négatif. La

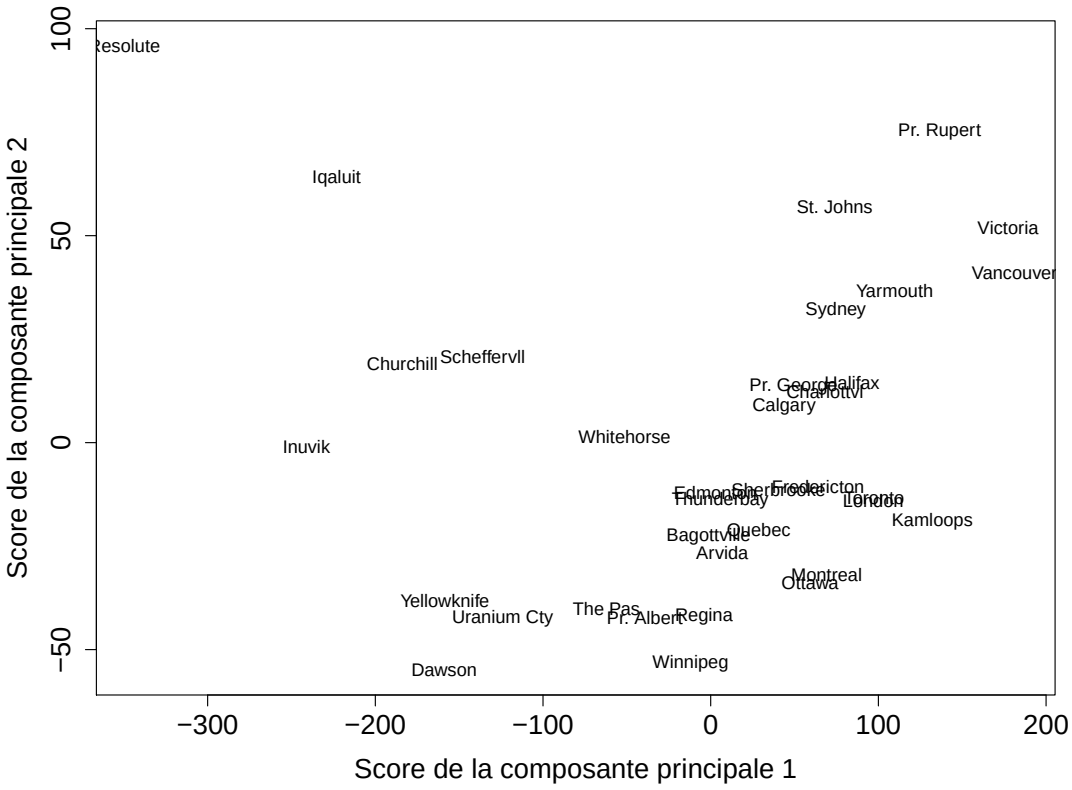

Figure 1.10: Représentation en deux dimensions des 35 villes canadiennes dans le plan défini par les 2 premières fonctions propres.

deuxième composante principale (FP2) représentée en rouge à la figure 1.9(b), capture 8 % de la variation totale et décrit la variation de la différence entre les températures hivernales et estivales. Les villes dont le score $\langle \xi_2, x_i \rangle$ est élevé ont une plus petite différence de température entre leur hiver et leur été que la moyenne des villes, tandis que les villes avec des petites valeurs pour ce score ont une plus grande différence (hiver très froid et été très chaud). La figure 1.11(a) illustre les températures moyennes quotidiennes des villes suivantes : St. Johns,

Winnipeg, Vancouver et Resolute. Le graphique en (b) présente les températures moyennes quotidiennes centrées, c'est-à-dire auxquelles on soustrait la fonction moyenne empirique calculée avec les 35 courbes, de ces 4 villes canadiennes. Nous pouvons voir qu'en (a), les températures de la ville de Vancouver ne varient pas beaucoup au cours de l'année comparativement aux 3 autres villes canadiennes. De plus, la ville de St. Johns a un hiver un peu plus chaud et un été un peu plus froid que les autres villes, alors que la ville de Winnipeg a un hiver moins froid que la ville de Resolute mais un été plus chaud que les autres villes. On s'attendait à de tels résultats puisque la ville de Winnipeg a un score négatif élevé pour la deuxième composante principale tandis que la ville de St. Johns a un score positif élevé. La prochaine section présente une autre méthode qui permet de réduire la dimension des données.

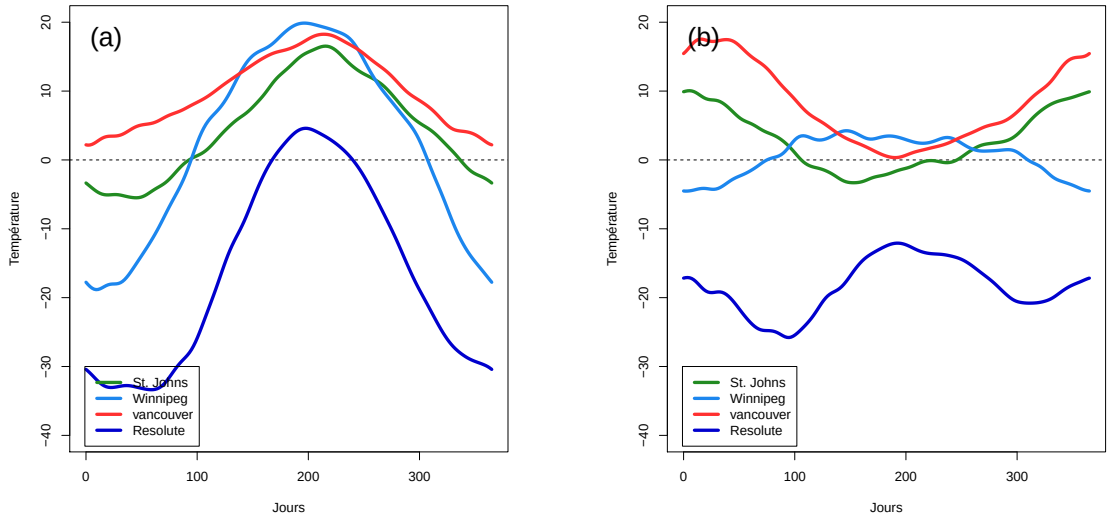

Figure 1.11: (a) Les températures moyennes quotidiennes de 4 villes canadiennes et (b) les courbes de températures de ces 4 villes centrées.

1.3.4 Méthode des moindres carrés partiels dans le cas multivarié

Comme l'analyse en composantes principales, la méthode des moindres carrés partiels (MMCP) est une méthode de réduction de dimension qui identifie un nouvel ensemble de directions $\mathbf{z}_1, ..., \mathbf{z}_q$ qui sont des combinaisons linéaires des prédicteurs originaux, puis ajuste un modèle linéaire via les moindres carrés en utilisant ces directions. La différence principale avec l'ACP est que la MMCP suppose qu'on observe une variable réponse $\mathbf{y} = (y_1, ..., y_n)^T$ et elle tente de trouver des directions qui aident à expliquer à la fois la variable réponse et les prédicteurs, c'est-à-dire de maximiser la covariation entre $\mathbf{X}$ et $\mathbf{y}$. La MMCP le fait en plaçant plus de poids sur les prédicteurs les plus fortement corrélés avec $\mathbf{y}$. Tout d'abord, on suppose que le vecteur $\mathbf{y}$ est centré et les d prédicteurs sont centrés et réduits pour que chaque colonne de $\mathbf{X}$ ait une moyenne nulle et une variance unitaire, et ce pour les mêmes raisons que l'ACP. Notons par $\hat{\phi}_{1j}$ le coefficient de la projection de $\mathbf{y}$ sur $\mathbf{x}_j$, i.e. $\hat{\phi}_{1j} = < \mathbf{x}_j, \mathbf{y} >$, pour $j = 1, ..., d$. L'idée derrière les moindres carrés partiels est d'obtenir la première direction $\mathbf{z}_1$ en faisant une régression de $\mathbf{y}$ sur $\mathbf{x}_1$ pour calculer $\hat{\phi}_{11}$, une régression de $\mathbf{y}$ sur $\mathbf{x}_2$ pour calculer $\hat{\phi}_{12}$ et ainsi de suite, jusqu'à ce qu'on obtienne une régression de $\mathbf{y}$ sur $\mathbf{x}_d$ pour calculer $\hat{\phi}_{1d}$. Nous obtenons donc la première direction, $\mathbf{z}_1 = \sum_{j=1}^{d} \phi_{1j}\mathbf{x}_d$. Pour obtenir la deuxième direction $\mathbf{z}_2$, nous régressons chaque prédicteur sur $\mathbf{z}_1$ et nous allons prendre en considération les résidus. Ces résidus peuvent être interprétés comme les informations restantes qui n'ont pas été expliquées par $\mathbf{z}_1$. Nous calculons ensuite $\mathbf{z}_2$ en utilisant les données orthogonalisées de la même manière que $\mathbf{z}_1$ a été calculée basée sur les données originales. Cette approche peut être répétée jusqu'à ce que $Q \leq d$ directions aient été obtenues. On obtient finalement une séquence de directions orthogonales $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_Q$. Si nous devions construire toutes les directions telles que $Q = d$, nous obtiendrions une solution équivalente aux estimations habituelles des moindres carrés. Dans notre contexte, nous nous intéressons seulement

aux directions $\mathbf{z}_1, ..., \mathbf{z}_Q, Q < d$. La procédure que nous allons utiliser est :

Étape 1. Standardiser chaque $\mathbf{x}_j$ pour avoir une moyenne nulle et une variance unitaire. Poser $\hat{\mathbf{y}}^{(0)} = \bar{y}\mathbf{1}$ et $\mathbf{x}_j^{(0)} = \mathbf{x}_j, j = 1, ..., d$.

Étape 2. Pour $q = 1, ..., d$, la direction obtenue est $\mathbf{z}_q = \sum_{j=1}^{d} \hat{\phi}_{qj} \mathbf{x}_j^{(q-1)}$,
où $\hat{\phi}_{qj} =< \mathbf{x}_j^{(q-1)}, \mathbf{y} >$.

Étape 3. Sauvegarder la séquence de vecteurs $\mathbf{z}_1, ..., \mathbf{z}_Q$.

La procédure complète de la MMCP est décrite dans «Elements of Statistical Learning» de Hastie *et al.* (2001).

CHAPITRE II

MÉTHODES DE CLASSIFICATION NON-PARAMÉTRIQUES POUR DES DONNÉES FONCTIONNELLES

Dans ce chapitre, nous allons présenter une méthode non-paramétrique, développée par Ferraty et Vieu (2006), qui a pour but de faire de la classification supervisée de données fonctionnelles. La méthode est basée sur l'estimateur de Nadaraya-Watson et fait appel à la méthode des K plus proches voisins (K-PPV) pour la sélection du paramètre d'ajustement contrôlant l'étendue du noyau. De plus, nous allons introduire différentes semi-métriques permettant de mesurer la proximité de deux fonctions.

2.1 Introduction au problème de classification

Une variable qualitative (aussi appelée variable catégorielle ou bien facteur) prend pour valeurs des catégories, des classes ou bien des niveaux. Le problème consistant à prédire une variable réponse catégorielle est appelé un problème de classification. L'objectif de la classification est principalement de définir des règles permettant de classer des nouvelles observations dont la classe est inconnue à partir de variables explicatives. Notons que lorsqu'il n'y a que deux classes, on parle de classification binaire et lorsqu'il y a plus de deux classes, on parle de classification en classes multiples. En apprentissage supervisé, on dispose d'un en-

semble de données, appelé données d'entraînement, consistant en n réalisations indépendantes du vecteur (X, Y), où X représente la variable explicative et Y la variable réponse. On va supposer que Y est une variable catégorielle univariée tandis que la variable X peut prendre plusieurs formes : X est un vecteur dans un contexte d'analyse multivariée et une fonction dans un contexte d'analyse fonctionnelle. Pour distinguer ces deux contextes, on notera X une variable multivariée et $\mathcal{X}$ une variable fonctionnelle. Les données d'entraînement sont utilisées afin de déterminer un ensemble de règles permettant de classer des objets dans des classes prédéfinies. Il existe plusieurs méthodes de classification supervisée ; par exemple, la régression logistique, les arbres de décision, les forêts aléatoires et les machines à vecteurs de support. Contrairement à l'approche supervisée, l'ensemble de données utilisé en apprentissage non supervisé est seulement composé de variables explicatives, c'est-à-dire qu'il n'y a pas de variables réponses dans le jeu de données. Le but de l'apprentissage non supervisé est d'explorer la structure des données, par exemple en partionnant les données en groupes homogènes. Dans ce mémoire, nous allons présenter le problème de la classification supervisée dans le cas où la variable explicative est fonctionnelle.

Soit $\mathcal{X}$ une variable aléatoire fonctionnelle à valeurs dans un espace semi-métrique (E, d) où E est un ensemble muni d'une semi-métrique d, Y une variable réponse et un échantillon $(\mathcal{X}_i, Y_i)_{i=1,...,n}$ de n paires indépendantes et identiquement distribuées à $(\mathcal{X}, Y)$ prenant valeurs dans $E \times \bar{G}$, avec $\bar{G} = (1, ..., G)$ où G est le nombre de classes. Nous allons utiliser la notation (x_i, y_i) pour décrire l'observation du couple $(\mathcal{X}_i, Y_i)$. Si on observe une nouvelle fonction x dont on ne connait pas la classe, la meilleure façon de prédire sa classe est d'utiliser le classifieur de Bayes si nous connaissons les probabilités a posteriori :

$$p_g(x) = P(Y = g \mid \mathcal{X} = x),\ g \in \bar{G}.$$

La règle de Bayes consiste à prédire la classe d'une nouvelle courbe observée x en

choisissant la classe ayant la probabilité a posteriori la plus élevée, i.e. la classe

$$\hat{y}(x) = \arg\max_{g \in \bar{G}} p_g(x).$$

Dans le cas de la classification binaire, on peut définir le classifieur de Bayes $h^* : E \to \{0, 1\}$ comme suit :

$$h^*(x) = \begin{cases} 1 & \text{si } P(Y = 1 \mid \mathcal{X} = x) \geq \frac{1}{2} \\ 0 & \text{sinon.} \end{cases}$$

Nous allons maintenant donner une preuve de l'optimalité du classifieur de Bayes. Pour ce faire, nous allons montrer que la probabilité d'erreur de classification de h^* est inférieure ou égale à celle de n'importe quel autre classifieur. Notons que la probabilité d'erreur de classification d'un classifieur h est :

$$P(h(\mathcal{X}) \neq Y \mid \mathcal{X} = x) = 1 - P(h(\mathcal{X}) = Y \mid \mathcal{X} = x).$$

Théorème 1. *Soit le classifieur de Bayes h^* et un classifieur $h : E \to \{0, 1\}$, alors*

$$P(h(\mathcal{X}) \neq Y \mid \mathcal{X} = x) \geq P(h^*(\mathcal{X}) \neq Y \mid \mathcal{X} = x),$$

pour tout $x \in E$.

Démonstration. La probabilité d'erreur de classification d'un classifieur h pour une fonction fixe x peut s'écrire :

$$\begin{aligned} &P(h(\mathcal{X}) \neq Y \mid \mathcal{X} = x) \\ &\qquad = 1 - P(h(\mathcal{X}) = Y \mid \mathcal{X} = x) \\ &\qquad = 1 - P(h(\mathcal{X}) = 1, Y = 1 \mid \mathcal{X} = x) - P(h(\mathcal{X}) = 0, Y = 0 \mid \mathcal{X} = x) \\ &\qquad = 1 - \mathbb{1}_{h(x)=1} P(Y = 1 \mid \mathcal{X} = x) - \mathbb{1}_{h(x)=0} P(Y = 0 \mid \mathcal{X} = x) \\ &\qquad = 1 - \mathbb{1}_{h(x)=1} P(Y = 1 \mid \mathcal{X} = x) - \mathbb{1}_{h(x)=0} (1 - P(Y = 1 \mid \mathcal{X} = x)), \end{aligned}$$

où $\mathbb{1}_A$ est la fonction indicatrice qui vaut 1 lorsque A est vrai et 0 sinon. Comparons la probabilité d'erreur de classification d'un classifieur quelconque h au classifieur h^* :

$$\begin{aligned} &P(h(X) \neq Y \mid X = x) - P(h^*(X) \neq Y \mid X = x) \\ &\qquad = P(Y = 1 \mid X = x)(\mathbb{1}_{h^*(x)=1} - \mathbb{1}_{h(x)=1}) \\ &\qquad + (1 - P(Y = 1 \mid \mathcal{X} = x))(\mathbb{1}_{h^*(x)=0} - \mathbb{1}_{h(x)=0}). \end{aligned}$$

Notons que $\mathbb{1}_{h^*(x)=0} = 1 - \mathbb{1}_{h^*(x)=1}$ et $\mathbb{1}_{h(x)=0} = 1 - \mathbb{1}_{h(x)=1}$, le terme de droite de l'équation précédente peut donc s'écrire comme :

$$(2P(Y = 1 \mid \mathcal{X} = x) - 1)(\mathbb{1}_{h^*(x)=1} - \mathbb{1}_{h(x)=1}). \tag{2.1}$$

L'expression 2.1 est toujours plus grande ou égale à 0. En effet, considérons le cas où $\mathbb{1}_{h^*(x)=1} \neq \mathbb{1}_{h(x)=1}$. Si $h^*(x) = 1$, alors $(\mathbb{1}_{h^*(x)=1} - \mathbb{1}_{h(x)=1}) = 1 > 0$ et $P(Y = 1 \mid \mathcal{X} = x) \geq 1/2$, ce qui implique que $2P(Y = 1 \mid \mathcal{X} = x) - 1 \geq 0$ tandis que si $h^*(x) = 0$, alors $(\mathbb{1}_{h^*(x)=1} - \mathbb{1}_{h(x)=1}) = -1 < 0$ et $P(Y = 1 \mid \mathcal{X} = x) \leq 1/2$, ce qui implique que $2P(Y = 1 \mid \mathcal{X} = x) - 1 \leq 0$. Dans le cas où $\mathbb{1}_{h^*(x)=1} = \mathbb{1}_{h(x)=1}$, l'expression (2.1) est égale à 0.

On obtient donc que :

$$P(h(\mathcal{X}) \neq Y \mid \mathcal{X} = x) \geq P(h^*(\mathcal{X}) \neq Y \mid \mathcal{X} = x), \forall x \in E.$$

□

En pratique, on ne connaît habituellement pas les probabilités a posteriori. Il existe plusieurs méthodes pour les estimer dont une étant la régression non-paramétrique. Remarquons que la prédiction d'une variable réponse Y étant donné une variable fonctionnelle $\mathcal{X}$ amène à se concentrer sur l'espérance conditionnelle puisque l'on peut réécrire la probabilité p_g sous forme d'une espérance conditionnelle :

$$p_g(x) = E(\mathbb{1}_{Y=g} \mid \mathcal{X} = x), g \in \bar{G}. \tag{2.2}$$

Cette espérance conditionnelle étant une fonction de régression, on peut donc l'estimer à l'aide d'une méthode de régression non-paramétrique ; nous présenterons cette approche à la section 2.4.3.

2.2 Méthode des K plus proches voisins

Nous aurons besoin de la méthode des K plus proches voisins dans la construction de la règle de classification. La méthode des K plus proches voisins est une méthode non-paramétrique utilisée pour la classification et la régression. Pour une nouvelle observation x dont on souhaite prédire la classe, l'algorithme des K-PPV identifie les K observations de l'ensemble des données d'entraînement qui sont les plus proches en termes de distance euclidienne ou de toutes autres semi-métriques de x : ces K observations sont appelées les K plus proches voisins de x. La nouvelle observation est classée selon un vote de majorité de ses voisins : l'observation sera attribuée à la classe la plus courante parmi celles de ses K voisins les plus proches. Par exemple, si $K = 1$, alors la nouvelle observation est simplement assignée à la classe de son voisin le plus proche. La méthode des K plus proches voisins basée sur la distance euclidienne dans le cas binaire, $Y \in \{\text{vert}, \text{bleu}\}$, et pour $X \in \mathbb{R}^2$, est illustré à la figure 2.1. Les variables explicatives $x_1, ..., x_{16}$ sont représentées par les points et les variables réponses $y_1, ..., y_{16}$ par la couleur des points. Le point turquoise représente une nouvelle observation x à classer. Pour $K = 5$ voisins (cercle noir), la classe prédite de la nouvelle observation est bleue, car $N_{\text{bleu}} = 3 > N_{\text{vert}} = 2$, où N_i représente le nombre d'observations dans la classe i. Pour $K = 9$ voisins (cercle rouge), la classe prédite de la nouvelle observation est verte, car $N_{\text{vert}} = 6 > N_{\text{bleu}} = 3$; la classe prépondérante est la classe verte.

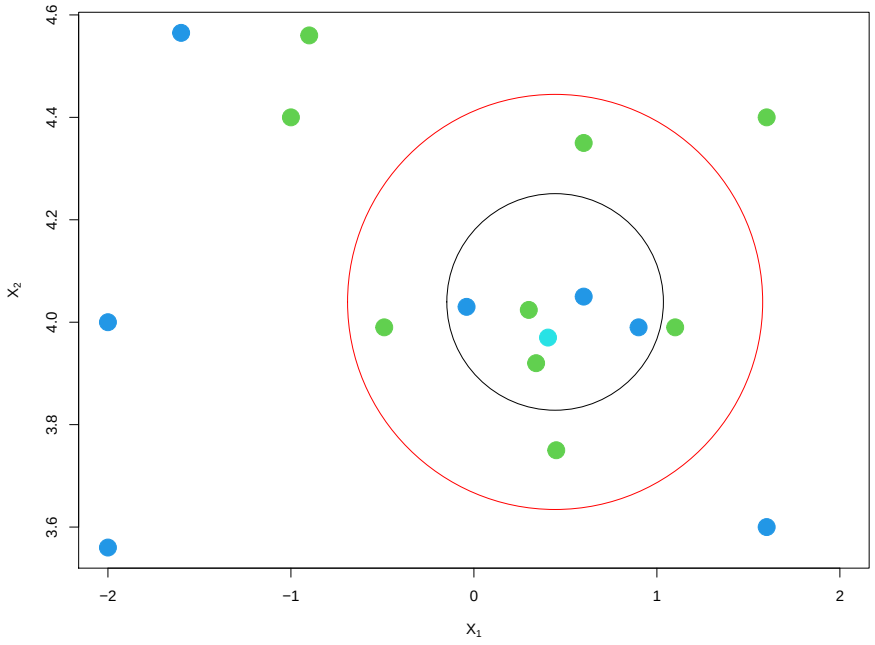

Figure 2.1: Illustration de la classification binaire par la méthode des K plus proches voisins avec $X \in \mathbb{R}^2$.

2.3 Les semi-métriques pour les données fonctionnelles

La méthode de classification qui est présentée dans ce chapitre necéssite la connaissance de la distance entre chacune des observations fonctionnelles de nos données. On a donc besoin de définir une notion de distance pour des fonctions. Il en existe plusieurs, on considère la distance L^2 comme étant la distance classique en analyse de données fonctionnelles. Soit deux courbes x et x', on définit la distance euclidienne comme suit :

$$d^{L^2}(x, x') = \sqrt{\int \Big(x(t) - x'(t)\Big)^2 dt}.$$

Nous allons introduire trois familles de semi-métriques qui serviront à mesurer la distance entre deux fonctions. La première famille de semi-métrique que nous al-

lons voir est basée sur l'analyse en composantes principales fonctionnelles (ACPF), la deuxième sur la méthode des moindres carrés partiels (MMCP) et la troisième sur les dérivées. Les deux premières sont bien adaptées aux courbes rugueuses tandis que la troisième est adaptée aux données lisses. Rappelons tout d'abord la définition d'une semi-métrique. Selon Ferraty et Vieu (2006), d est une semi-métrique sur un espace F si :

$$1.\ \forall x \in F, d(x,x) = 0;$$

$$2.\ \forall x_i, x_j, x_k \in F, d(x_i,x_k) \leq d(x_i,x_j) + d(x_j,x_k).$$

Notons qu'une semi-métrique n'est pas une métrique puisqu'elle ne doit pas nécessairement respecter la condition : $\forall x_i, x_j \in F, d(x_i,x_j) = 0 \implies x_i = x_j$. Nous allons voir en premier lieu la semi-métrique d_q^{ACP} basée sur la méthode de l'ACPF. Nous avons vu au chapitre 1 que l'ACPF peut être utilisée afin de projeter de façon optimale les données fonctionnelles dans un espace de dimension finie q, où q représente le nombre de composantes principales fonctionnelles. La distance d_q^{ACP} est obtenue en prenant la distance euclidienne entre les courbes projetées par l'ACPF en dimension q. En effet, la semi-métrique basée sur l'ACPF est définie comme suit :

$$\begin{aligned} d_q^{ACP}(x,x') &= \sqrt{\sum_{k=1}^{q} \Big(\langle x, \xi_k \rangle - \langle x', \xi_k \rangle \Big)^2} \\ &= \sqrt{\sum_{k=1}^{q} \left(\int [x(t) - x'(t)] \xi_k(t) dt \right)^2} \end{aligned} \tag{2.3}$$

où ξ_k est la $k^{\text{ème}}$ fonction propre de l'opérateur de covariance de $\mathcal{X}$. En pratique, les fonctions ξ_k sont estimées à l'aide des observations $x_1, ..., x_n$ tel qu'expliqué à la section 1.3.2 du chapitre 1. Nous allons voir un example d'un calcul de cette semi-métrique en dimension $q = 2$. Nous allons utiliser le jeu de données *tecator* tiré du paquet *fda.usc* de R. Chaque courbe spectrométrique observée correspond

à l'absorbance mesurée sur 100 longueurs d'onde (la longueur d'onde varie de 850 nm à 1050 nm). Si nous voulons calculer la distance entre la 1$^{\text{ère}}$ courbe et la 100$^{\text{ème}}$ courbe du jeu de données illustrées respectivement en rouge et en noire à la figure 2.2(a), nous devons projeter chacune des courbes dans un espace de dimension 2. On obtient ainsi deux points dans un plan défini par les 2 premières composantes principales, comme l'on peut voir à la figure 2.2(b) : la courbe rouge est représentée par le point en rouge et la courbe noire est représentée par le point en noir. Ensuite, la distance est obtenue en calculant la distance euclidienne entre ces deux points (illustrée par la ligne bleue). On obtient $d_2^{ACP}(x_1, x_{100}) = 10$.

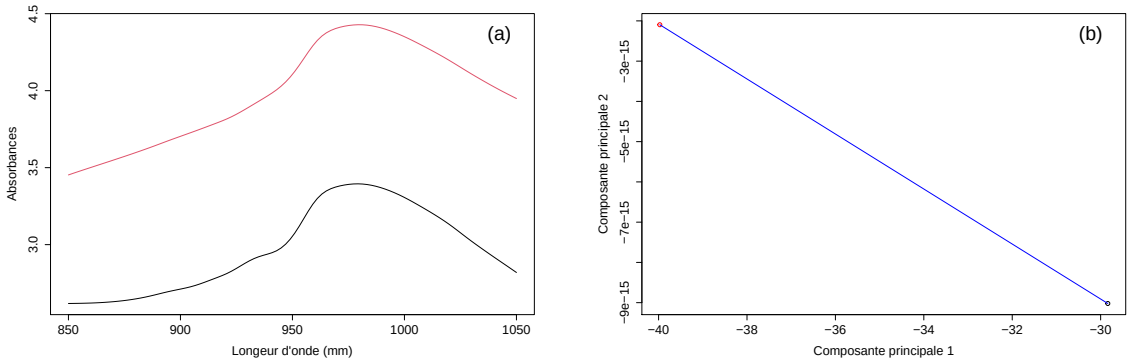

Figure 2.2: La figure (a) représente la 1$^{\text{ère}}$ courbe et la 100$^{\text{ème}}$ courbe tirées du jeu de données *tetacor* du paquet *fda.usc* de R, la figure (b) illustre la projection de ces deux courbes dans un espace de dimension 2.

Nous allons voir la deuxième famille de semi-métrique d_q^{MMCP} basée sur la méthode des moindres carrés partiels. Le principe de cette semi-métrique est similaire à la semi-métrique d_q^{ACP}. Les courbes sont projetées dans un espace de dimension q par la MMCP. Encore une fois, la distance d_q^{MMCP} est obtenue en calculant la distance euclidienne entre les courbes projetées en dimension q. La formule de la

semi-métrique basée sur la MMCP que nous allons utiliser est :

$$d_q^{MMCP}(x, x') = \sqrt{\sum_{k=1}^{p} \left(\int [x(t) - x'(t)] \mathbf{z}_k^q dt \right)^2},$$

où $\mathbf{z}_k^q$ est la $k^{\text{ème}}$ composante obtenue par la MMCP. Les composantes $\mathbf{z}_k^q$ peuvent être estimées à l'aide des observations $(x_1, y_1), ..., (x_n, y_n)$ tel que mentionné à la section 1.3.4.

La dernière famille de semi-métrique, notée d_q^{deriv}, correspond à la distance L^2 entre la dérivée d'ordre q de deux courbes :

$$d_q^{deriv}(x, x') = \sqrt{\int \left(x^{(q)}(t) - x'^{(q)}(t) \right)^2 dt}, \tag{2.4}$$

où $x^{(q)}$ et $x'^{(q)}$ représentent la $q^{\text{ème}}$ dérivée des courbes x et x' respectivement. Notons que lorsque $q = 0$, on obtient la distance L^2, notée d_0^{deriv}. La figure 2.3 illustre un exemple du calcul de cette semi-métrique à l'aide des deux courbes du jeu de données *tetacor* présentées précedemment, pour $q = 0, 1, 2$ et 3. La semi-métrique peut être approximée par la somme de la longueur au carré des segments en turquoises. On obtient par exemple $d_0^{deriv}(x_1, x_{100}) = 14$. Notons que plus l'ordre de la dérivée est élevée, et plus la semi-métrique devient petite.

2.4 Les méthodes par noyau pour les données fonctionnelles

Nous allons maintenant définir ce qu'est une fonction noyau. Une fonction noyau, aussi appelée noyau, est une fonction de pondération utilisée entre autres dans les méthodes d'estimation non-paramétrique. Les noyaux peuvent être utilisés par exemple dans la régression non-paramétrique. Dans notre cas, nous allons utiliser la régression non-paramétrique par noyau pour estimer les probabilités p_g, $g \in \bar{G}$, puisque ces probabilités comme nous l'avons présenté à l'équation 2.2, peuvent s'écrire comme une espérance conditionnelle.

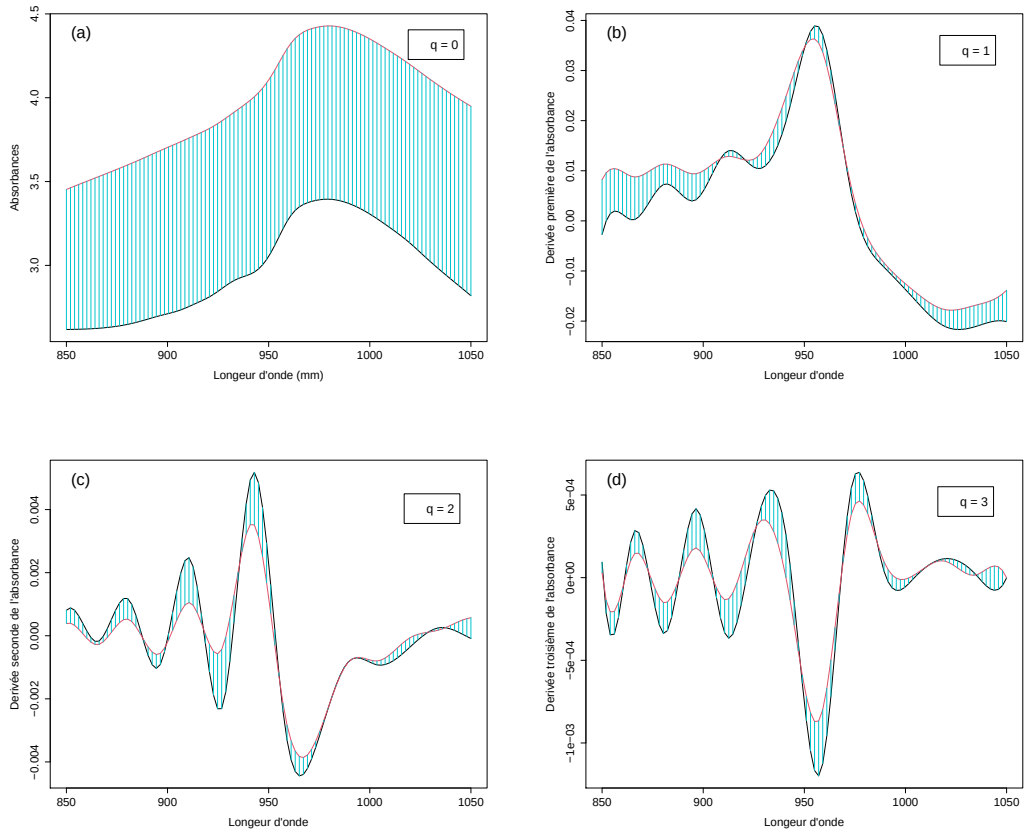

Figure 2.3: Les figures (a), (b), (c) et (d) illustrent respectivement la dérivée d'ordre $q = 0$, la dérivée première ($q = 1$), la dérivée seconde ($q = 2$) et la dérivée troisième ($q = 3$) des deux courbes tirées du jeu de données *tetacor*. Les distances $d_0^{deriv}, d_1^{deriv}, d_2^{deriv}, d_3^{deriv}$ peuvent être approximées par la somme de la longueur au carré des segments en turquoises.

2.4.1 Les fonctions noyaux définies sur des scalaires

Nous allons présenter dans cette section plusieurs types de noyaux. Une fonction noyau $K : \mathbb{R} \to \mathbb{R}^+$ est toute fonction intégrable à valeur réelle non-négative qui

satistfait

$$\int_{-\infty}^{\infty} K(u)du = 1.$$

Une fonction noyau telle que

$$K(-u) = K(u), \forall u \in \mathbb{R}$$

est dite symétrique. Il existe plusieurs noyaux dont les principaux sont : uniforme, triangle, epanechnikov, quadratique et gaussien. Le tableau 2.1 présente la forme symétrique de chacun de ces noyaux. On remarque que les quatre premiers noyaux ont un support compact (fermé et borné), i.e. que l'ensemble $\{u \in \mathbb{R} : K(u) > 0\}$ est compact, tandis que le noyau gaussien possède un support infini.

Noyau	$K(u)$
Uniforme	$\frac{1}{2}, u \in [-1, 1]$
Triangle	$(1-u), u \in [-1, 1]$
Epanechnikov	$\frac{3}{4}(1-u^2), u \in [-1, 1]$
Quadratique	$\frac{15}{16}(1-u^2)^2, u \in [-1, 1]$
Gaussien	$\frac{1}{\sqrt{\pi}} exp\left(\frac{-u^2}{2}\right), u \in \mathbb{R}$

Tableau 2.1: Présentation de fonctions noyaux symétriques.

2.4.2 Estimation par noyau

Nous allons maintenant présenter la méthode d'estimation par noyau d'une fonction de densité. Cette méthode nous sera utile dans la section suivante afin de présenter la méthode de régression par noyau. Soit $x_1, x_2, ..., x_N$ des réalisations indépendantes d'une variable aléatoire univariée X dont la fonction de densité f est inconnue. L'objectif est d'estimer la fonction f. L'estimateur par noyau de la

fonction de densité est :

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right),$$

où $K(\cdot)$ est un noyau et h est un paramètre d'échelle, nommé largeur de la fenêtre (de l'anglais "bandwidth"), qui agit comme un paramètre de lissage. Par exemple, pour un noyau gaussien, l'expression $\frac{1}{h}K\left(\frac{x-x_i}{h}\right)$ représente la fonction de densité d'une variable aléatoire $\mathrm{N}(x_i, h^2)$ évalué en x, le paramètre h peut être vu comme la variance du noyau, tandis que pour un noyau uniforme, elle représente la fonction de densité d'une variable aléatoire $U(x_i - h, x_i + h)$ évaluée en x et donc h peut être vu comme la largeur du noyau. Cette dernière interprétation de h peut être obetenue pour tous les noyaux à support compact. Lorsqu'on estime la valeur de f en un point x, si la valeur de h est grande, alors même les données éloignées de x auront une influence sur $\hat{f}_h(x)$ alors que si la valeur de h est petite, seules les observations proches de x auront une influence sur l'estimation. Dans le second cas, on dira que l'estimateur est local. L'estimateur par noyau est similaire à celui obtenu avec un histogramme. En effet, dans un histogramme, on partitionne le support de f en intervalles contigus et la fonction de densité en un point x est estimée par la proportion des observations x_i qui se trouvent dans l'intervalle contenant x. Quant à elle, la méthode par noyau consiste à centrer un noyau K de largeur (variance) h sur chaque observation x_i et l'estimation de la fonction de densité est obtenue en faisant la moyenne de ces fonctions noyaux évaluées en x. La figure 2.4 illustre un jeu de données composé de quatre observations : $x_1 = 2, x_2 = 3, x_3 = 5$ et $x_4 = 9$. En (a), un noyau gaussien est centré à chaque observation et les quatre noyaux sont illustrées par les courbes en pointillées (bleu). On fait la moyenne des quatre noyaux gaussiens évaluées en x. La fonction ainsi obtenue (illustrée en noire) est l'estimation de la fonction de densité. La figure 2.4(b) montre l'effet du paramètre h sur l'estimation. On remarque que plus h est grand et plus la fonction $\hat{f}_h(x)$ est lisse. Les figures 2.4(c) et (d) illustrent l'estimation de la densité avec une largeur

de fenêtre de 0.6 à l'aide d'un noyau triangle et epanechnikov respectivement. On remarque que le noyau gaussien n'accorde jamais un poids nul aux observations tandis que les noyaux triangle et epanechnikov qui sont à support fini, accordent un poids nul aux observations en dehors de leur support.

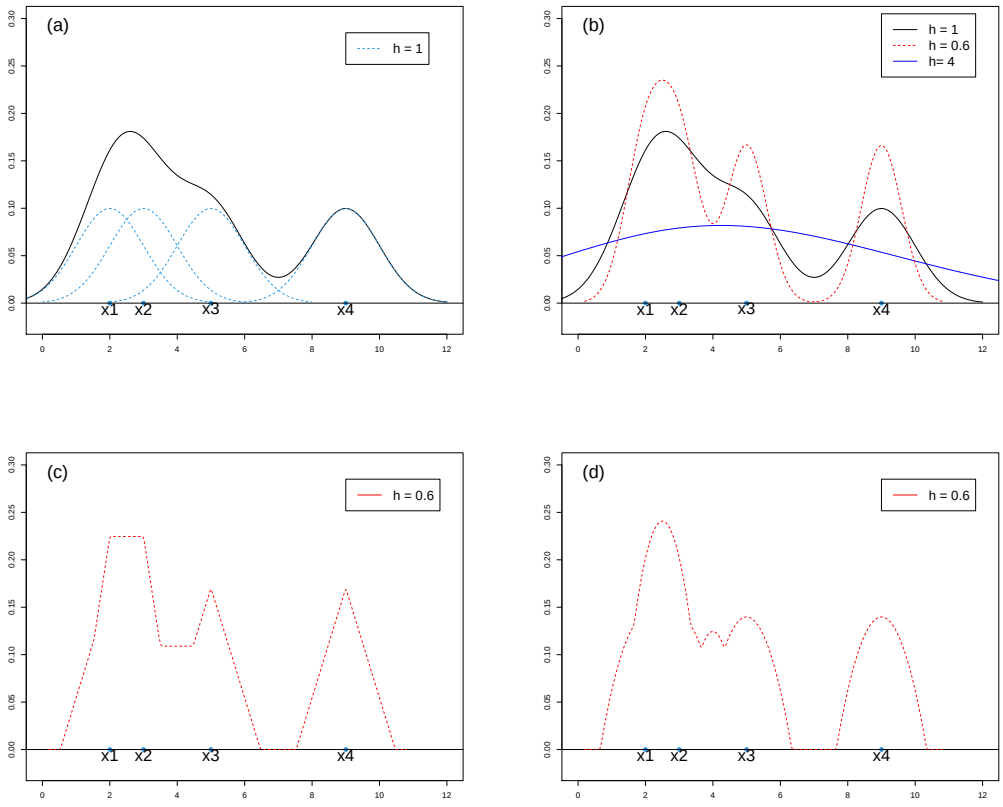

Figure 2.4: Le graphique (a) illustre l'estimation d'une fonction de densité à partir d'un noyau gaussien, la figure (b) présente l'effet de la variation de la largeur de la fenêtre sur l'estimation de la fonction de densité, les figures (c) et (d) illustrent l'estimation de la fonction de densité construite à l'aide d'un noyau triangle et d'un noyau epanechnikov respectivement.

2.4.3 Régression non-paramétrique

Dans cette section, nous présentons la régression non-paramétrique par noyau. Les méthodes non-paramétriques permettent une modélisation flexible de la relation entre une variable X et une variable réponse Y, car elles ne requièrent pas la spécification d'un modèle gouverné par un nombre fini de paramètres. Dans une régression, l'objectif est d'estimer l'espérance conditionnelle de Y sachant $X = x$ qu'on écrit comme une fonction de x :

$$E(Y|X = x) = m(x),$$

où m est une fonction inconnue. Considérons tout d'abord le cas où X est une variable aléatoire univariée, on peut alors écrire :

$$E(Y \mid X = x) = \int y f_{Y|X}(y \mid x)dy = \int y \frac{f_{X,Y}(x, y)}{f_X(x)}dy.$$

Il est possible d'estimer les fonctions de densité $f_X(x)$ et $f_{X,Y}(x, y)$ à l'aide des estimateurs par noyau introduits à la section 2.4.2. En effet, on a directement que :

$$\hat{f}_X(x) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(x - x_i\right), \text{où } K_h\left(x - x_i\right) = \frac{1}{h}K\left(\frac{x - x_i}{h}\right),$$

et on peut estimer la fonction de densité bivariée $f_{X,Y}(x, y)$ par

$$\hat{f}_{X,Y}(x, y) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(x - x_i\right) K_h\left(y - y_i\right).$$

On peut donc estimer $m(x)$ par :

$$\begin{aligned}
\hat{m}(x) &= \int y \frac{\hat{f}_{X,Y}(x, y)}{\hat{f}_X(x)}dy \\
&= \int \frac{y \sum_{i=1}^{n} K_h\left(x - x_i\right) K_h\left(y - y_i\right)}{\sum_{i=1}^{n} K_h\left(x - x_i\right)}dy \\
&= \frac{\sum_{i=1}^{n} K_h\left(x - x_i\right) \int y\, K_h\left(y - y_i\right) dy}{\sum_{i=1}^{n} K_h\left(x - x_i\right)} \\
&= \frac{\sum_{i=1}^{n} K_h\left(x - x_i\right) y_i}{\sum_{i=1}^{n} K_h\left(x - x_i\right)}.
\end{aligned} \tag{2.5}$$

Notons que l'expression $\int y\, K_h\,(y - y_i)\, dy$ de l'équation 2.5 est égale à y_i car en faisant le changement de variable $\psi = y - y_i$, on obtient

$$\int (\psi + y_i) K_h(\psi) d\psi = \int \psi K_h(\psi) d\psi + y_i \int K_h(\psi) d\psi.$$

Le premier terme à droite de l'égalité est égale à 0, car on a supposé la fonction noyau symétrique et l'intégrale du deuxième terme est égal à 1 par définition d'une fonction noyau. L'estimateur $\hat{m}(x)$ ainsi obtenu est appelé l'estimateur de Nadaraya-Watson. On peut le réécrire sous une forme de moyenne pondérée des valeurs de y_i :

$$\hat{m}(x) = \sum_{i=1}^{n} w_i y_i,$$

avec les poids :

$$w_i = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right)}.$$

On estime donc m par une moyenne pondérée des y_i, en utilisant un noyau afin de calculer le poids w_i associé à chaque observation y_i. Le poids w_i dépend du type de noyau utilisé ainsi que de la distance entre x et x_i. Nous allons maintenant présenter la généralisation de l'estimateur de Nadaraya-Watson au cas où les données $x_1, ..., x_n$ sont fonctionnelles. Dans le cas univarié, on remarque qu'on applique le noyau K au scalaire $\frac{x-x_i}{h} \in \mathbb{R}$ qui joue le rôle de distance entre x et x_i. Dans le cas fonctionnel, il n'est pas possible d'appliquer directement un noyau à $\frac{x-x_i}{h}$, car cette expression n'est plus un scalaire. Pour passer du cas univarié au cas fonctionnel, nous allons d'abord transformer $\frac{x-x_i}{h}$ en un scalaire afin de pouvoir appliquer un noyau à ce dernier. Il est possible de faire la transformation grâce aux semi-métriques vues à la section 2.3. En effet, il nous suffit d'appliquer le noyau K à $d(\frac{x-x_i}{h})$, où d est une semi-métrique. Notons que les semi-métriques sont toujours des quantités positives. Le noyau K que nous allons utiliser doit donc avoir un support positif, c'est-à-dire, $\{u \in \mathbb{R} \mid K(u) > 0\} \subseteq \mathbb{R}^+$. Cela nous amène à utiliser des noyaux asymétriques. Il est à noter que la méthode de Ferraty

et Viau (2006) exige que les noyaux soient asymétriques et bornés, c'est-à-dire de support $[0, 1]$. Le noyau gaussien ne sera donc pas utilisé par la suite puisqu'il est à support infini. Le tableau 2.2 présente la version asymétrique des noyaux à support fini présentés au tableau 2.1 et la figure 2.5 les illustre. La version fonc-

Noyau asymétrique	$K(u)$
Uniforme	$1, u \in [0, 1]$
Triangle	$2(1-u), u \in [0, 1]$
Epanechnikov	$\frac{3}{2}(1-u^2), u \in [0, 1]$
Quadratique	$\frac{15}{8}(1-u^2)^2, u \in [0, 1]$

Tableau 2.2: Présentation de fonctions noyaux asymétriques.

tionelle de l'estimateur de Nadaraya-Watson peut être utilisée afin d'estimer les probabilités postérieures p_g introduites à l'équation 2.2. En effet, en utilisant, pour $g \in \bar{G}$, $\{\mathbb{1}_{y_i=g}\}_{i=1}^n$ comme variables réponses observées, on obtient :

$$\hat{p}_g(x) = \hat{E}[\mathbb{1}_{y_i=g} \mid \mathcal{X} = x] = \sum_{i=1}^{n} \mathbb{1}_{y_i=g} w_i, \text{ où } w_i = \frac{K\left(\frac{d(x,x_i)}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{d(x,x_j)}{h}\right)} \tag{2.6}$$

et où $K(\cdot)$ est un noyau asymétrique de support $[0, 1]$. Notre objectif est d'estimer les G probabilités a posteriori, ce qui revient à faire G régressions sur des variables réponses binaires. En effet, prenons un exemple où $G = 3$. Afin d'estimer $\hat{p}_1$, on transforme les variables y_i en $z_i = 1$ si $y_i = 1$ et $z_i = 0$ sinon. Ensuite, on fait une regression non-paramétrique fonctionnelle des z_i sur les x_i afin d'obtenir $\hat{p}_1$. On refait la même procédure, en posant respectivement $z_i = \mathbb{1}_{y_i=2}$ et $z_i = \mathbb{1}_{y_i=3}$ pour estimer $\hat{p}_2$ et $\hat{p}_3$.

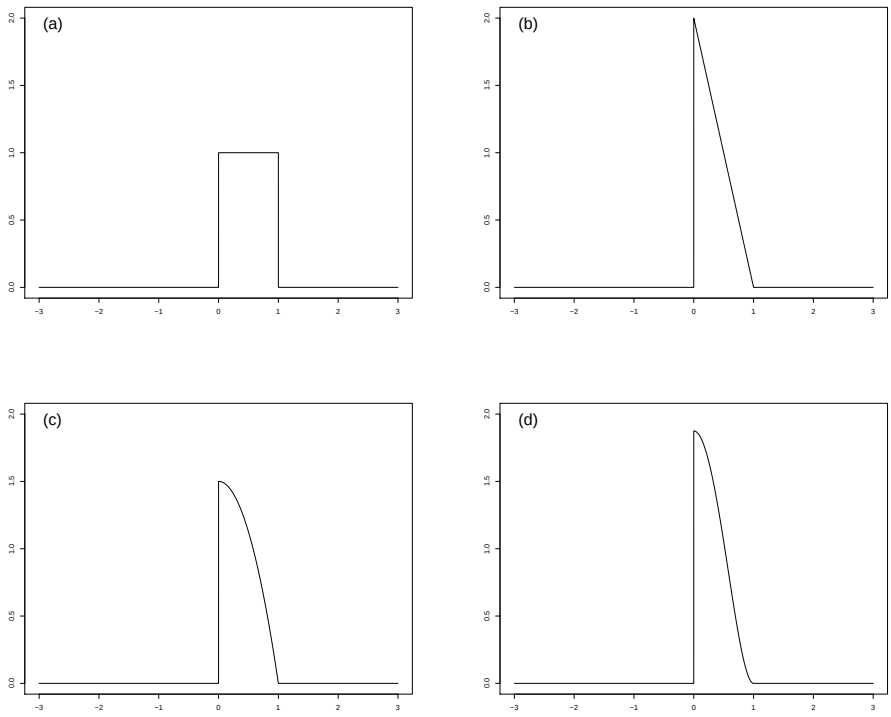

Figure 2.5: Les graphiques (a), (b), (c) et (d) présentent la version asymétrique du noyau uniforme, triangulaire, epanechnikov et quadratique respectivement.

2.5 Méthodes de classification non-paramétriques

Maintenant que nous avons vu l'estimateur de Nadaraya-Watson adapté aux données fonctionnelles et que nous savons comment estimer p_g, nous pouvons prédire la valeur de la réponse catégorielle en fonction de la variable fonctionnelle explicative. Rappelons qu'une fois qu'on a obtenu les G probabilités estimées $(\hat{p}_1, ..., \hat{p}_G)$ par la régression non-paramétrique par noyau, nous pouvons prédire la classe d'une nouvelle courbe observée en choisissant la classe ayant la probabilité a posteriori estimée la plus élevée.

Pour plus de simplicité et pour voir comment fonctionne un tel estimateur, il est

plus simple de réécrire l'estimateur du noyau vu à l'équation 2.6 comme suit :

$$\hat{p}_{g,h}(x) = \sum_{i:y_i=g} w_{i,h}(x),\ g \in \bar{G},\ h > 0,$$

avec

$$w_{i,h}(x) = \frac{K\left(\frac{d(x,x_i)}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{d(x,x_j)}{h}\right)}.$$

Notons que nous avons ajouté un indice h afin de rappeler que le poids $w_{i,h}(x)$ accordé à l'observation x_i dépend de la largeur de la fenêtre h. On remarque que plus x_i est proche de x et plus le poids $w_{i,h}(x)$ est grand. Notons que puisque le support de K est $[0, 1]$, l'estimateur $\hat{p}_g(x)$ ne prend en compte que les observations y_i pour lesquelles les x_i correspondants sont distants de x d'au plus h. En effet, $K\left(\frac{d(x,x_i)}{h}\right) > 0$ si $d(x, x_i) < h$ ce qui implique $w_{i,h}(x) > 0$, tandis que $w_{i,h}(x) = 0$ lorsque $d(x, x_i) > h$. Ainsi, h contrôle le nombre de termes dans la moyenne pondérée, et donc le choix du paramètre h est important dans l'estimation. En d'autres termes, si la largeur de la fenêtre est grande, il y aura plus de courbes qui seront pris en compte dans l'estimation de la classe d'une nouvelle courbe observée.

2.5.1 Comment trouver le paramètre h ?

Il est très important de choisir une bonne valeur du paramètre h, nous allons donc présenter une méthode basée sur la minimisation d'une fonction de perte afin de le choisir judicieusement. Une fonction de perte, dans le cas d'un problème de classification, est une fonction qui pénalise les mauvaises classifications. Une fonction de perte couramment utilisée est la fonction de perte $0-1$ qui correspond au taux de mauvaises classifications d'un classifieur f. Ce taux peut être estimé à l'aide de notre échantillon. En effet, soit un échantillon $(x_1, y_1), ..., (x_n, y_n)$ où $y_i \in \bar{G}$ et soit $\hat{y}_i$ la prédiction associée à la courbe x_i, on estime le taux de

mauvaises classifications par :

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{y_i\neq\hat{y}_i}. \tag{2.7}$$

Plus la valeur de l'expression 2.7 est petite et plus le classifieur est performant.

Soit $H \subset \mathbb{R}^+$ un ensemble de valeurs possibles pour h et soit une fonction de perte Perte(h), on veut choisir le h, noté h_{perte}, qui minimise la fonction Perte(h) :

$$h_{\text{perte}} = \arg\min_{h\in H} \text{Perte}(h).$$

Cette tâche est compliquée puisque h peut prendre une infinité de valeurs. Par contre, comme nous l'avons mentionné à la fin de la section précédente, le paramètre h contrôle le nombre d'observations x_i considérées dans l'estimation et en fait pour un même nombre d'observations considérées, on obtient la même classification. Nous avons donc que plusieurs valeurs de h vont donner le même résulat pour la classification. Par exemple, à la figure 2.6, on s'intéresse au nombre d'observations associées à un poids non nul en fonction de la largeur de la fenêtre, pour des observations $x_1, ..., x_4$ univariées. Les fonctions de densité estimées par l'estimateur par noyau sont obtenues en prenant la moyenne des quatre fonctions noyaux triangulaires évaluées en x pour différentes largeurs de fenêtre $h = 0.25, 0.3, 0.35, 0.40$ (illustrées par les lignes en pointillées de couleur fuschia, noire, bleue et rouge respectivement). On remarque que pour tous les valeurs de h considérés dans l'estimation de $f(x_i)$ est le même : 2 pour x_1 et x_2 et 1 pour x_3 et x_4. Ainsi, lorsque l'on va faire de la classification, on va obtenir différentes valeurs de $\hat{p}_{g,h}$, mais $\arg\max_g \hat{p}_{g,h}$ sera le même pour ces 4 valeurs de h.

Une stratégie afin de simplifier le problème est de donc de remplacer le paramètre h par $h_k \in \mathbb{R}$, un paramètre qui dépend d'une valeur entière $k \in \{1, 2, ..., n\}$. La largeur de fenêtre h_k est choisie telle que :

$$card\,\{i : d(x, x_i) < h_k\} = k,$$

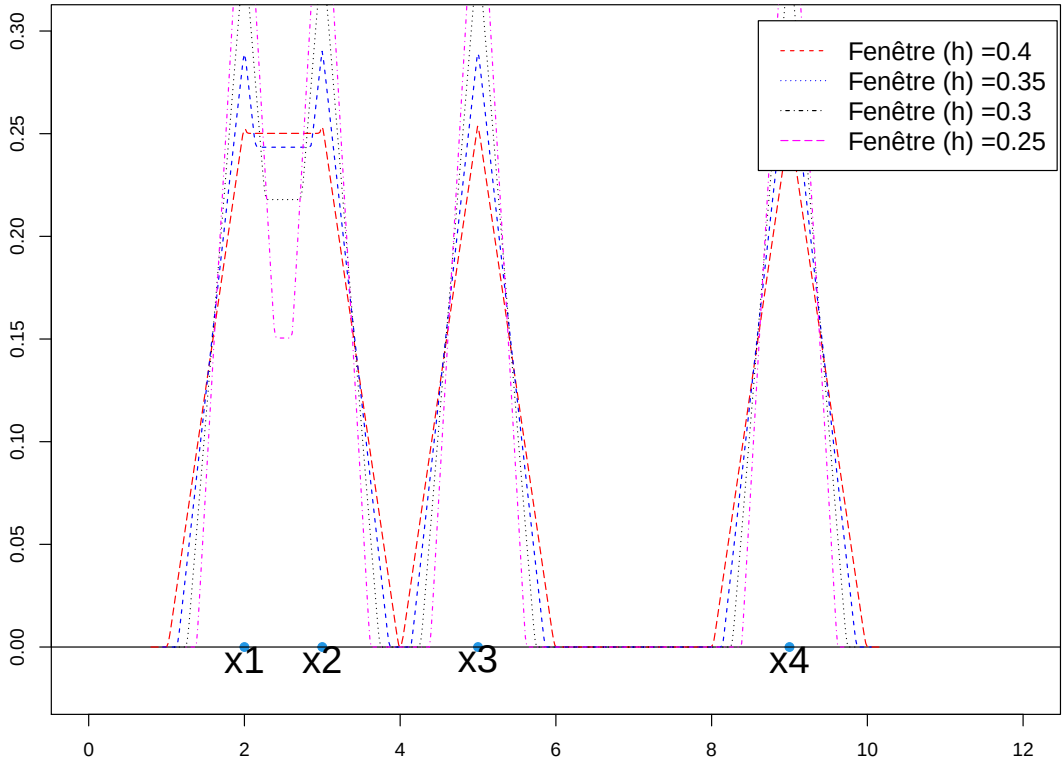

Figure 2.6: Présentation de plusieurs largeurs de fenêtre pour observer le même nombre d'observations pour $h = 0.25$, 0.3, 0.34 et 0.40.

et donc k représente ici le nombre d'observations x_i qui recevront un poids non-nul dans l'estimation $\hat{p}_{g,h_k}(x)$. Il sera donc possible de minimiser la fonction Perte(h) pour $h \in \{h_k\}_{k=1}^{n}$ plutôt que $h \in H$, ce qui revient à trouver la valeur de $k \in \{1, ..., n\}$ qui minimise la fonction Perte(h_k). Ainsi, on peut voir $\hat{p}_{g,h_k}(x)$ comme une version des K plus proches voisins de notre estimateur par noyau. Afin de mettre l'emphase sur ce dernier point, on notera notre estimateur $\hat{p}_{g,k}(x)^{\text{K-PPV}}$

plutôt que $\hat{p}_{g,h_k}(x)$:

$$\hat{p}_{g,k}(x)^{\text{K-PPV}} = \frac{\sum_{i=1}^{n} \mathbb{1}_{y_i=g} K\left(\frac{d(x,x_i)}{h_k}\right)}{\sum_{i=1}^{n} K\left(\frac{d(x,x_i)}{h_k}\right)}.$$

On constate qu'au lieu d'utiliser un simple vote de majorité des k plus proches voisins de x en accordant le même poids à toutes les observations comme dans la méthode classique des K-PPV, l'estimateur par noyau $\hat{p}_{g,k}(x)^{\text{K-PPV}}$ donne un poids en fonction de la distance entre le voisin et x.

On aimerait trouver le paramètre k optimal pour classer une nouvelle observation x, mais comme sa classe nous est inconnue, nous allons plutôt utiliser le k optimal pour l'observation x_{i_0} qui est la plus proche de x, i.e. telle que $i_0 = \arg\min_{i=1,\ldots,n} d(x, x_i)$, car on suppose qu'elle a un comportement semblable à celui de x. Nous allons obtenir le k optimal pour x_{i_0} par validation croisée (VC) :

$$k^{VC}(x_{i_0}) = \arg\min_{k} VC(k, i_0),$$

où notre fonction de perte est :

$$VC(k, i_0) = \sum_{g=1}^{G} \left(\mathbb{1}_{y_{i_0}=g} - \hat{p}_{g,k}^{(-i_0)}(x_{i_0})\right)^2,$$

avec

$$\hat{p}_{g,k}^{(-i_0)}(x_{i_0}) = \frac{\sum_{\{i:\, y_i=g, i\neq i_0\}} K\left(\frac{d(x,x_{i_0})}{h_k(x_{i_0})}\right)}{\sum_{i=1, i\neq i_0}^{n} K\left(\frac{d(x,x_{i_0})}{h_k(x_{i_0})}\right)}.$$

On cherche donc à calculer les probabilités a posteriori $\hat{p}_{g,k}^{(-i_0)}(x_{i_0})$ pour la courbe x_{i_0} sans utiliser la donnée (x_{i_0}, y_{i_0}). Notons que lorsque $\mathbb{1}_{y_{i_0}=g}$ vaut 1, la fonction de perte sera petite si $p_{g,k}^{(-i_0)}(x_{i_0})$ est grande (on aurait une grande probabilité de prédire la classe g pour x_{i_0}), tandis que lorsque $\mathbb{1}_{y_{i_0}=g}$ vaut 0, la fonction de perte sera petite si $p_{g,k}^{(-i_0)}(x_{i_0})$ est petite (on aurait une probabilité faible de prédire la classe g pour x_{i_0}). On remarque donc que cette fonction est differente de la fonction de perte 0-1 introduite en (2.7). Par exemple, considérons le cas de la

classification binaire ($G = 2$) où y_{i_0} est égal à 1, $\hat{p}_{1,k}^{(-i_0)} = 0.5$ et $\hat{p}_{2,k}^{(-i_0)} = 0.5$. Dans ce cas, la fonction de perte est égale à $(1-0.5)^2 + (0-0.5)^2 = 0.5$. Si on avait plutôt $\hat{p}_{1,k}^{(-i_0)} = 0.6$ et $\hat{p}_{2,k}^{(-i_0)} = 0.4$, alors la fonction de perte serait égale à 0.32, tandis que si $\hat{p}_{1,k}^{(-i_0)} = 0.9$ et $\hat{p}_{2,k}^{(-i_0)} = 0.1$, la fonction de perte serait égale à 0.02. On remarque donc que plus on est en train de faire une bonne prédiction, plus la fonction de perte est minimisée, ainsi la fonction VC est belle et bien une fonction de perte. Une fois qu'on a obtenu le paramètre $k^{VC}(x_{i_0})$ par la procédure de validation croisée, on peut trouver le paramètre $h_k^{VC}(x_{i_0})$ correspondant et finalement calculer les probabilités a posteriori :

$$\hat{p}_g^{VC}(x) = \frac{\sum_{i=1}^{n} \mathbb{1}_{(y_i=g)} K\left(\frac{d(x,x_i)}{h_k^{VC}(x_{i_0})}\right)}{\sum_{i=1}^{n} K\left(\frac{d(x,x_i)}{h_k^{VC}(x_{i_0})}\right)}. \tag{2.8}$$

En résumé, pour prédire la classe d'une nouvelle observation x, on calcule les probabilités avec l'équation 2.8 :

$$\hat{p}_1^{VC}(x),\ \hat{p}_2^{VC}(x), ..., \hat{p}_G^{VC}(x).$$

Par la suite, on affecte la nouvelle courbe x à la classe g dont la probabilité a posteriori estimée est la plus élevée.

Nous verrons dans les prochains chapitres l'application de la méthode de classification non-paramétrique que nous venons de voir à des données fonctionnelles simulées (chapitre 3) et réelles (chapitre 4).

CHAPITRE III

ÉTUDE DE SIMULATIONS

Nous avons présenté une méthode de classification non-paramétrique développée par Ferraty et Vieu (2006) au chapitre 2. L'objectif du présent chapitre est d'évaluer la performance de cette méthode de classification. Nous allons explorer l'effet des différents paramètres de la méthode de classification sur sa performance à l'aide de données simulées. Notons qu'afin de s'impliquer la présentation, nous considérons que des problèmes de classification binaire (G=2).

3.1 Simulation de données fonctionnelles

Dans cette section, nous allons présenter tout d'abord comment simuler une réalisation d'une fonction aléatoire $\mathcal{X}$. Si on veut simuler une réalisation d'une fonction d'aléatoire de moyenne $\mu(t) = \sum_{j=1}^{p} c_j \phi_j(t)$ et de fonction de covariance $v(s,t) = \sum_{j=1}^{p} a_j \phi_j(s)\phi_j(t)$, où les a_j et ϕ_j sont les valeurs et fonctions propres de l'opérateur de covariance défini par $v(s,t)$, les c_j représentent les coefficients de la fonction moyenne, on peut utiliser la formule suivante :

$$x(t) = \sum_{j=1}^{p} (a_j z_j + c_j)\phi_j(t),$$

où $z_1, ..., z_p$ sont des réalisations de variables aléatoires indépendantes et identiquement distribuées à une loi normale centrée réduite.

3.2 Comparaison des paramètres d'intérêt pour différents scénarios

Nous avons réalisé une étude de simulation afin d'évaluer la performance de la méthode par rapport à deux paramètres : la semi-métrique et le noyau. Nous avons considéré trois types de noyaux : uniforme, triangle et quadratique. Pour chaque noyau, on a utilisé les semi-métriques basées sur les MMCP de dimension 4 à 8 (noté *mcp4* à *mcp8*), les semi-métriques basées sur l'ACP de dimension 4 à 8 (noté *acp4* à *acp8* et les semi-métriques basées sur les dérivées d'ordre 0 à 3 (noté *deriv0* à *deriv3*) pour un total de $14 \times 3 = 42$ combinaisons de paramètres possibles.

Nous avons de plus considéré trois scénarios pour la structure globale des populations de courbes. Dans le premier scénario, les fonctions moyennes des deux populations sont différentes, mais leur fonction de covariance est la même. Pour le deuxième scénario, la fonction moyenne est la même pour les deux populations mais la fonction de covariance est différente. Le troisième scénario consiste en la situation où la fonction moyenne et la fonction de covariance sont différentes pour les deux populations.

Pour chacun des scénarios, on s'intéresse à 3 niveaux de difficulté pour la classification : facile, moyen et difficile. Pour le scénario 1, les 3 niveaux sont définis en fonction de la moyenne ; les fonctions moyennes sont très différentes dans le cas facile, peu différentes dans le cas moyen et très similaires pour le cas difficile. Pour le scénario 2, les 3 niveaux sont définis en fonction de la covariance ; les fonctions de covariance sont très différentes pour le cas facile, peu différentes pour le cas moyen et très similaires pour le cas difficile. Pour le scénario 3, les 3 niveaux sont déterminés en fonction des moyennes et des covariances ; les fonctions moyennes et les fonctions de covariance sont très différentes pour le cas facile, peu différentes pour le cas moyen et très similaires pour le cas difficile. Notons qu'on a

choisi ces trois scénarios et ces trois niveaux de difficulté dans le but de voir quelle combinaison de semi-métrique et de noyau donne le meilleur résultat pour chaque scénario et chaque niveau de difficulté.

On utilise le taux d'erreur de classification afin de mesurer la performance de la méthode de classification pour chacune des 42 combinaisons de paramètres considérées. Afin de calculer de tels taux, pour chaque scénario et chaque niveau de difficulté, nous avons simulé 50 échantillons de $N = 400$ courbes définies sur l'intervalle $[0, 1]$ et évaluées en 100 points équidistants $t_1, ..., t_{100}$, où $n_0 = 200$ courbes proviennent de la population Π_0 et $n_1 = 200$ courbes de la population Π_1. Chaque échantillon a ensuite été séparé de façon aléatoire en un échantillon d'entraînement de 200 courbes et un échantillon de test de 200 courbes. L'échantillon d'entraînement est utilisé afin d'ajuster le modèle, i.e. afin d'obtenir le classifieur et celui de test afin de calculer le taux d'erreur de classification. Finalement, le taux d'erreur de classification pour une certaine semi-métrique et pour un certain noyau est obtenu en prenant la moyenne des 50 erreurs calculées avec l'échantillon de test. Notons que nous avons utilisé la fonction R *funopadi.knn.lcv*, développée par Ferraty et Vieu (2006), afin d'obtenir les taux d'erreur de classification qu'on vient tout juste de présenter. Pour les 3 scénarios, nous avons remarqué que le type de fonction noyau n'a pas beaucoup d'effet sur la performance de la méthode de classification. Ainsi, les résultats ont été présentés avec la fonction noyau triangle. Les résultats pour les deux autres types de fonction noyau (uniforme et quadratique) sont présentés dans l'annexe. De plus, la fonction R *funopadi.knn.lcv* exige qu'on spécifie un nombre de noeuds pour le calcul de la semi-métrique basées sur les dérivées ; nous en avons utilisé 10.

Nous allons maintenant présenter en détail les trois scénarios ainsi que les niveaux de difficulté qui leur sont associés. Dans le scénario 1, les courbes des populations Π_0 et Π_1 ont été simulées avec une covariance commune $v(s, t)$ et des moyennes

respectives $\mu_0(t)$ et $\mu_1(t)$:

$$v(s,t) = \sum_{j=1}^{7} j^{-2.5} \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$\mu_0(t) = \sum_{j=1}^{7} c_{j0} \cdot 0.9 \cdot \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$\mu_1(t) = \sum_{j=1}^{7} c_{j1} \cdot 0.9 \cdot \sin\Big(1.2\pi(j-0.2)t\Big).$$

Dans le cas facile, les coefficients c_{1j} à $c_{7j}, j \in \{0,1\}$ sont :

$$(c_{10}, c_{20}, c_{30}, c_{40}, c_{50}, c_{60}, c_{70}) = (0.65, -0.75, 0.7, -0.85, 0.45, 0, 0.55),$$
$$(c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}) = (0.1, -0.5, 0.5, -0.75, 0.4, 0.1, 0).$$

Les coefficients des fonctions moyennes dans le cas moyen sont :

$$(c_{10}, c_{20}, c_{30}, c_{40}, c_{50}, c_{60}, c_{70}), = (0.65, -0.5, 0.7, -0.65, 0.45, 0, 0.35),$$
$$(c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}) = (0.1, -0.5, 0.5, -0.75, 0.4, 0.1, 0).$$

Finalement, dans le cas difficile, on a :

$$(c_{10}, c_{20}, c_{30}, c_{40}, c_{50}, c_{60}, c_{70}) = (0.1, -0.5, 0.4, -0.75, 0.6, 0, 0.1),$$
$$(c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}) = (0.1, -0.5, 0.5, -0.75, 0.5, 0.1, 0).$$

Les figures 3.1, 3.2 et 3.3 présentent respectivement le cas facile, moyen et difficile. Pour les 3 figures, un échantillon de courbes est illustré en (a). Les courbes en rouge proviennent de la population Π_0 et celles en bleu de la population Π_1. Les fonctions moyennes $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu) sont représentées en (b). Les courbes des deux populations ont été générées avec la fonction de covariance $v(s,t)$ qui est illustrée en (c). On constate que les courbes varient davantage autour de leur moyenne dans l'intervalle $[0.4, 0.8]$. Le graphique en (d) présente les erreurs de classification pour les différents types de semi-métriques considérées.

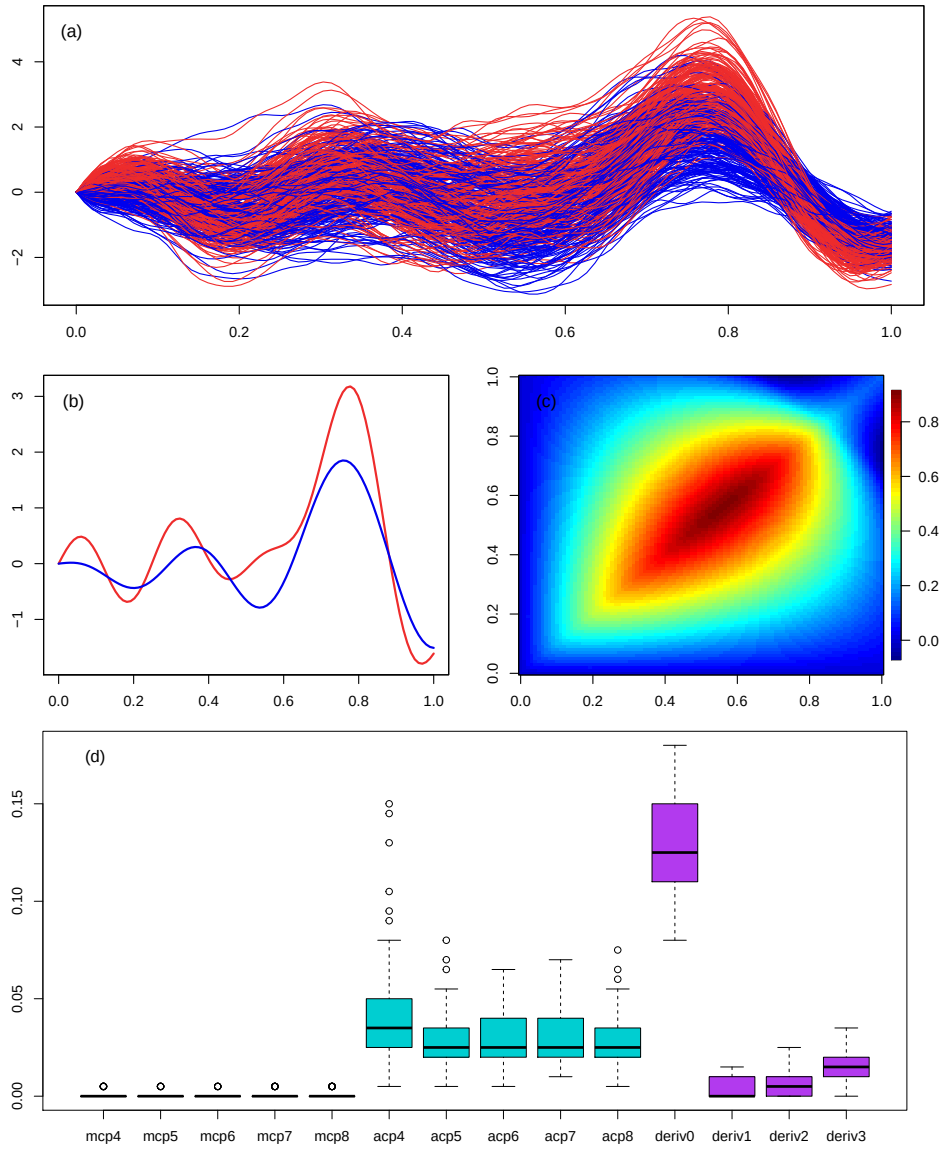

Figure 3.1: Scénario 1 : cas facile. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). La figure (b) illustre les fonctions moyennes $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu) et la figure (c) représente la fonction de covariance commune aux populations Π_0 et Π_1. La figure (d) présente les erreurs de classification pour les différentes semi-métriques.

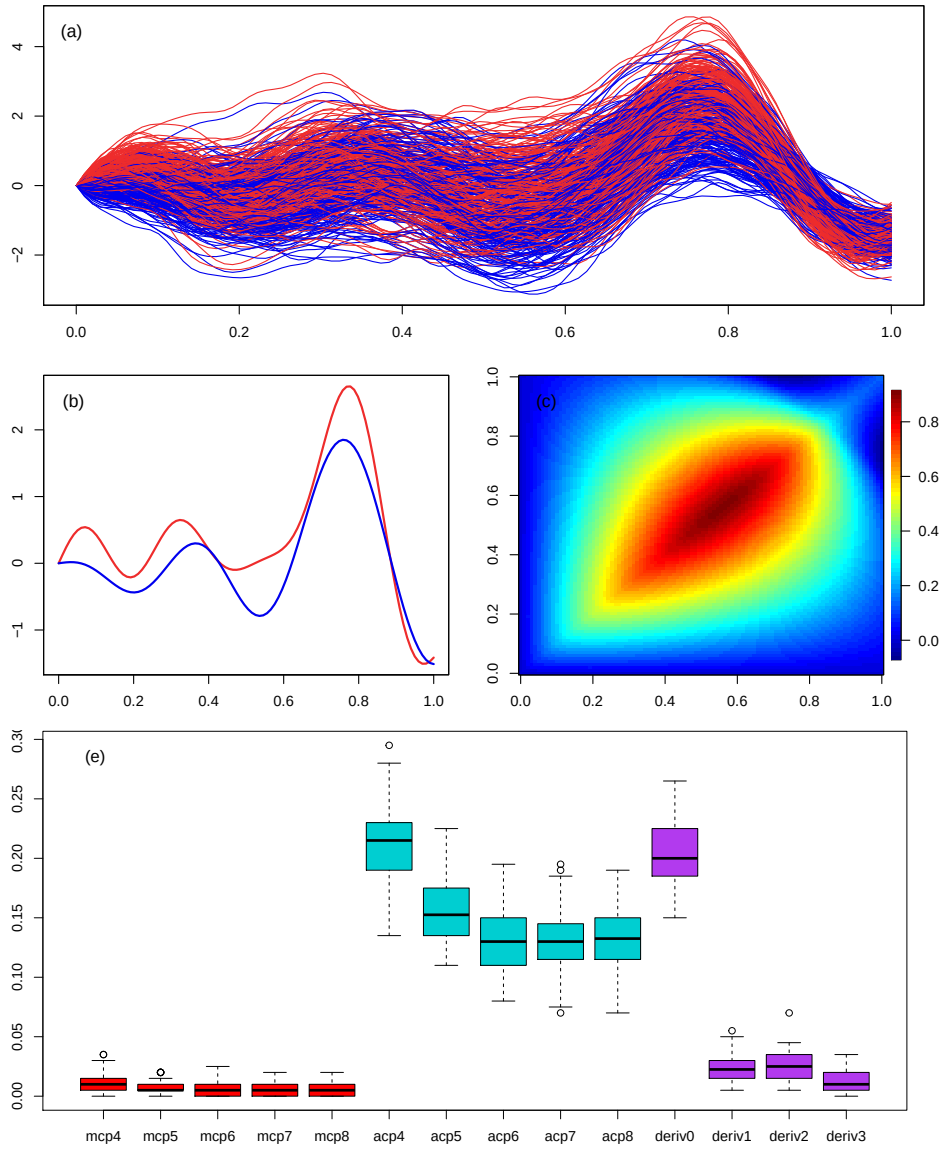

Figure 3.2: Scénario 1 : cas moyen. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). La figure (b) illustre les fonctions moyennes $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu) et la figure (c) représente la fonction de covariance commune aux populations Π_0 et Π_1. La figure (d) présente les erreurs de classification pour les différentes semi-métriques.

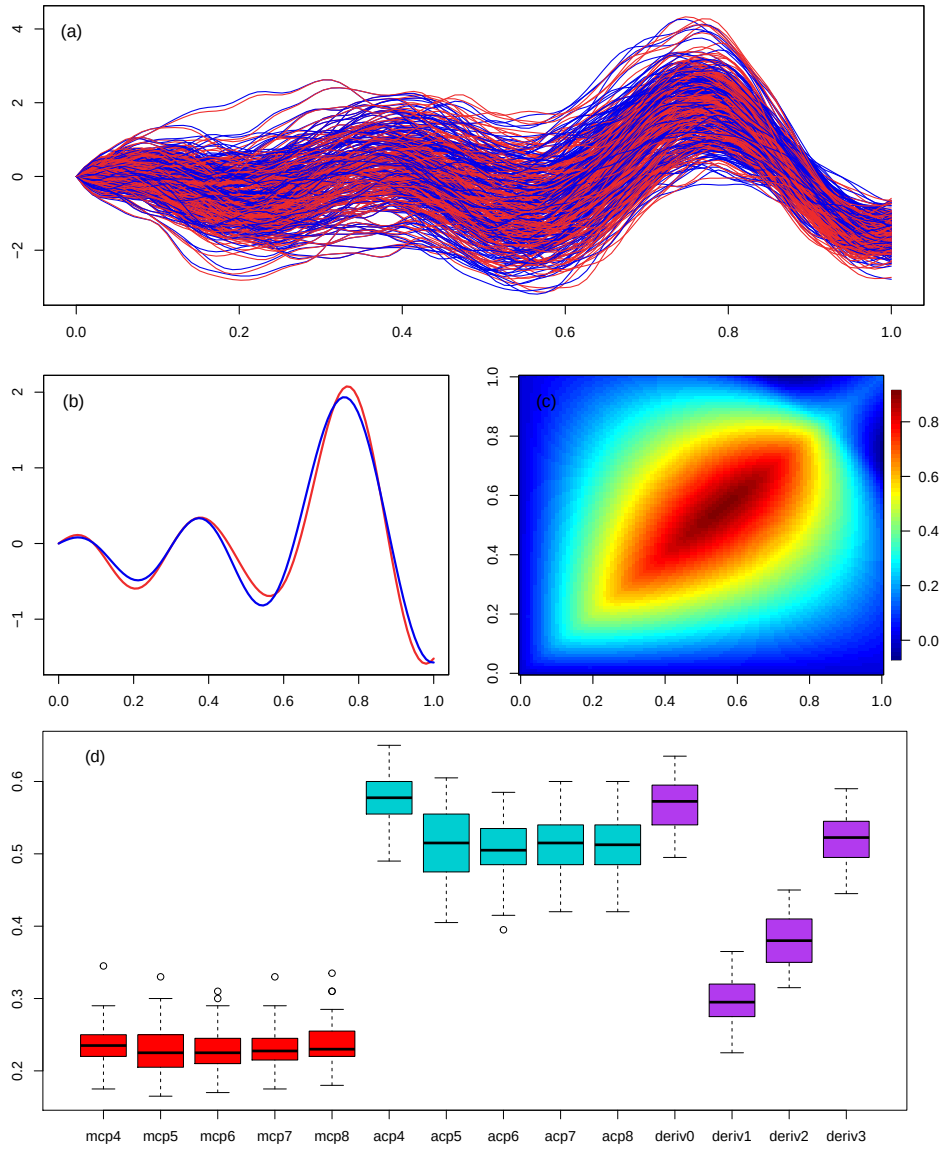

Figure 3.3: Scénario 1 : cas difficile. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). La figure (b) illustre les fonctions moyennes $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu) et la figure (c) représente la fonction de covariance commune aux populations Π_0 et Π_1. La figure (d) présente les erreurs de classification pour les différentes semi-métriques.

Nous allons maintenant présenter le deuxième scénario. Les données fonctionnelles ont été simulées avec la fonction moyenne :

$$\mu(t) = \sum_{j=1}^{7} c_j \cdot 0.9 \cdot \sin\Big(1.2\pi(j-0.2)t\Big),$$

où pour les 3 niveaux de difficulté, les coefficients sont :

$$(c_1, c_2, c_3, c_4, c_5, c_6, c_7) = (0.65, -0.5, 0.7, -0.65, 0.45, 0, 0.35).$$

Pour le niveau facile, les courbes de la population Π_0 et Π_1 ont été générées respectivement avec les covariances :

$$v_0(s,t) = \sum_{j=1}^{7} 0.9^2 j\big[\exp(j-0.9)\big]^{-1} \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$v_1(s,t) = \sum_{j=1}^{7} 0.9^2 \big[\exp(j-0.9)\big]^{-2} \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big).$$

Les valeurs propres associées à v_0 varient de 0.9 à 6.9×10^{-12} et celles associées à v_1 varient de 0.82 et 5.3×10^{-26}. Pour le niveau moyen, les fonctions de covariance sont :

$$v_0(s,t) = \sum_{j=1}^{7} j/\exp(j-0.5) \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$v_1(s,t) = \sum_{j=1}^{7} 1.2 \cdot \exp(j+0.3)^{-1} \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big).$$

Les valeurs propres associées à v_0 varient de 0.61 à 4.63×10^{-12} et les valeurs propres associées à v_1 varient de 0.33 à 8.32×10^{-14}. Pour le niveau difficile, les courbes ont été simulées avec les fonctions de covariance suivantes :

$$v_0(s,t) = \sum_{j=1}^{7} j/\exp(j-0.2) \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$v_1(s,t) = \sum_{j=1}^{7} 2(j+0.5)/\exp(j-0.2) \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big).$$

Les valeurs propres associées à v_0 varient de 0.45 à 3.43×10^{-12} et celles associées à v_1 varient de 1.35 à 6.97×10^{-12}.

Les figures 3.4, 3.5 et 3.6 présentent respectivement le cas facile, moyen et difficile. Pour les 3 figures, un échantillon de courbes est illustré en (a). Les courbes en rouge proviennent de la population Π_0 et celles en bleu de la population Π_1. La fonction moyenne $\mu(t)$ est représentée en (b). Les courbes de la population Π_0 ont été générées avec la fonction de covariance $v_0(s,t)$ qui est illustrée en (c). On peut remarquer que les courbes varient plus autour de leur moyenne dans l'invervalle $[0.6, 0.8]$. Les courbes de la population Π_1 ont été générées avec la fonction de covariance $v_1(s,t)$ qui est illustrée en (d). On peut constater que les courbes varient davantage autour de leur moyenne dans l'intervalle $[0.4, 0.8]$ pour les niveaux facile et moyen, alors que pour le niveaux difficile, les courbes varient plus dans l'intervalle $[0.6, 0.8]$. Le graphique en (e) présente les erreurs de classification pour les différents types de semi-métriques considérées.

Nous allons maintenant présenter le scénario 3 où les fonctions de covariance et les fonctions moyennes sont différentes pour les populations Π_0 et Π_1. Les données fonctionelles sont générées à partir des moyennes suivantes :

$$\mu_0(t) = \sum_{j=1}^{7} c_{j0} \cdot 0.9 \cdot \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$\mu_1(t) = \sum_{j=1}^{7} c_{j1} \cdot 0.9 \cdot \sin\Big(1.2\pi(j-0.2)t\Big).$$

Le cas facile se distingue des autres cas par les coefficients et les fonctions de covariance suivants :

$$(c_{10}, c_{20}, c_{30}, c_{40}, c_{50}, c_{60}, c_{70}) = (0.65, -0.75, 0.7, -0.85, 0.45, 0, 0.55),$$

$$(c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}) = (0.1, -0.5, 0.5, -0.75, 0.4, 0.1, 0),$$

$$v_0(s,t) = \sum_{j=1}^{7} j/\exp(j-0.2) \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

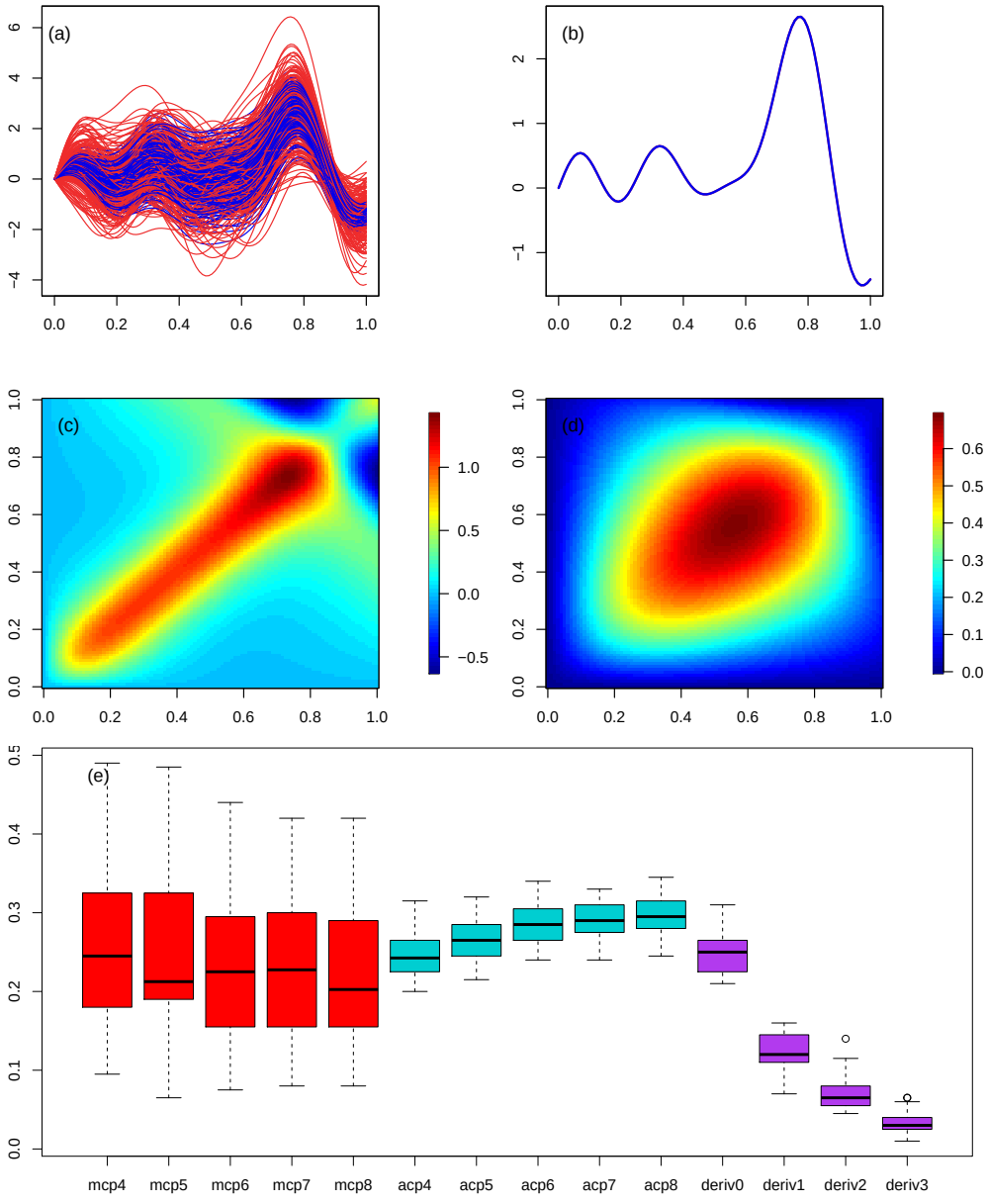

Figure 3.4: Scénario 2 : cas facile. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). La figure (b) illustre la fonction moyenne $\mu(t)$. Les figures (c) et (d) présentent la fonction de covariance $v_0(s,t)$ et $v_1(s,t)$ respectivement. La figure (e) présente les erreurs de classification pour les différentes semi-métriques.

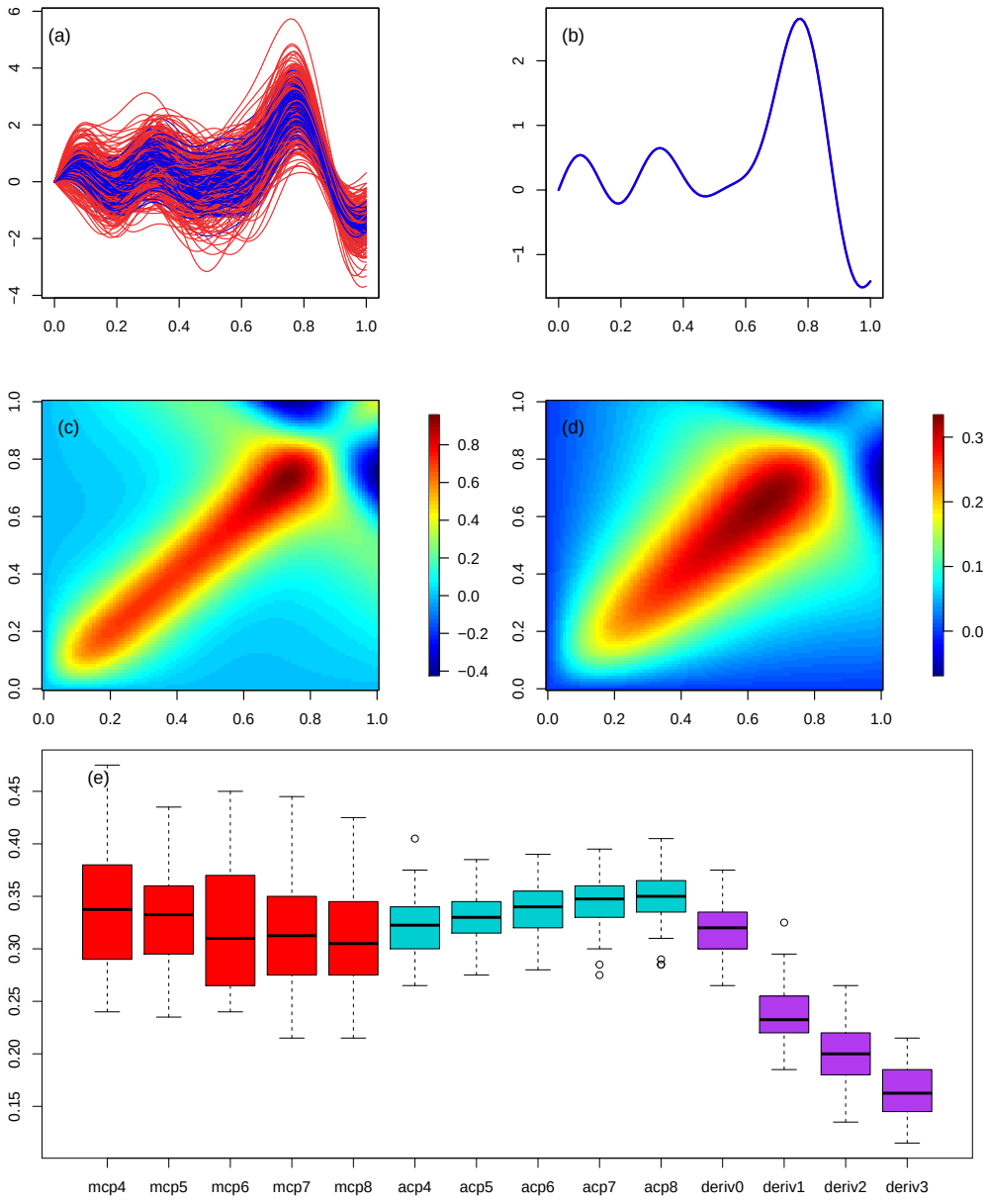

Figure 3.5: Scénario 2 : cas moyen. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). La figure (b) illustre la fonction moyenne $\mu(t)$. Les figures (c) et (d) présentent la fonction de covariance $v_0(s,t)$ et $v_1(s,t)$ respectivement. La figure (e) présente les erreurs de classification pour les différentes semi-métriques.

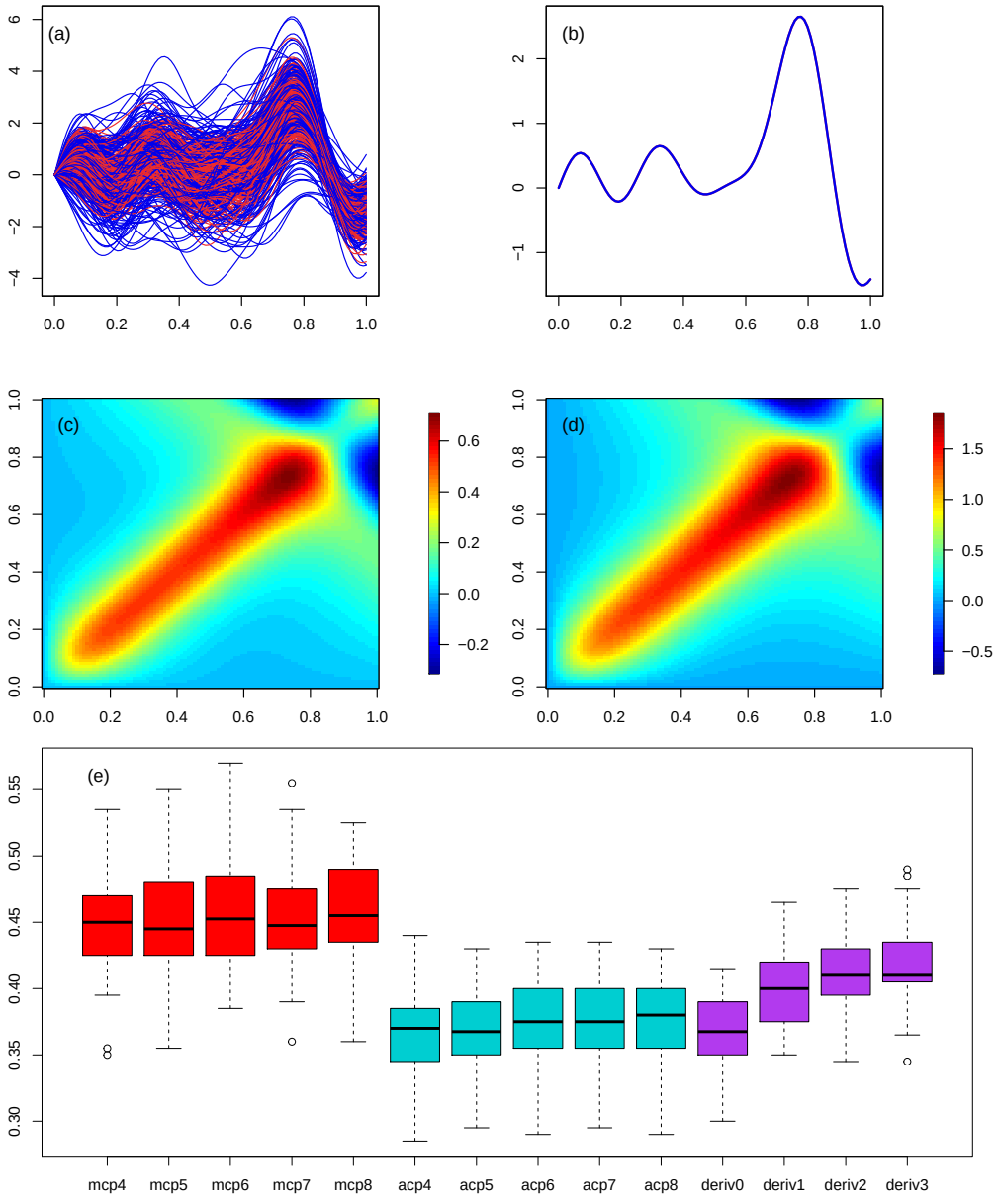

Figure 3.6: Scénario 2 : cas difficile. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). La figure (b) illustre la fonction moyenne $\mu(t)$. Les figures (c) et (d) présentent la fonction de covariance $v_0(s,t)$ et $v_1(s,t)$ respectivement. La figure (e) présente les erreurs de classification pour les différentes semi-métriques.

$$v_1(s,t) = \sum_{j=1}^{7} 1.2\Big(\exp(j-0.9)\Big)^{-2} \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big).$$

Les coefficients de la fonction moyenne et les fonctions de covariance pour le cas moyen sont :

$$(c_{10}, c_{20}, c_{30}, c_{40}, c_{50}, c_{60}, c_{70}) = (0.65, -0.5, 0.7, -0.65, 0.45, 0, 0.35),$$
$$(c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}) = (0.1, -0.5, 0.5, -0.75, 0.4, 0.1, 0),$$

$$v_0(s,t) = \sum_{j=1}^{7} j/\exp(j-0.5) \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$v_1(s,t) = \sum_{j=1}^{7} 1.2\Big(\exp(j+0.3)\Big)^{-1} \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big).$$

Pour le cas difficile, les coefficients de la fonction moyenne et les fonctions de covariance sont :

$$(c_{10}, c_{20}, c_{30}, c_{40}, c_{50}, c_{60}, c_{70}) = (0.1, -0.5, 0.4, -0.75, 0.6, 0, 0.1),$$
$$(c_{11}, c_{21}, c_{31}, c_{41}, c_{51}, c_{61}, c_{71}) = (0.1, -0.5, 0.5, -0.75, 0.5, 0.1, 0),$$

$$v_0(s,t) = \sum_{j=1}^{7} j/\exp(j-0.2) \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big),$$

$$v_1(s,t) = \sum_{j=1}^{7} 2(j+0.5)/\exp(\text{j-0.2}) \cdot 0.9^2 \cdot \sin\Big(1.2\pi(j-0.2)s\Big) \sin\Big(1.2\pi(j-0.2)t\Big).$$

Les figures 3.7, 3.8 et 3.9 présentent respectivement le cas facile, moyen et difficile. Pour les 3 figures, un échantillon de courbes est illustré en (a). Les courbes en rouge proviennent de la population Π_0 et celles en bleu de la population Π_1. Les fonctions moyennes $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu) sont représentées en (b). Les courbes de la population Π_0 ont été générées avec la fonction de covariance $v_0(s,t)$ qui est illustrée en (c). On peut constater que les courbes varient davantage autour de leur moyenne dans l'invervalle $[0.6, 0.8]$. Les courbes de la population Π_1 ont été générées avec la fonction de covariance $v_1(s,t)$ qui est illustrée en (d).

On remarque que pour les niveaux facile, moyen et difficile, les courbes varient davantage autour de leur moyenne dans l'intervalle $[0.4, 0.7], [0.4, 0.8]$ et $[0.5, 0.8]$ respectivement. Le graphique en (e) présente les erreurs de classification pour les différents types de semi-métriques considérées.

3.3 Discussion

Selon les résultats présentés aux figures 3.1(e) à 3.9(e), il est évident que la semi-métrique basée sur l'analyse en composante principale rend la méthode de classification moins performante puisque pour les scénarios 1 et 3, les taux de mauvaise classification basées sur l'ACP (*acp4* à *acp8*) sont beaucoup plus grands que ceux basées sur la MMCP et les dérivées. Pour le niveau difficile du scénario 2, on peut voir que c'est le seul cas où les taux de mauvaise classification des semi-métriques basées sur l'ACP et la distance euclidienne sont plus petits que ceux des autres semi-métriques. On peut remarquer que le taux d'erreur de classification de la semi-métrique *acp4* est plus grand que ceux des semi-métriques *acp5* à *acp8* pour les scénarios 1 et 3. Cela peut s'expliquer par le fait que la projection en dimension 4 n'est pas assez grande pour obtenir une bonne représentation des données. Par contre, lorsque les fonctions moyennes des deux populations sont identiques comme dans le scénario 2, la projection en dimension 4 semble être suffisante.

On constate que la performance de la méthode de classification basée sur les dérivées d'ordre 1, 2 et 3 surpasse celle des autres types de semi-métriques dans les cas facile et difficile du scénario 2. On remarque aussi que dans le cas difficile du scénario 1, les erreurs de classification de *deriv1*, *deriv2* et *deriv3* varient entre 0.29 et 0.55 lorsque les courbes ont une apparence rugueuse. Dans le cas où les courbes sont lisses, pour le niveau difficile du scénario 2 par exemple, les taux de mauvaise classification varient entre 0.375 et 0.44. Notons que dans le cas difficile du scénario

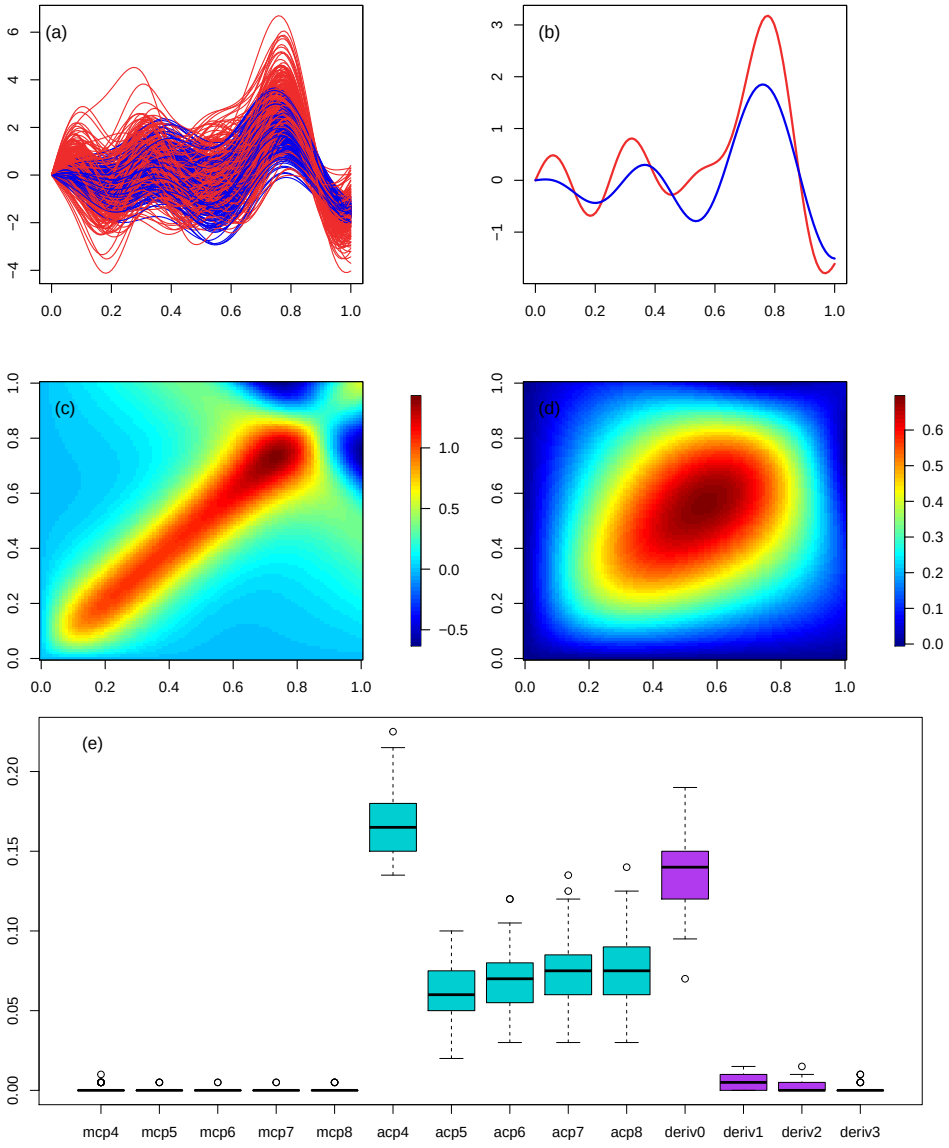

Figure 3.7: Scénario 3 : cas facile. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). Les figures (b) illustre la fonction moyenne $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu). Les figures (c) et (d) présentent la fonction de covariance $v_0(s,t)$ et $v_1(s,t)$ respectivement. La figure (e) présente les différentes erreurs de classification pour les différentes semi-métriques.

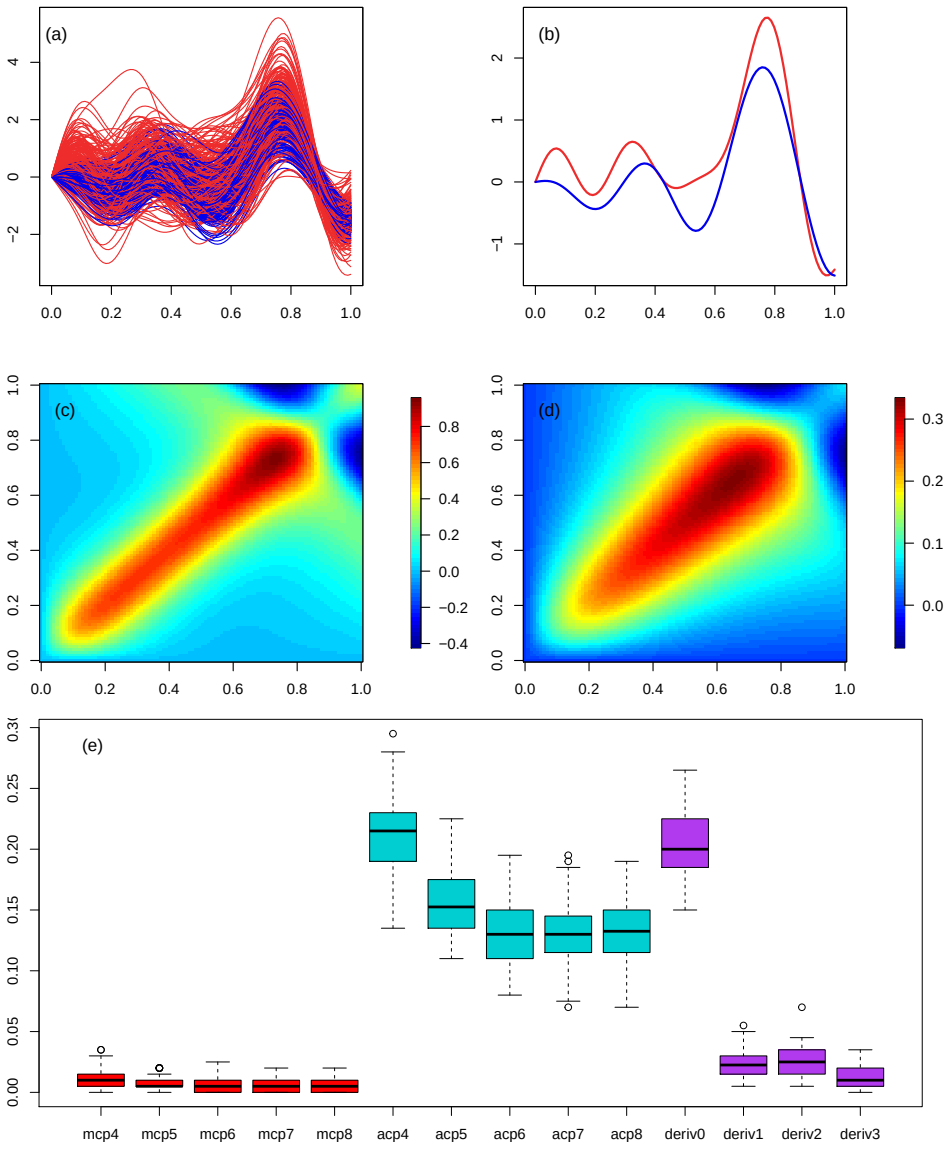

Figure 3.8: Scénario 3 : cas moyen. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). Les figures (b) illustre la fonction moyenne $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu). Les figures (c) et (d) présentent la fonction de covariance $v_0(s,t)$ et $v_1(s,t)$ respectivement. La figure (e) présente les différentes erreurs de classification pour les différentes semi-métriques.

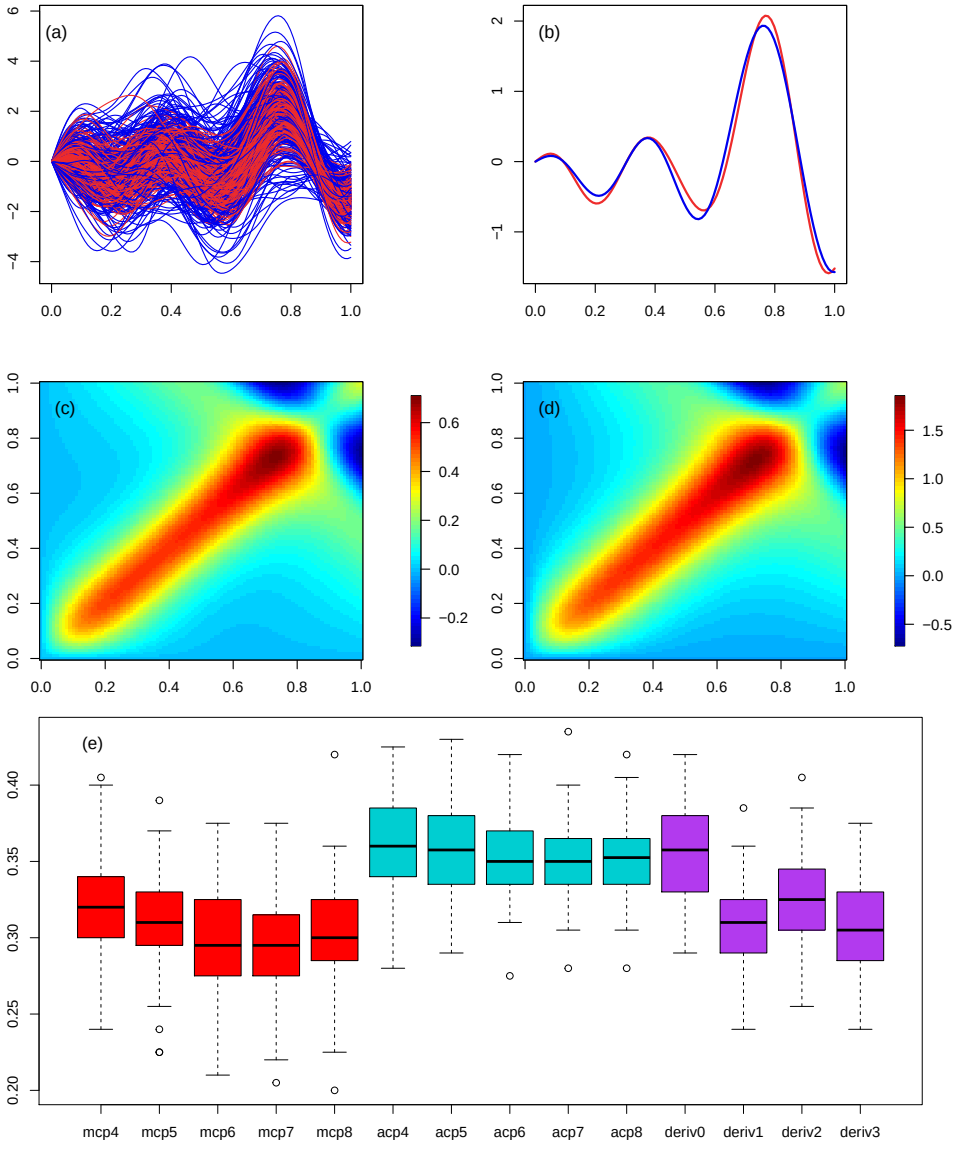

Figure 3.9: Scénario 3 : cas difficile. La figure (a) illustre un échantillon de 200 courbes provenant de la population Π_0 (rouge) et de la population Π_1 (bleu). Les figures (b) illustre la fonction moyenne $\mu_0(t)$ (rouge) et $\mu_1(t)$ (bleu). Les figures (c) et (d) présentent la fonction de covariance $v_0(s,t)$ et $v_1(s,t)$ respectivement. La figure (e) présente les différentes erreurs de classification pour les différentes semi-métriques.

3, les taux de mauvaise classification varient entre 0.29 et 0.345. Ces résultats semblent nous montrer que les semi-métriques *dériv* ont tendance à améliorer la performance de la méthode de classification non-paramétrique lorsque les courbes sont lisses. Il est à noter que lorsque les moyennes sont identiques comme dans le scénario 2, les taux d'erreur basées sur la MMCP (*mcp4* à *mcp8*) tendent à augmenter lorsque les fonctions de covariance sont de plus en plus similaires alors que les taux d'erreur basées sur les dérivées ont tendance à augmenter un peu moins.

Généralement, on obtient une meilleure classification avec les semi-métriques basées sur la MMCP, qu'avec les deux autres types de semi-métriques puisque les taux de mauvaise classification basées sur la MMCP sont les plus petits pour tous les niveaux du scénario 1, de même que pour les niveaux facile et moyen du scénario 3. On peut également voir dans le cas difficile du scénario 1 que les courbes rouges et bleues sont quasiment homogènes et les erreurs de classification des semi-métriques basées sur la MMCP (*mcp4* à *mcp8*) sont autour de 0.25, ce qui semble montrer que la méthode de classification avec les semi-métriques basées sur la MMCP est la plus performante. En comparant les taux d'erreur de classification des scénarios 1 et 3 au scénario 2, on constate que les semi-métriques basées sur la MMCP rendent la méthode de classification moins performante que les semi-métriques basées sur les dérivées lorsque les fonctions moyennes des deux populations sont les mêmes tandis que lorsque la moyenne est différente pour chacune des populations comme dans le cas des scénarios 1 et 3, on peut voir que les semi-métriques basées sur la MMCP sont les plus performantes. Il semble donc que le fait que les fonctions moyennes des deux populations soient identiques ou différentes a un effet sur la performance de la méthode de classification avec la semi-métrique basée sur les moindres carrés partiels.

En résumé, lorsqu'on veut classer des courbes provenant de deux populations ayant

des moyennes différentes, la semi-métrique basée sur la MMCP semble donner les meilleures résultats. Par contre, lorsque les populations ont des moyennes égales (ou très similaires) et que la différence est au niveau de leur fonction de covariance, alors la semi-métrique basée sur les dérivées semble donner de meilleurs résultats.

Nous avons fait cette étude de simulations afin de déterminer les paramètres optimaux de la méthode de classification par noyau dans différents contextes pour ensuite l'appliquer à de vraies données.

CHAPITRE IV

ASSOCIATION FONCTIONNELLE ET APPLICATION

Dans ce chapitre, nous allons présenter l'utilisation du taux de classification de la méthode par noyau à des fins d'association, puis l'appliquer à des données génétiques. L'analyse que nous allons faire va donc être similaire à une analyse du génome faite avec de multiples test du χ^2, sauf que nous allons utiliser l'erreur de classification comme une mesure d'association.

4.1 Quelques notions de base en génétique

Dans cette section, nous allons introduire plusieurs termes en génétique que nous allons utiliser dans la prochaine section.

4.1.1 Génome, chromosome, gène et allèle

L'information génétique (ADN) dont une cellule hérite est le génome. Lorsqu'une cellule se reproduit, ses molécules d'ADN sont répliquées et transmises à la prochaine génération. Le génome est l'ensemble complet de l'ADN d'un organisme. Les brins d'ADN sont à la fois importants et délicats, il est donc essentiel qu'ils soient soigneusement enroulés et protégés pendant le processus de division cellulaire. Pour les renforcer et les garder en sécurité, l'ADN est bouclé et enroulé dans

une structure appelée chromosome. Les molécules d'ADN sont composées d'acides nucléiques. Les nucléotides de l'ADN contiennent quatre bases azotées : adénine (A), thymine (T), guanine (G) et cytosine (C), représentées comme A, T, G et C lors de la description d'une séquence d'ADN. Un chromosome est constitué de deux brins antiparallèles enroulés l'un autour de l'autre pour former une double hélice sur laquelle sont codés des milliers de gènes. La région précise du chromosome où se trouve un gène particulier est appelée locus. Il existe plusieurs formes possibles d'un même gène et ces formes sont appelées allèles. Presque toutes les cellules vivantes contiennent des chromosomes. Par exemple chez l'homme, chaque cellule somatique contient 23 paires de chromosomes tandis que le génome d'une drosophile contient 4 paires de chromosomes. La structure détaillée d'un chromosome humain est illustrée à la figure 4.1. Pour résumé, on peut considérer le code ADN comme un ensemble d'instructions soigneusement organisées en paragraphes (gènes) et en chapitres (chromosomes), alors l'ensemble du manuel du début à la fin serait le génome.

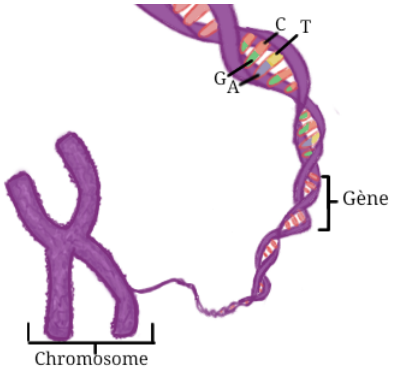

Figure 4.1: Description de la structure d'un chromosome humain.

4.1.2 Phénotype et génotype

L'apparence ou un caractère observable d'un organisme est appelée phénotype, et la constitution allélique d'un individu est appelée génotype. Des exemples de phénotypes observés sont le groupe sanguin, la couleur des yeux, la texture des cheveux, mais aussi une maladie génétique (la leucémie par exemple). On peut considérer un exemple de génotype et de phénotype d'un rat à la figure 4.2. Dans cet exemple, l'allèle de la couleur des rats est soit R (brune) ou r (blanche) comme vu à la figure 4.2(a) où l'on suppose que l'allèle brun est dominant (R) et l'allèle blanc est récessif (r), l'un hérité de la mère et l'autre hérité du père. En génétique, la dominance est le phénomène d'une variante (allèle) d'un gène sur un chromosome masquant l'effet d'un autre allèle (recessif) différent du même gène sur l'autre copie du chromosome. Si le descendant hérite deux allèles différents, on dira qu'il est hétérozygote et si le descendant hérite deux allèles identiques, on dira que le descendant est homozygote. Dans cet exemple, on a considéré que le père et la mère sont hétérozygotes (Rr). On peut voir à la figure 4.2(b) que lorsque le rat (descendant) est homozygote (Rr) ou hétérozygote (RR), il aura la couleur brune, alors que lorsque le rat (descendant) hérite deux allèles récessifs (rr), il aurait la couleur blanche. Le trait récessif peut être exprimé lorsque le descendant reçoit deux copies de l'allèle récessif. Une distinction entre le phénotype et le génotype est que ce dernier est hérité des parents d'un organisme alors que le phénotype ne l'est pas.

4.1.3 Variabilité génétique et marqueurs génétique

Les variations génétiques sont des différences dans la séquence d'ADN d'un organisme à l'autre et celles-ci comprennent des mutations et des polymorphismes. Une mutation est une altération permanente de la séquence nucléotique de l'ADN d'un

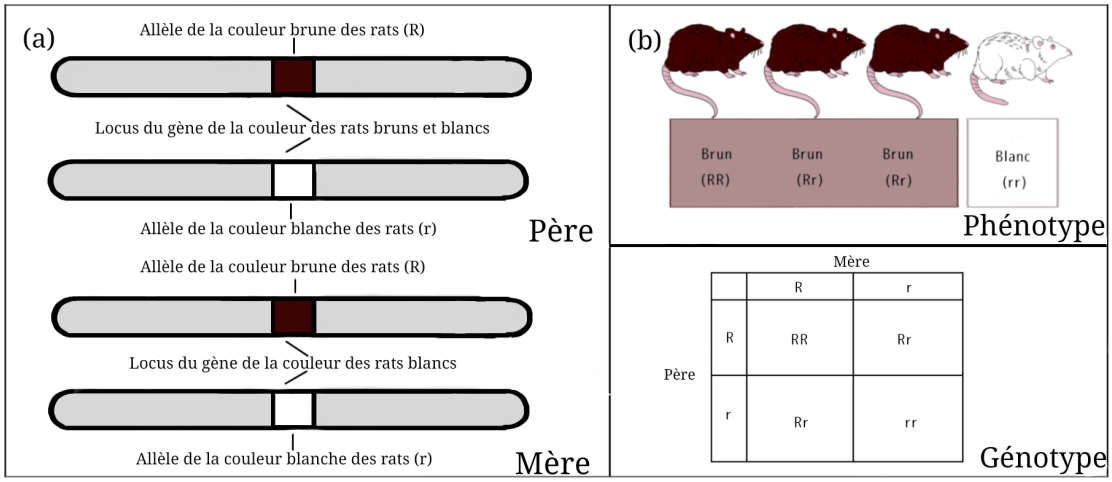

Figure 4.2: Exemple simple du génotype et du phénotype chez un rat. La figure (a) présente la composition allélique du père hétérozygote (Rr) et de la mère hétérozygote (Rr). La figure (b) détaille comment un rat peut hériter la couleur brune ou blanche en fonction des allèles portés par ses parents.

organisme, alors que le polymorphisme est défini comme des variations dans la séquence d'ADN qui coexistent dans une population à des fréquences relativement élevées (> 1 %). La santé, l'apparence, le comportement et d'autres caractéristiques d'une personne dépendent en partie de ces polymorphismes. Environ 90 % de la variation du génome humain se présente sous la forme de polymorphisme nucléotide simple (SNP en version abrégée de l'anglais). Un SNP se réfère à une altération d'une seule paire de bases qui est courante dans la population.

Un marqueur génétique est un gène ou une séquence polymorphe d'ADN à un emplacement connu sur un chromosome. C'est-à-dire qu'un marqueur génétique peut être une séquence d'ADN courte, comme une séquence autour d'une seule paire de bases (SNP), ou de séquences répétées, comme les microsatellites. Les SNPs peuvent agir comme des marqueurs biologiques, aidant à localiser les gènes

associés à une maladie. Dans ce mémoire, lorsqu'on fait référence à un marqueur, nous allons nous référer à des SNPs. La figure 4.3 présente un exemple de SNP chez 2 individus.

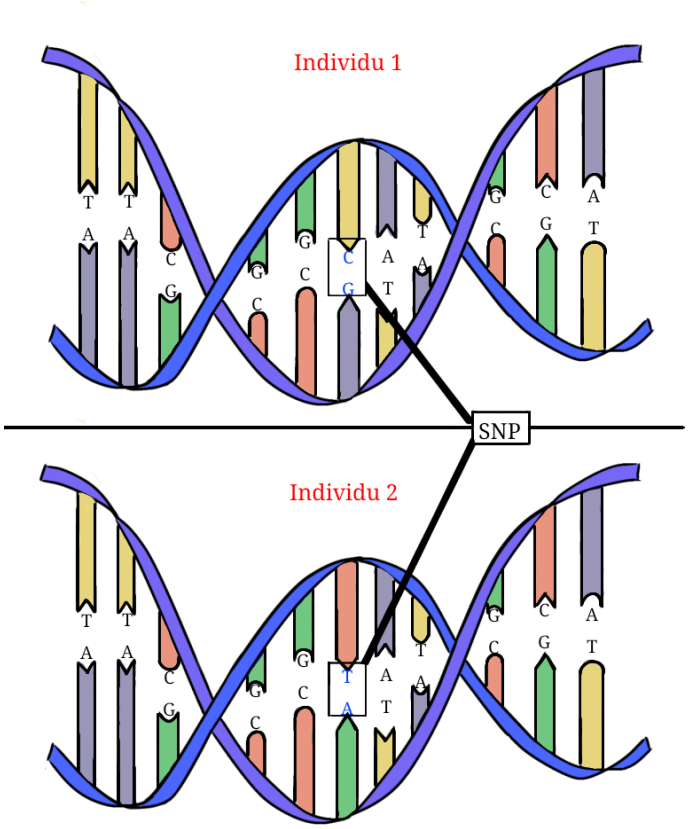

Figure 4.3: Exemple d'un SNP. Il s'agit de deux fragments d'ADN séquencés provenant de deux individus non reliés. La séquence d'ADN en haut diffère de la séquence d'ADN en bas à un seul emplacement nucléotidique.

4.2 Méthode d'association fonctionnelle

L'objectif dans ce chapitre est de décrire l'association entre les génotypes et les phénotypes, en prenant comme example et fil conducteur des données sur le gravi-

tropisme de l'arabette des dames (Moore *et al.* (2013)). L'arabette des dames est une petite plante à fleurs de la famille de la moutarde (Brassicacées). Des images numériques de racines de semis de 162 lignées consanguines recombinantes dérivées d'un croisement biparental de deux variétés (*îles du Cap-Vert* et *Landsberg erecta*) de l'arabette des dames ont été collectées automatiquement. Une lignée consanguine recombinante est formée en croisant deux souches consanguines suivies d'accouplements entre frères et soeurs pour créer une nouvelle lignée consanguine dont le génome est une mosaïque des génomes parentaux. Pour chacune des 162 lignées consanguines recombinantes, il y a 8 à 20 réplications de graines qui ont été germées puis pivotées de 90 degrés, pour changer l'orientation de la gravité. La croissance des semis a été capturée toutes les 2 minutes pendant 8 heures. Les auteurs s'intéressent à l'angle de la pointe de la racine au cours du temps. De plus, on considère des données de génotype à 234 marqueurs sur cinq chromosomes, pour lesquels nous nous intéressons à deux lignées, soit la lignée *Cvi* ou *Ler*. Dans notre cas, on considère ces espèces comme les classes de nos données fonctionnelles. Nous utilisons la moyenne des 8 à 20 réplications pour obtenir 162 courbes moyennes de lignées consanguines recombinantes et celles-ci représentent les phénotypes de l'angle de la pointe des racines.

Pour chacun des marqueurs, nous avons constaté que les moyennes des phénotypes pour les 2 classes (*Cvi* et *Ler*) sont très similaires et les covariances pour ces deux groupes sont différentes. Pour illustrer les données, nous allons présenter les fonctions moyennes et les fonctions de covariance pour chacun des groupes pour le marqueur 28 appartenant au chromosome 2, et pour le marqueur 160 sur le chromosome 5. Elles sont illustrées aux figures 4.4 et 4.5 respectivement. Pour les deux figures, les courbes de chacun des groupes sont illustrées en (a). La fonction moyenne pour le groupe *Ler* (rouge) et la fonction moyenne pour le groupe *Civ* (bleu) sont illustrées en (b). Les fonctions de covariance pour les groupes *Ler* et *Civ*

sont présentées aux graphiques (c) et (d) respectivement. Comme nous pouvons le constater, pour ces deux marqueurs, les moyennes sont très semblables. Par contre, les fonctions de covariance sont différérentes. Dans ce cas, la semi-métrique que nous allons utiliser sera la semi-métrique basée sur les dérivées tel que le suggère le chapitre 3. L'analyse que nous allons effectuer revient à décrire l'association entre

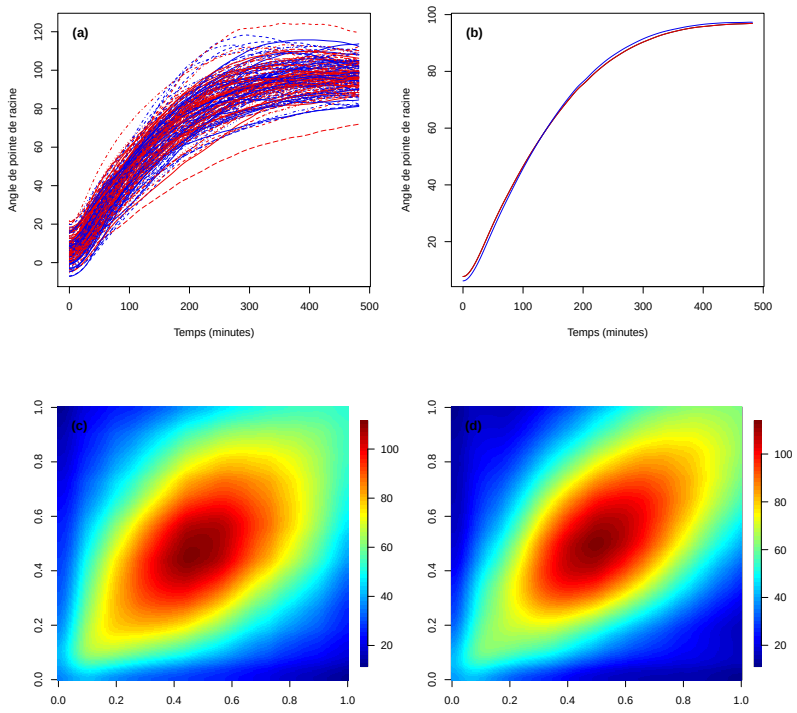

Figure 4.4: Marqueur 28 du chromosome 2. La figure (a) présente les 162 courbes selon le groupe *Ler* (rouge) et *Civ* (bleu). Le graphique (b) illustre les fonctions moyennes pour les classes *Ler* (rouge) et *Civ* (bleu), les figures (c) et (d) présentent les fonctions de covariance pour le groupe *Ler* et *Civ* respectivement.

le phénotype et le génotype pour chacun des 234 marqueurs. Pour un marqueur, on veut trouver s'il y a une association entre les phénotypes et les génotypes. Au lieu de considérer une valeur-p, nous allons considérer le taux d'erreur de

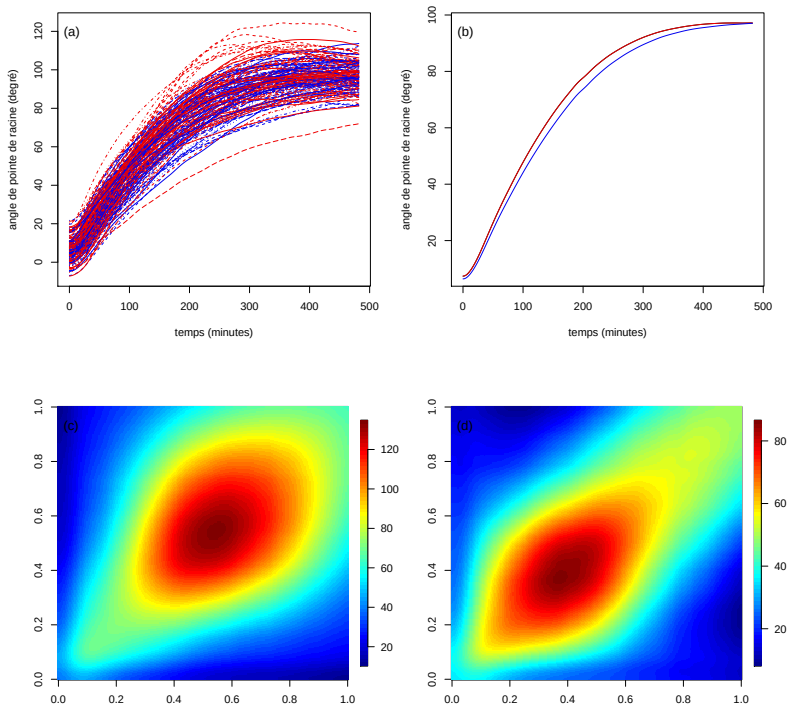

Figure 4.5: Marqueur 160 du chromosome 5. La figure (a) présente les 162 courbes selon le groupe *Ler* (rouge) et *Civ* (bleu). Le graphique (b) illustre les fonctions moyennes pour les classes *Ler* (rouge) et *Civ* (bleu), les figures (c) et (d) présentent les fonctions de covariance pour le groupe *Ler* et *Civ* respectivement.

classification comme étant une mesure d'association. Pour chaque marqueur, nous allons utiliser la méthode de classification non-paramétrique pour classer les 162 courbes (phénotypes) dans la bonne classe qui correspond au génotype *Cvi* ou *Ler*. Si le marqueur génétique est lié au phénotype alors l'erreur de classification attendue serait petite. Au contraire, si le marqueur génétique n'est pas lié au phenotype, alors l'erreur de classification attendue serait grande. Nous allons à obtenir 234 taux d'erreur de classification qui correspondent à 234 marqueurs répartis sur cinq chromosomes. S'il y a une forte association entre les phénotypes

et les marqueurs génétiques, on peut s'attendre à voir que les taux d'erreur de classification soient petits alentour d'une région précise tout au long des cinq chromosomes.

4.3 Résultats obtenus

Les données ont été prélablement lissées par les splines de lissage. En nettoyant les données, on a remarqué que le marqueur 231 sur le chromosome 5 présente beaucoup de données manquantes, alors ce marqueur a été omis de nos analyses. Donc, nous allons obtenir 233 taux d'erreurs de classification au lieu de 234.

Nous prenons comme référence les résultats de Kwak *et al.* (2016) et nous allons les comparer avec nos résultats obtenus. Kwak *et al.* (2016) ont appliqué aux données génétiques de Moore *et al.* (2013) l'analyse quantitative multiple des locus de caractères quantitatifs (de l'anglais "multiple quantitative trait loci analysis"). Leur approche évalue la vraisemblance sur de nombreuses positions (marqueurs) le long du génome afin de calculer la statistique de test, qui est le score LOD. Dans leur cas, un score LOD élevé pour un marqueur nous indique que le marqueur est probablement lié au phénotype de l'angle de la pointe des racines. Notre approche interprète différemment : pour un marqueur qui peut être potentiellement lié au phénotype, l'erreur de classification doit être petite. Notre analyse a le même objectif que leur méthode, à la différence que nous ne faisons pas de test statistique ; notre analyse est donc descriptive. Afin de faciliter l'interprétation des résultats, nous allons comparer nos taux d'erreurs de classification aux résultats de référence en prenant 1 moins le score LOD pour chaque position des marqueurs. La figure 4.6 présente 233 erreurs de classification en fonction de la position du marqueur comparativement aux résultats de l'analyse quantitative multiple des locus de caractères quantitatifs. Les courbes en rouges représentent les erreurs de clas-

sification et les courbes en noires se réfèrent aux résultats de Kwak *et al.* (2016).

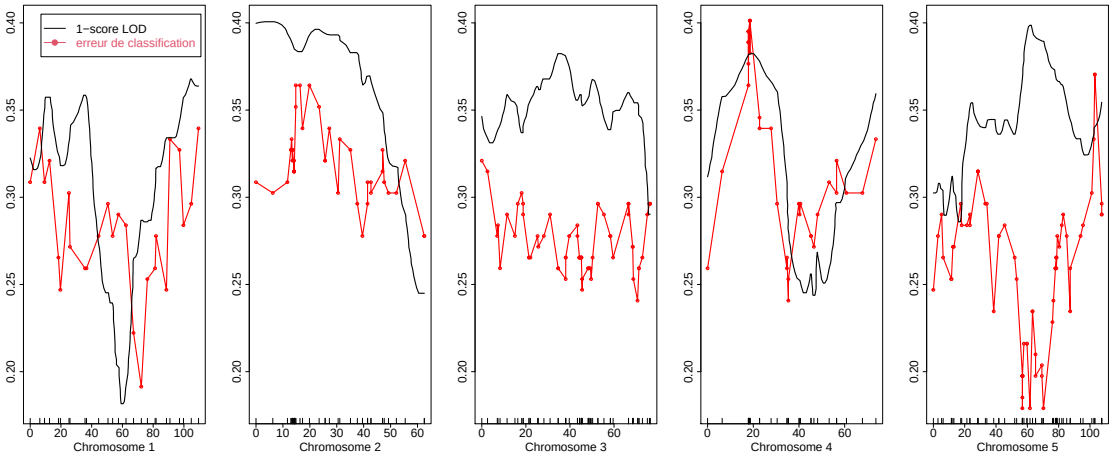

Figure 4.6: Comparaison des résultats de Kwak *et al.* (2016) et les résultats dérivés de la méthode de classification non-paramétrique.

4.4 Discussion

En regardant les taux de mauvaise classification des 5 graphiques de la figure 4.6, nos résultats indiquent que les marqueurs les plus liés au phénotype sont sur les chromosomes 1 et 5 où les taux d'erreur sont les plus faibles (plus petits que 0.20), alors que les chromosomes 2 et 3 semblent avoir des marqueurs moins liés. On peut remarquer que le marqueur pour lequel le taux d'erreur est de 0.24 sur le chromosome 4 semble nous donner un indice de liaison. En comparant le chromosome 4 par rapport à tous les autres chromosomes, on peut voir que cet indice n'est pas assez convaincant pour indiquer que le marqueur serait probablement pertinent.

Nous allons comparer les données génétiques par rapport aux marqueurs 209 et 229 sur le chromosome 5 afin de voir si leur différence est évidente. Ces marqueurs

correspondent respectivement aux taux d'erreurs pour lesquels le taux d'erreur de classification est le plus petit (0.18) et le taux d'erreur de classification est le plus élevé (0.37). Les graphiques 4.7 et 4.8 présentent les données brutes selon les classes *Cvi* et *Ler* pour les marqueurs 209 et 229 respectivement. Pour ces deux figures, les 162 phénotypes groupés selon les classes *Ler* (rouge) et *Cvi* (bleu) sont illustrées en (a). Les fonctions moyennes pour les groupes *Ler* (rouge) et *Cvi* (bleu) sont présentées à la figure (b). Les graphiques (c) et (d) présentent les fonctions de covariance pour le groupe *Ler* et *Civ* respectivement. Remarquons que les fonctions moyennes sont très similaires et que les fonctions de covariance sont un peu différentes. On constate que les fonctions moyennes sont presque identiques et que les fonctions de covariance sont différentes. Remarquons que même si les moyennes et les groupes sont presque homogènes, la méthode de classification arrive quand même à bien classer les courbes.

Nous pouvons voir que les profils de nos résultats pour les chromosomes 1 et 4 comparativement aux profils de Kwak *et al.* (2016) sont très similaires. Il est clair que nos résultats semblent logiques, car ils ont l'air à concorder avec ceux de Kwak *et al.* (2016). Il est évident qu'on n'obtienne pas exactement les mêmes résultats qu'aux résultats de référence, car les méthodes utilisées sont totalement différentes, mais on peut voir que notre approche par la méthode de classification non-paramétrique donne somme toute des résultats similaires.

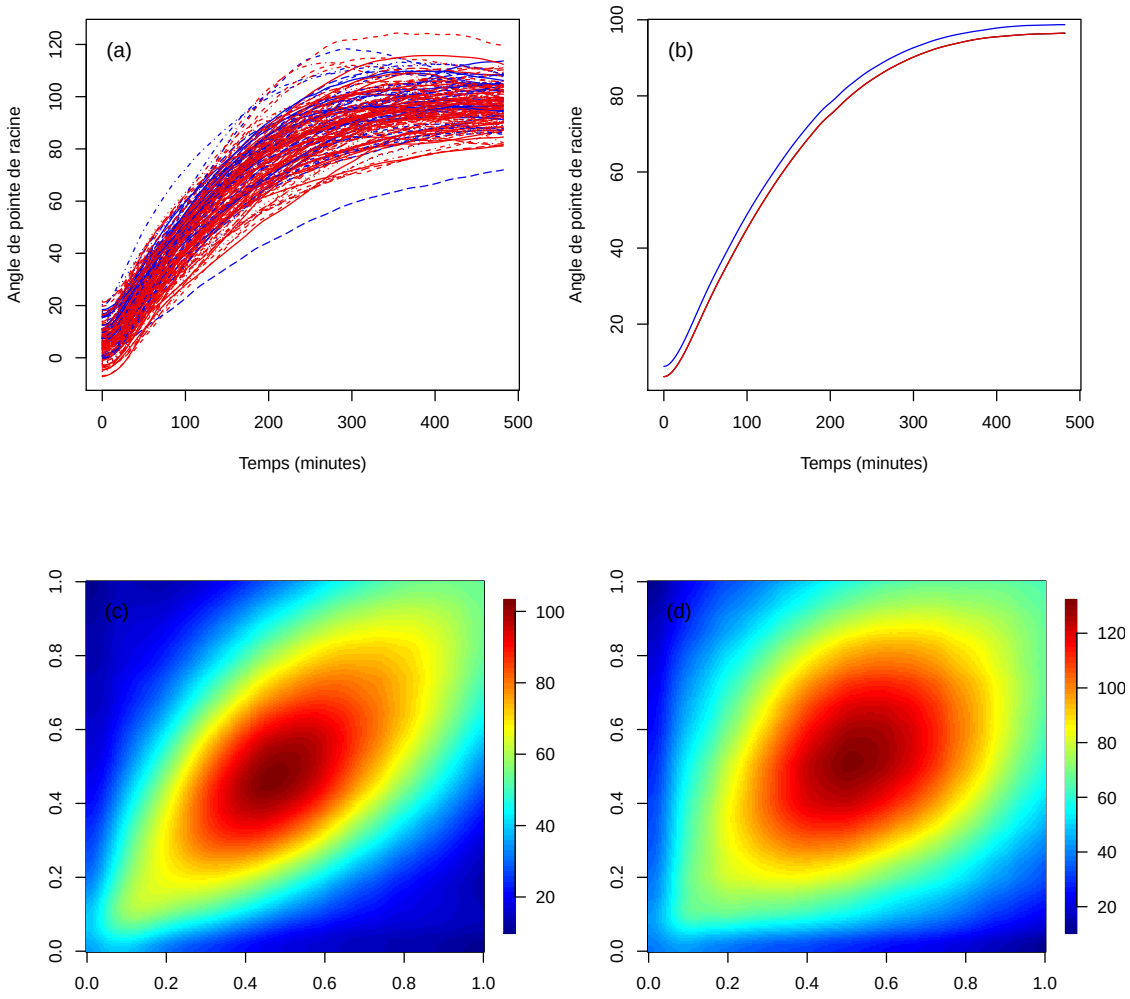

Figure 4.7: Marqueur 209 du chromosome 5. La figure (a) présente les 162 courbes selon le groupe *Ler* (rouge) et *Civ* (bleu). Le graphique (b) illustre les fonctions moyennes pour les classes *Ler* (rouge) et *Civ* (bleu), les figures (c) et (d) présentent les fonctions de covariance pour le groupe *Ler* et *Civ* respectivement.

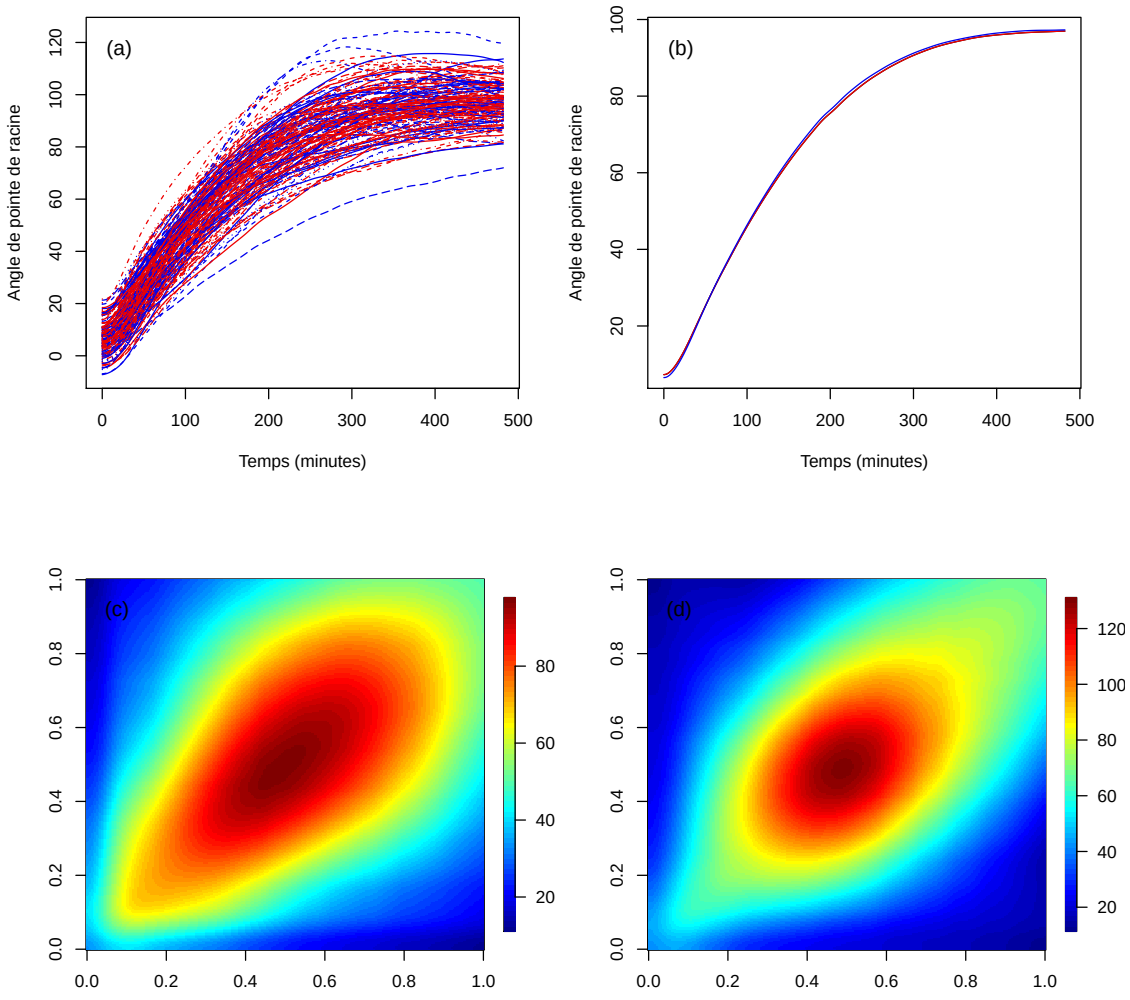

Figure 4.8: Marqueur 229 du chromosome 5. La figure (a) présente les 162 courbes selon le groupe *Ler* (rouge) et *Civ* (bleu). Le graphique (b) illustre les fonctions moyennes pour les classes *Ler* (rouge) et *Civ* (bleu), les figures (c) et (d) présentent les fonctions de covariance pour le groupe *Ler* et *Civ* respectivement.

CONCLUSION

L'objectif de ce mémoire était de proposer une nouvelle approche d'association fonctionnelle utilisant des outils décrits dans Ferraty et Vieu (2006), et la comparer à l'approche par vraisemblance de Kwak *et al.* (2016) afin d'étudier l'association entre un caractère et des marqueurs génétiques. Nous avons évalué le comportement de la méthode par une étude de simulation. Nous avons choisi trois scénarios pour lesquels chaque scénario varie par rapport aux fonctions moyennes et fonctions de covariance. Pour chaque scénario, trois niveaux de difficulté de classification ont été proposés afin d'étudier les différentes combinaisons possibles de semi-métrique et du type de noyau.

Après avoir présenté notre nouvelle approche permettant de mesurer l'association entre génotype et phénotype, et l'analyse brute de nos données génétiques de Moore *et al.* (2013), nous avons utilisé les résultats obtenus avec l'étude de simulation du chapitre 3, pour sélectionner la semi-métrique basée sur les dérivées et le noyau triangle dans le but de trouver s'il y a une association entre le caractère d'intérêt et les marqueurs génétiques des données génétiques. Notre approche semble indiquer que les marqueurs génétiques les plus liés au caractère d'intérêt, sont sur les chromosomes 1 et 5. Après avoir comparé les résultats obtenus avec notre méthode proposée à ceux de Kwak *et al.* (2016), nous avons obtenu des résultats similaires à ceux de Kwak *et al.* (2016) pour les chromosomes 1 et 4, alors que pour les chromosomes $2, 3$ et 5, nous avons obtenu des résultats différents. Nous avons constaté que même sans avoir à effectuer des tests statistiques, nous avons eu des resultats très similaires à l'analyse quantitative multiple des locus de caractères quantitatifs pour les chromosomes 1 et 4.

Ainsi, l'approche que nous proposons fonctionne assez pour poursuivre une étude plus approfondie dans cette voie.

APPENDICE A

RÉSULTATS D'EXPÉRIENCES DE SIMULATION

Cette annexe contient les résultats de la classification des courbes simulées en utilisant soit le noyau quadratique ou soit le noyau uniforme.

A.1 Résultats obtenus avec le noyau quadratique

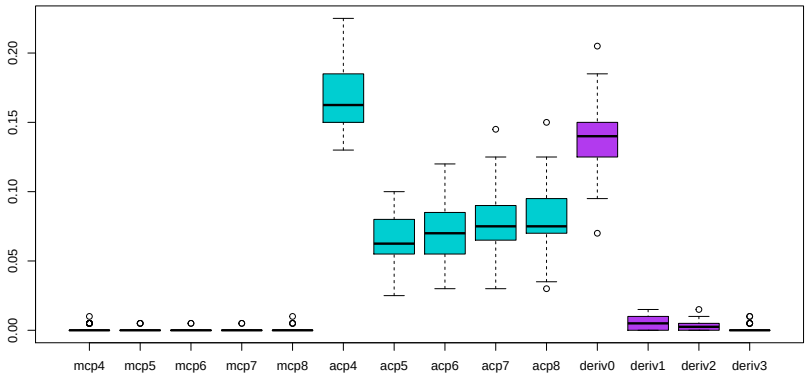

Figure A.1: Scénario 1 : cas facile. Présentation des taux d'erreur de classification.

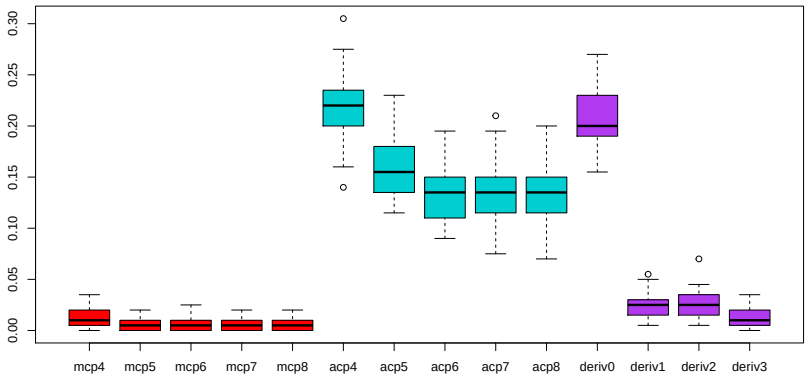

Figure A.2: Scénario 1 : cas moyen. Présentation des taux d'erreur de classification.

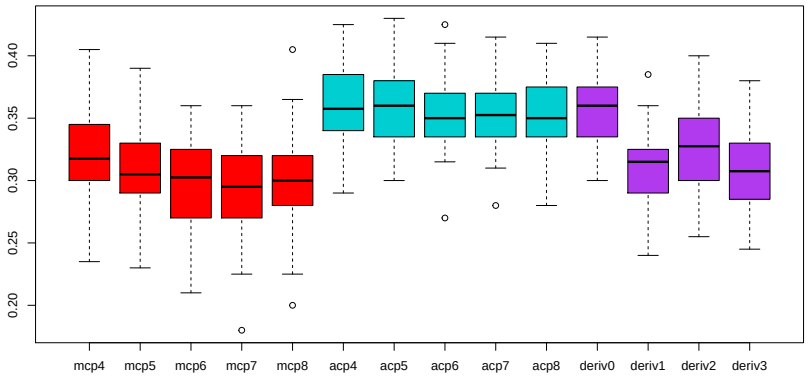

Figure A.3: Scénario 1 : cas difficile. Présentation des taux d'erreur de classification.

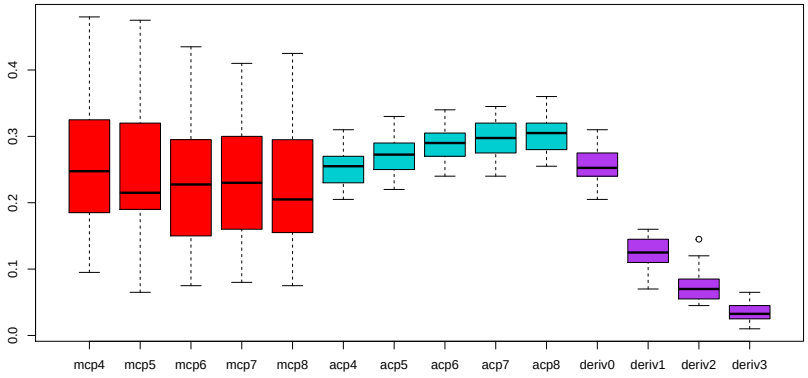

Figure A.4: Scénario 2 : cas facile. Présentation des taux d'erreur de classification.

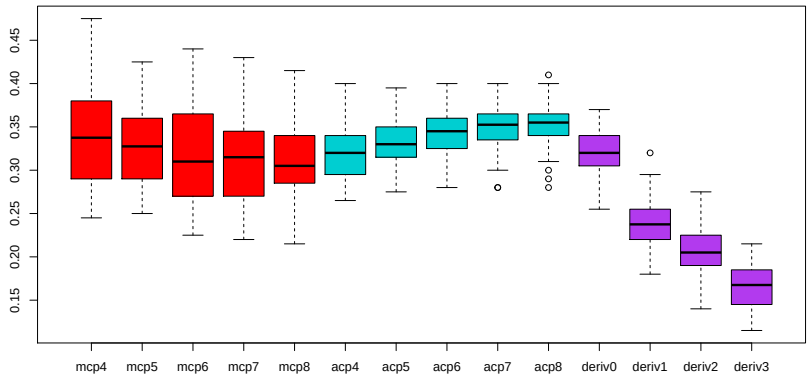

Figure A.5: Scénario 2 : cas moyen. Présentation des taux d'erreur de classification.

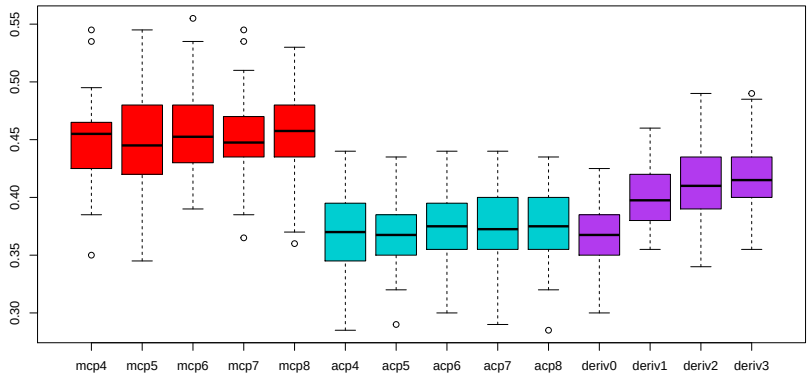

Figure A.6: Scénario 2 : cas difficile. Présentation des taux d'erreur de classification.

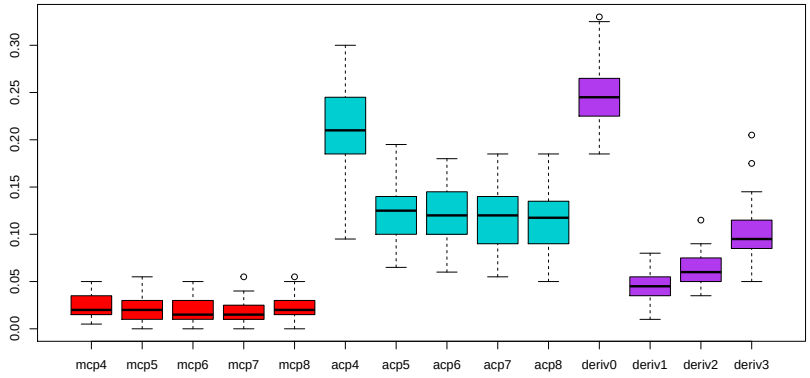

Figure A.7: Scénario 3 : cas facile. Présentation des taux d'erreur de classification.

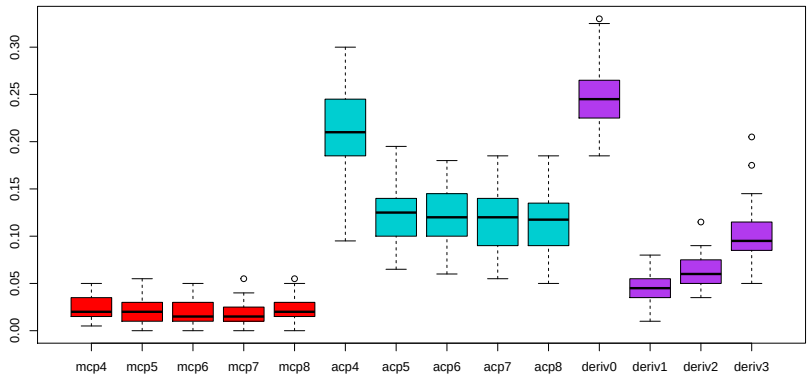

Figure A.8: Scénario 3 : cas moyen. Présentation des taux d'erreur de classification.

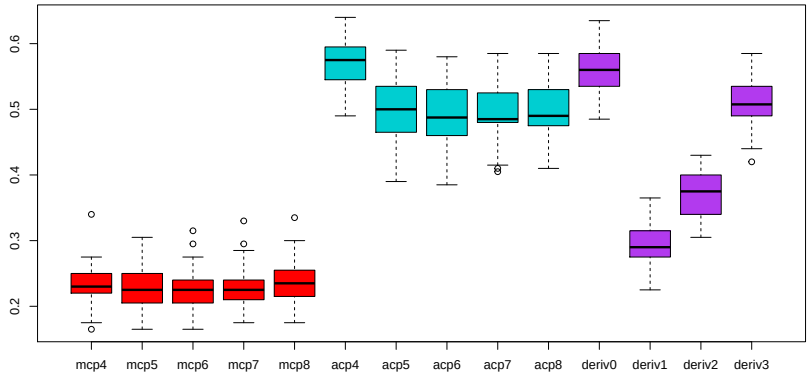

Figure A.9: Scénario 3 : cas difficile. Présentation des taux d'erreur de classification.

A.2 Résultats obtenus avec le noyau uniforme

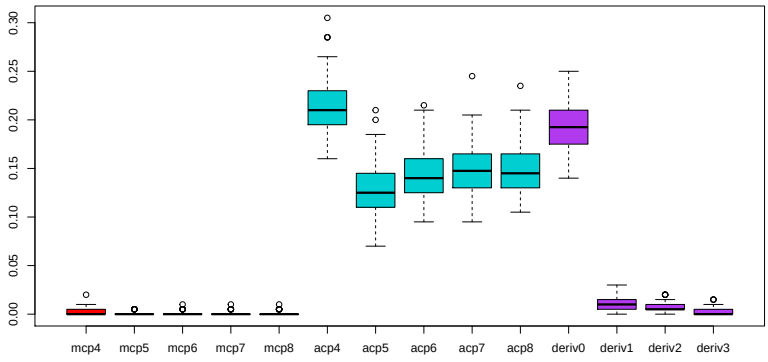

Figure A.10: Scénario 1 : cas facile. Présentation des taux d'erreur de classification.

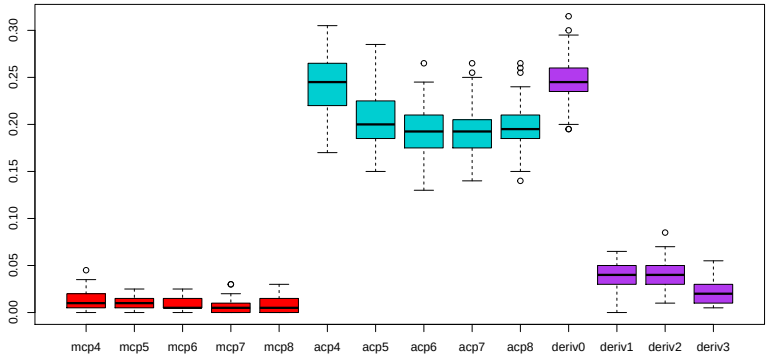

Figure A.11: Scénario 1 : cas moyen. Présentation des taux d'erreur de classification.

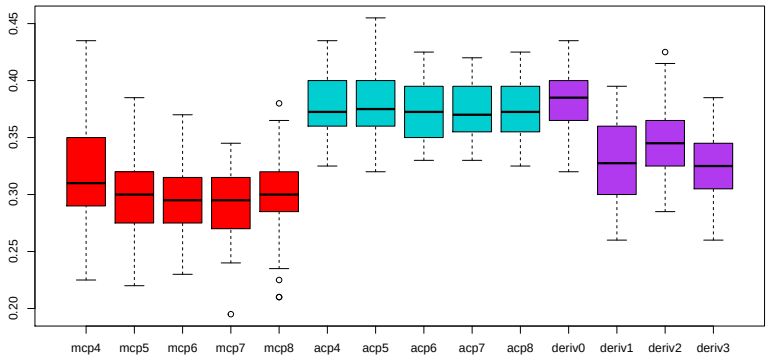

Figure A.12: Scénario 1 : cas difficile. Présentation des taux d'erreur de classification.

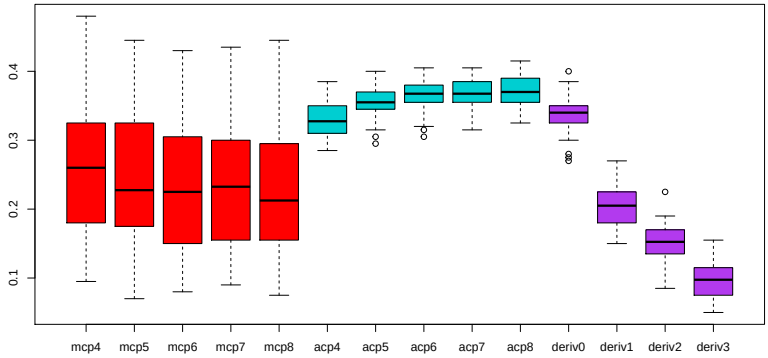

Figure A.13: Scénario 2 : cas facile. Présentation des taux d'erreur de classification.

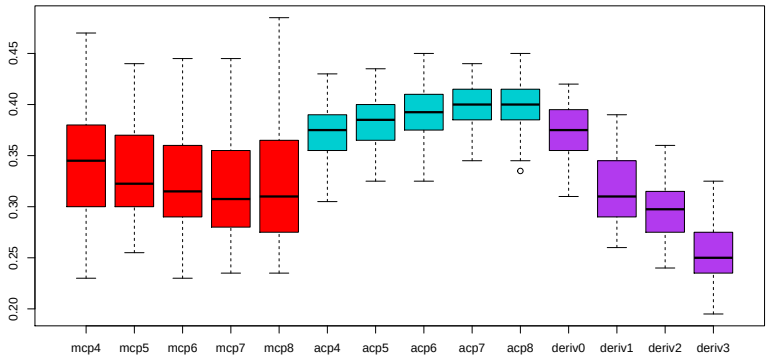

Figure A.14: Scénario 2 : cas moyen. Présentation des taux d'erreur de classification.

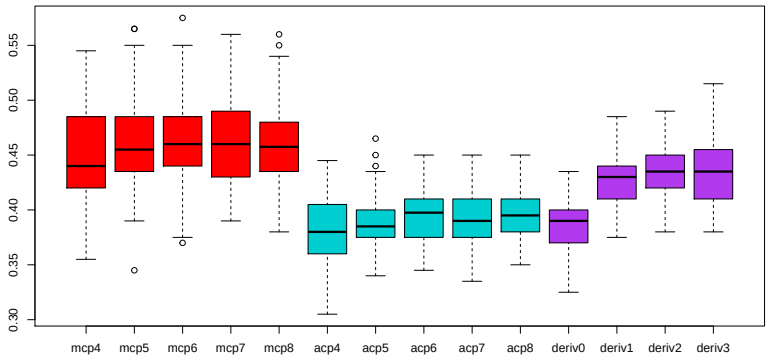

Figure A.15: Scénario 2 : cas difficile. Présentation des taux d'erreur de classification.

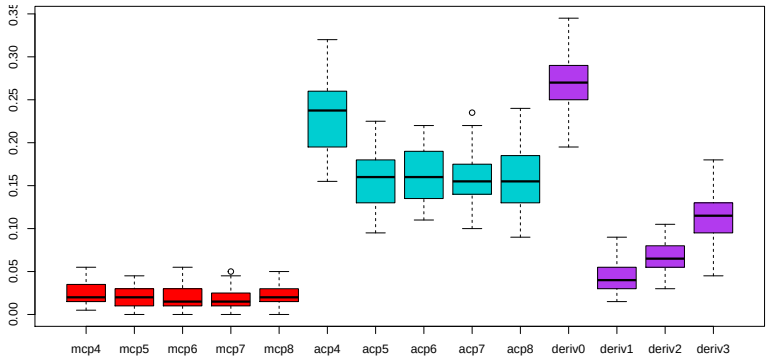

Figure A.16: Scénario 3 : cas facile. Présentation des taux d'erreur de classification.

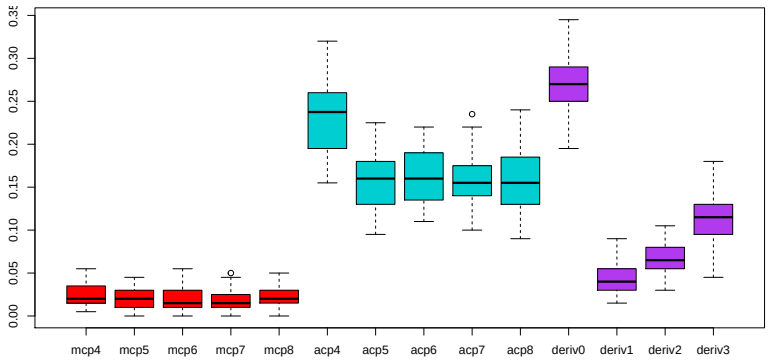

Figure A.17: Scénario 3 : cas moyen. Présentation des taux d'erreur de classification.

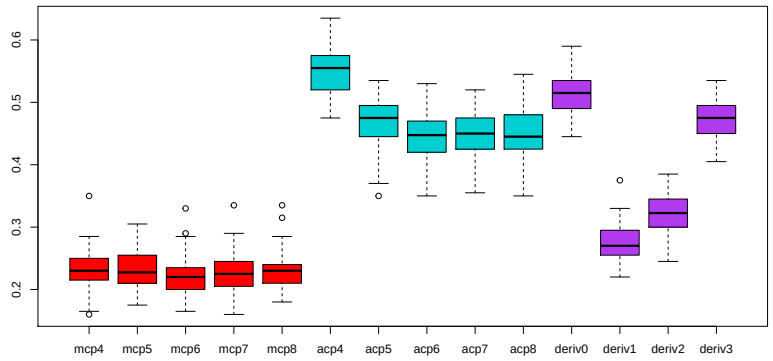

Figure A.18: Scénario 3 : cas difficile. Présentation des taux d'erreur de classification.

RÉFÉRENCES

Balding, D. J. (2006). A tutorial on statistical methods for population association studies. *Nature Reviews/Genetics*, *7*(10), 781–791.

Campbell, N. A. et Reece, J. B. (2012). *Biologie*. Éditions du Renouveau Pédagogique., (4 éd.).

de Boor, C. (2001). *A Practical Guide to Splines*. Springer., (Édition revisée éd.).

Delaigle Aurore, Hall, P. (2012). Achieving near perfect classification for functional data. *Royal Statistical Society*, *74*(2), 267–286.

Ferraty, F. et Vieu, P. (2006a). *Nonparametric Functional Data Analysis*. Springer.

Ferraty, F. et Vieu, P. (2006b). *Reference manual for implementing NonParametric Functional Data Analysis (NPFDA)*. Springer.

Hastie, T., Tibshirani, R. et Friedman, J. (2001). *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer., (2 éd.).

James, G., Witten, D., Hastie, T. et Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer., (4 éd.).

Kwak, I.-Y., Moore, C. R., Spalding, E. P. et Broman, K. W. (2016). Mapping quantitative trait loci underlying function-valued traits using functional principal component analysis and multi-trait mapping. *G3 : Genes/Genomes/Genetics*, *6*(1), 79–86.

Lajos, H. et Piotr, K. (2012). *Inference for Functional Data with Applications*. Springer.

Larribe, F. (2003). *Cartographie génétique fine par le graphe de recombinaison ancestral*. (Thèse de doctorat). Université de Montréal.

Leng, X. et Müller, H.-G. (2006). Classification using functional data analysis for temporal gene expression data. *Bioinformatics*, *22*(1), 68—-76.

Moore, C. R., Livny, M., Goldberg, Kwak, I.-Y., Broman, K. W. et Spalding, E. P. (2013). High-throughput computer vision introduces the time axis to a quantitative trait map of a plant growth response. *Genetics*, *195*(3), 1077–1086.

Nickle, T. et Barrette-Ng, I. Book : Online Open Genetics (Nickle and Barrette-Ng). `https://bio.libretexts.org/@go/page/3931`. [Online ; accessed 2021-11-01].

Ramsey, J.O. et Silverman, B. (1997). *Functional Data Analysis*. Springer., (2 éd.).

Ramsey, J.O. et Graves, S. (2009). *Functional Data Analysis with R and MATLAB*. Springer.