Mémoire de Master 2 / Spécialité: Probabilités et Modèles Aléatoires

Thème : **Inférence et Prédiction de données fonctionnelles binaires À l'aide de Processus Gaussiens multitâches**

Rédigé par:

Paguidame SAMBIANI

Superviseurs :

Sophie Dabo,

(Université de Lille)

Rim Essifi,

(Université de Paris Nanterre)

Arthur Leroy

(Université de Manchester)

Année académique : **2022-2023**

Remerciements

J'aimerais pour commencer, exprimer ma profonde gratitude à des personnes sans le concours desquelles ce travail n'aurait pas pu être réalisé.
J'exprime ma profonde gratitude au Professeure Sophie DABO de l'Université de Lille, pour m'avoir donné cette grande opportunité de travailler avec elle malgré ses multiples occupations. Merci pour tout chère Professeure.
Je remercie chaleureusement Docteure Rim Essifi (Post-Doctorante de l'équipe MODAL du Centre INRIA de Lille) qui m'a soutenu tout au long de mon stage par ses précieux conseils que par sa disponibilité à mon égard, par son suivi et correction de mon mémoire de stage, qu'elle trouve ici l'expression de toute ma gratitude.
J'aimerais également dire un sincère merci au Docteur Arthur Leroy (Post-doctorant à L'université de Manchester), pour sa constante disponibilité à mon égard tout au long de mon stage et dont la contribution a été crutiale pour l'implémentation de mon code en R.
Je remercie vivement le Professeur Cristian Preda, Chef de l'équipe MODAL du Centre INRIA de Lille pour son soutien et ses précieux conseils durant mon stage au Centre INRIA de Lille.
Je remercie vivement l'équipe de l'action exploratoire PATH de l'équipe MODAL du Centre INRIA de Lille et du laboratoire METRICS de l'université et du CHU de Lille.
Je remercie chaleureusement la Professeure Irina Kourkova, responsable du master Probabilités et Modèles aléatoires de Sorbonne Université qui m'a soutenu tout au long de mes études de master à Sorbonne Université aussi bien par ses enseignements que j'ai reçus et surtout par sa constante disponibilité à mon égard.
Je remercie également tous les Professeurs de Sorbonne Université pour la formation tant intellectuelle que morale qu'ils m'ont inculquée.
J'aimerais pour finir exprimer ma sincère reconnaissance à la Fondation Sciences Mathémtiques de Paris (FSMP) pour son aide financière et son suivi durant mon cursus de formation de master à Sorbonne Université.

Table des matières

CHAPITRE UN

INTRODUCTION

De nos jours, les systèmes de santé européens sont confrontés à de multiples défis, principalement le vieillissement de la population, l'augmentation des maladies chroniques et des patients souffrant de multi-morbidité, des ressources financières et humaines contraintes (Barnett et al, (2012) [40]). Il y a lieu donc de penser de façon urgente à une organisation des soins des patients dans un milieu hospitalier en parcours de soins. L'analyse des parcours de soins et de leur adéquation aux besoins et aux moyens est ainsi devenue un enjeu majeur sur les plans scientifiques et administratifs. cet effet, de nombreux auteurs scientifiques se sont penchés sur ce sujet, dont on peut citer par exemple les travaux de Threapleton et al, (2017) [41], Schrijvers et al, (2012) [42].
Bien que de nombreuses approches statistiques (clustering, régression, analyse de survie) de données complexes avec dépendance temporelle ou spatio-temporelle ont connu un essor important cette dernière décennie ([[44], [45], [46]], etc.), elles nécessitent d'être étendues aux données de parcours patient afin de répondre aux questions suivantes : identifier des parcours-type et des parcours atypiques, prédire des états futurs d'un parcours de soin, prédire des événements (certains récurrents) comme des ré-hospitalisations, un décès, des interventions, etc.

Dans ce sens, il exite une riche littérature sur les modèles statistiques joints, qui intègrent des variables temps/espace dépendantes permettant de tenir compte des modifications de médication et de l'état du patient, mais en pratique qui sont assez difficiles à implémenter et peu utilisés en clinique. Les modèles génératifs de parcours de soin quant à eux permettent de modéliser des données de parcours des soins de longueurs différentes avec des états multivariés et même avec des informations manquantes.
Ainsi donc dans cette étude, notre objectif est de développer un modèle génératif de clustering qui puisse prendre en considération de manière simultanée des informations pertinentes pour le clinicien, telles le parcours du patient. Ces informations sont traitées comme des variables dépendantes du temps et peuvent potentiellement être liées entre elles par des variables latentes.
Notre approche repose sur l'utilisation d'un modèle de prédiction basé sur des mélanges de processus gaussiens multitâches, comme décrit dans l'étude de Leroy et al (2022) [1]. Il s'agit d'un modèle probabiliste couramment employé en apprentissage automatique et en analyse de données pour modéliser des ensembles de données interdépendantes liées à une ou plusieurs tâches. Dans ce modèle, chaque tâche est représentée comme un mélange de processus gaussiens, avec un processus moyen commun spécifique à chaque tâche. Ce processus moyen commun aide à capturer les similitudes entre les différents individus au sein de la tâche en question.

L'avantage de cette approche réside dans le fait qu'elle va au-delà d'une simple régression avec des processus gaussiens, qui fournit un ajustement convenable près des points de données, mais qui perd rapidement de sa précision en l'absence d'informations. Au contraire, ce nouveau modèle tire parti de la composante multitâches pour partager les informations entre les individus en estimant un processus moyen plus pertinent. Cela conduit à une amélioration significative de la prédiction moyenne, tout en réduisant l'incertitude entourant cette prédiction.

Ce qui distingue notre approche, c'est principalement la nature catégorielle des données liées à l'évolution du patient, qui peut être adaptée au formalisme décrit par Leroy et al (2022) [1] en utilisant un modèle de variables latentes suivant le même schéma. De plus, notre méthode d'estimation est stochastique, ce qui signifie que nous introduisons une composante stochastique dans l'étape E d'une méthode d'estimation EM des hyperparamètres du modèle. Cette procédure est mise en œuvre via R et représente la variante stochastique du package "MagmaclustR" de Leroy et al (2022) [1].
Ce travail s'inscrit dans le cadre du projet PATH, co-porté par Sophie Dabo PR en statistique, très impliquée dans les applications médicales et membre de MODAL et Jean-Baptiste Beuscart PU-PH clinicien en gériatrie et directeur de l'équipe METRICS, dont l'objectif est de permettre une meilleure compréhension des étapes clés dans les parcours de soins des patients en associant les producteurs de données au plus proche du patient, ceux qui les gèrent, ceux qui les pré-traitent, ceux qui les analysent, pour avoir au final un résultat au plus proche du terrain, et un retour vers le clinicien et le patient, le plus efficient possible. Cet objetif sera abordé à travers deux applications portant sur la ré-hospitalisation de la personne âgée et les complications postopéatoires.
Notre travail est structuré comme suit : nous commençons par une brève introduction à l'analyse fonctionnelle des données, où nous passons en revue quelques outils de lissage et de réduction de dimension. Ensuite, nous examinons les méthodes d'estimation des paramètres en statistique, en mettant particulièrement l'accent sur l'algorithme EM et ses variantes, qui sont largement utilisés dans les modèles impliquant des variables latentes, comme c'est le cas dans notre étude. Nous poursuivons en détaillant un exemple concret d'utilisation de la forme variationnelle de l'algorithme EM, tel que présenté par Leroy et al (2022) [1].

La partie centrale de notre travail consiste en la présentation du formalisme mathématique du nouveau modèle de clustering pour les données binaires, et de l'adoption d'une variante stochastique de l'algorithme EM (que nous notons SEM par la suite) pour l'estimation des hyper-paramètres. Nous terminons par un exposé sur la procédure d'implémentation de notre nouvelle approche via R, et donnons quelques exemples de sorties numériques illustratives de l'agorithme SEM.

CHAPITRE DEUX

ANALYSE DE DONNÉES FONCTIONNELLES À L'AIDE DE PROCESSUS GAUSSIENS

2.1 Introduction aux données fonctionnelles

2.1.1 Concept de données fonctionnelles

Considérons l'exemple d'une situation où nous mesurons la température d'une ville chaque jour pendant plusieurs années. En regardant ensuite ces données accumulées, on pourrait arriver à la conclusion que la température se comporte à peu près de manière similaire sur une année. En fait, on pourrait même s'imaginer qu'il y a une fonction sous-jacente du temps, qui donnerait lieu à nos mesures. On voudrait donc interpréter nos données mesurées non pas comme une séquence d'observations, mais comme des fonctions ou des courbes, chaque individu étant alors assimilé à une courbe. Dans la pratique, nous évoluons dans le cadre des données fonctionnelles lorsqu'on analyse en même temps une collection de séries temporelles ou de trajectoires d'un processus stochastique, l'unité d'observation étant alors une courbe ou un vecteur de courbes. Ce nouveau type de données, appelées « données fonctionnelles », est de nos jours impliqué dans de nombreux domaines d'application tels que la médecine (les mesures électro encéphalographiques, courbes de croissance et de poids, etc. Voir Sørensen et al (2013) ; Shangguan et coll (2020)), l'économie (dans l'analyse du cours des actions, voir par exemple Ramsay et Ramsey (2002)), sciences de l'environnement (mesure de la concentration d'un polluant toutes les heures pendant un mois, voir Cardot et coll (2007) ; Bouveyron et coll (2022)), etc.

2.1.2 Avantages et inconvenients du cadre fonctionnel des données

Comme avantages d'adopter un cadre fonctionnel pour des données, on peut citer :

— La conservation de la structure : contrairement à d'autres méthodes d'analyse de données qui considèrent chaque observation comme une valeur distincte, le cadre des données fonctionnelles conserve la structure temporelle ou spatiale des données en les traitant comme des fonctions. Cela permet de capturer des informations plus fines sur les tendances dans les données.

— Réduction de la dimensionnalité : Les données fonctionnelles peuvent souvent être décrites de manière plus compacte en utilisant des représentations fonctionnelles (séries de Fourier ou splines) plutôt qu'en utilisant toutes les observations individuelles. Cela permet de réduire la dimensionnalité des données et facilite ainsi l'analyse.

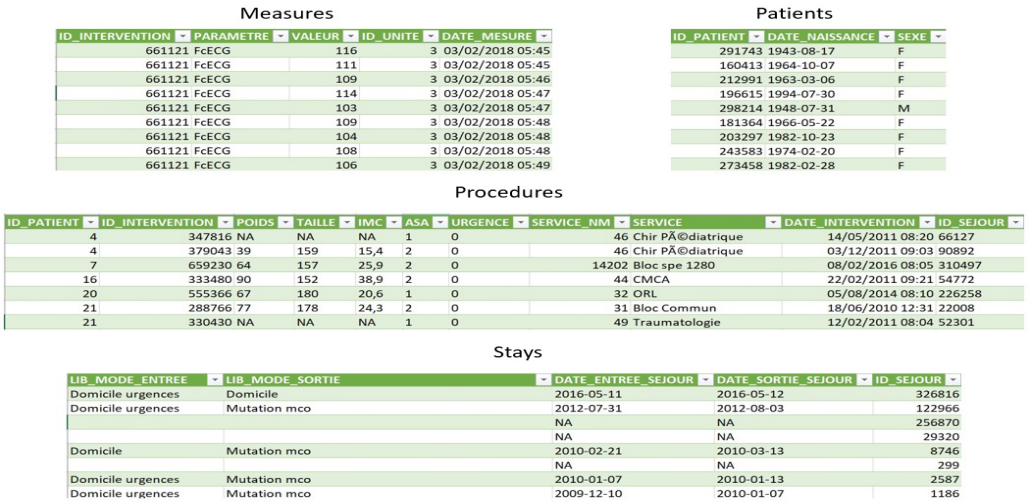
Cependant, le cadre fonctionnel peut entraîner d'autres complications pour la manipulation et la modélisation des données :

— La complexité computationnelle : d'un point de vue computationnel, la complexité dans le cadre fonctionnel peut être plus complexe par rapport à d'autres méthodes statistiques traditionnelles. Les calculs impliqués dans la manipulation et la modélisation des fonctions peuvent nécessiter plus de ressources informatiques et de temps de calcul.

— Interprétation complexe : L'interprétation des résultats de l'analyse des données fonctionnelles peut être plus complexe en raison de la nature abstraite des fonctions. Comprendre les coefficients, les poids ou les paramètres des modèles fonctionnels peut nécessiter une expertise supplémentaire dans le domaine d'application spécifique.

2.1.3 Exemples des données de parcours de soins

Dans le cadre de notre travail par exemple, les données dont dispose l'équipe MODAL de l'Inria de Lille sur les patients sont toutes dépendantes du temps, pour la plupart hétérogènes, qualitatives à plusieurs milliers de modalités possibles, principalement constituées de données manquantes, représentées dans des dizaines ou centaines de tables en relation et le plus souvent spatiales. Ces données sont issues d'une cohorte DAMAGE, cohorte prospective de 3532 personnes âgées hospitalisées en gériatrie et suivies durant 1 an, ainsi que d'une cohorte PAERPA Hauts-de-France, qui contient les données de plus de 40 000 personnes âgés de 75 ans ou plus vivant dans le Valenciennois-Quercitain (population réelle totale du territoire considéré) ; nous disposons également de données provenant de l'EDS du CHU de Lille (projet INCLUDE), et les données du Système Nationale des Données de Santé (SNDS, Health Data Hub) dont l'équipe METRICS a l'expertise. L'EDS INCLUDE contient les données de routine de plus d'1,5 million de patients venus aux CHU de Lille au cours des 12 dernières années, et est alimenté à partir des logiciels hospitaliers composant le dossier patient informatisé. Ces logiciels collectent les actes médicaux, les diagnostics, les administrations de médicaments, les résultats de biologie médicale, les passages dans les unités de soins, les signes vitaux, les échelles d'évaluation, et l'ensemble des données textuelles produites dans le cadre du soin.



Measures

ID_INTERVENTION	PARAMETRE	VALEUR	ID_UNITE	DATE_MESURE
661121	FcECG	116	3	03/02/2018 05:45
661121	FcECG	111	3	03/02/2018 05:45
661121	FcECG	109	3	03/02/2018 05:46
661121	FcECG	114	3	03/02/2018 05:47
661121	FcECG	103	3	03/02/2018 05:47
661121	FcECG	109	3	03/02/2018 05:48
661121	FcECG	104	3	03/02/2018 05:48
661121	FcECG	108	3	03/02/2018 05:48
661121	FcECG	106	3	03/02/2018 05:49

Patients

ID_PATIENT	DATE_NAISSANCE	SEXE
291743	1943-08-17	F
160413	1964-10-07	F
212991	1963-03-06	F
196615	1994-07-30	F
298214	1948-07-31	M
181364	1966-05-22	F
203297	1982-10-23	F
243583	1974-02-20	F
273458	1982-02-28	F

Procedures

ID_PATIENT	ID_INTERVENTION	POIDS	TAILLE	IMC	ASA	URGENCE	SERVICE_NM	SERVICE	DATE_INTERVENTION	ID_SEJOUR
4	347816	NA	NA	NA	1	0	46	Chir PÃ©diatrique	14/05/2011 08:20	66127
4	379043	39	159	15,4	2	0	46	Chir PÃ©diatrique	03/12/2011 09:03	90892
7	659230	64	157	25,9	2	0	14202	Bloc spe 1280	08/02/2016 08:05	310497
16	333480	90	152	38,9	2	0	44	CMCA	22/02/2011 09:21	54772
20	555366	67	180	20,6	1	0	32	ORL	05/08/2014 08:10	226258
21	288766	77	178	24,3	2	0	31	Bloc Commun	18/06/2010 12:31	22008
21	330430	NA	NA	NA	1	0	49	Traumatologie	12/02/2011 08:04	52301

Stays

LIB_MODE_ENTREE	LIB_MODE_SORTIE	DATE_ENTREE_SEJOUR	DATE_SORTIE_SEJOUR	ID_SEJOUR
Domicile urgences	Domicile	2016-05-11	2016-05-12	326816
Domicile urgences	Mutation mco	2012-07-31	2012-08-03	122966
		NA	NA	256870
		NA	NA	29320
Domicile	Mutation mco	2010-02-21	2010-03-13	8746
		NA	NA	299
Domicile urgences	Mutation mco	2010-01-07	2010-01-13	2587
Domicile urgences	Mutation mco	2009-12-10	2010-01-07	1186

FIGURE 2.1 – Vue globale d'un exemple de données de parcours de soins

Le cadre fonctionnel paraît donc adéquat dans ce contexte, où nous interprétons nos données comme des réalisations d'une v.a dans un espace (complexe) de courbes suivant le formalisme :

$$\mathbf{Y_t} = \left\{ (Y_{t_1}, \ldots, y_{t_d})^\top : t_j \in T_j, j = 1, \ldots, d \right\}, \quad \text{avec}$$

$$Y_{t_j} : \mathcal{P}_j \to \mathcal{S}_j$$

où suivant la nature de nos données, on distingue différentes possibilités pour les T_j et $\mathcal{S}_j$:

— $T_j \subseteq \mathbb{R}$, $\mathcal{S}_j = \mathbb{R}$ pour les courbes

— $T_j \subseteq \mathbb{R}$, $\mathcal{S}_j = \{e_1, e_2, \ldots, e_K\}$, pour des séquences de valeurs

— $T_j \subseteq \mathbb{R}^2$, $\mathcal{S}_j = \mathbb{R}$, pour des images ou des surfaces

2.2 Objets mathématiques pour le cadre fonctionnel

Définition 2.2.1. ***L'espace*** $L^2(T)$
Soit T, un intervalle quelconque de $\mathbb{R}$. *L'ensemble* $\left(L^2(T), \|\cdot\|, \langle\cdot,\cdot\rangle\right)$, *appellé espace des fonctions de carré intégrables est définit par*

$$L^2(T) = \left\{ x : T \to \mathbb{R}, \int_T x^2(t)dt < \infty \right\},$$

muni du produit scalaire défini par : $\forall x, y \in L^2(T), \langle x, y\rangle = \int_T x(t)y(t)dt$, *qui induit la norme d'un élément* $x \in L^2(T)$ *donnée par* $\|x\| = \sqrt{\langle x, x\rangle} = \left(\int_T x^2(t)dt\right)^{1/2}$, *ce qui fait de* $\left(L^2(T), \langle\cdot,\cdot\rangle\right)$ *un espace de Hilbert .*

Définition 2.2.2. ***Variable aléatoire fonctionnelle***
Soit $(\Omega, \mathcal{A}, \mathbb{P})$ *un espace probabilisé et* $(E, \mathcal{E})$ *un espace mesurable. On appelle variable aléatoire de* $\omega \mapsto E$, *toute fonction mesurable X de* $\omega \mapsto E$. *X est dite variable aléatoire fonctionnelle dans la cas où l'espace E est de dimension infinie, en général* $E = L^2(T)$.

Définition 2.2.3. ***Espérance d'une variable fonctionnelle***
Soit X une variable aléatoire fonctionnelle à valeurs dans $L^2(T)$. Si $\mathbb{E}[\|X\|] < \infty$, on définit l'espérance de X comme

$$(\mathbb{E}[X])(t) = \mathbb{E}[X(t)] = \int_\Omega X(t,\omega)d\mathbb{P}(\omega), \quad \forall t \in T$$

Dans la suite, on se placera dans le cas $E = L^2(T)$.

Définition 2.2.4. ***Opérateur de covariance***
Soit X une variable fonctionnelle et le processus linéaire $U(f) = \int_T X(t)f(t)dt$. Alors U est un opérateur de Hilbert-Schmidt, i.e que pour toute base orthogonale $(f_i)_i$ de $L^2(T)$, la quantité $\sum_i \|U(f_i)\|^2$ est finie et indépendante de la base que nous choisissons.
Si $\mathbb{E}\left[\|X\|^2\right] < \infty$, on définit l'opérateur de covariance de X comme l'opérateur de $L^2(T)$ dans lui-même définit par :

$$\langle f, \Gamma g\rangle = \mathbb{E}\Big[\langle f, X - \mathbb{E}[X]\rangle\langle g, X - \mathbb{E}[X]\rangle\Big] = \mathbb{E}\left[U(f)U(g)\right], \quad f,g \in L^2(T)$$

Proposition 2.2.5. *Si $\mathbb{E}\left[\|X\|^2\right] < \infty$, l'opérateur de covariance $\Gamma : L^2(T) \to L^2(T)$ de X est*
❶ *Linéaire,*
❷ *Continu : $\forall f \in L^2(T), \|\Gamma f\| \leq \mathbb{E}\left[\|X\|^2\right]\|f\|$,*
❸ *Autoadjoint : $\forall f,g \in L^2(T), \langle \lceil f, g\rangle = \langle f, \Gamma g\rangle$,*

❹ *Compact : en effet on montre que $\Gamma = U^* \circ U$, ce qui prouve que Γ qui est positif et autoadjoint est également nucléaire comme produit d'opérateurs de Hilbert-Schmidt.*

❺ *Diagonalisable en base orthonormée (décomposition des opérateurs autoadjoints compacts).*

Définition 2.2.6. ***Fonction de covariance***
Soit T un intervalle quelconque de $\mathbb{R}$ et considérons $X : \Omega \to L^2(T)$ une variable alétoire fonctionelle. La fonction de covariance de X est l'application :

$$V : (s,t) \in T^2 \longmapsto V(s,t) = \mathrm{Cov}(X(s), X(t)).$$

Si X est centrée (i.e $\mathbb{E}[X] = 0$), alors $V(s,t) = \mathbb{E}[X(s)X(t)]$.

Proposition 2.2.7. *Soit Γ, l'opérateur de covariance de X, une variable fonctionnelle centrée et C sa fonction de covariance. Alors $\forall f \in L^2(T), \forall t \in T$,*

$$(\Gamma f)(t) = \int_T V(s,t)f(t)dt.$$

On en déduit donc que Γ est un opérateur à noyau, de noyau la fonction de covariance.

Les données fonctionnelles dont on dispose peuvent le plus souvent avoir une dimensionnalité élevée, ce qui peut rendre le clustering difficile. L'analyse de données fonctionnelles dans ce contexte précis constitue le premier outil principal d'analyse et de clustering permettant de réduire la dimensionnalité en utilisant des techniques telles que l'Analyse en Composantes Principales Fonctionnelles (ACPF) que nous présentons dans la suite.

2.3 Analyse de données fonctionnelles

L'analyse de données fonctionnelles (FDA) étend les méthodes multivariées classiques lorsque les données sont des fonctions ou des courbes, principalement en l'analyse factorielle de données fonctionnelles. Cette dernière repose principalement sur la décomposition de Karhunen-Loève d'un processus continu $(X_t)_t$ de carré intégrable. Le sujet est largement étudié dans Deville (1974) [29], et par Besse [12], Saporta [30] qui étendent aux données fonctionnelles l'analyse en composantes principales, l'analyse des correspondances multiples pour les données fonctionnelles catégorielles et la régression linéaire sur les données fonctionnelles. De récents travaux sur les modèles de régression pour données fonctionnelles sont dus au groupe de recherche travaillant sur les statistiques fonctionnelles à Toulouse (STAPH), et citons également les travaux fondateurs de Ramsay et Silverman [21], [31], qui adaptent les méthodes statistiques classiques au cadre fonctionnel, le livre de Bosq [32] pour la modélisation des variables aléatoires fonctionnelles dépendantes et le livre récent de Ferraty et Vieu[25] sur les modèles non paramétriques pour données fonctionnelles contenant une revue des contributions les plus récentes sur ce sujet.

2.3.1 Formalisme des données fonctionnelles

On appelle données fonctionnelles, des réalisations $(X_1, \cdots, X_n)$ d'une certaine variable aléatoire fonctionnelle X, et pour une bonne définition de ce modèle, on regarde en général ces variables comme des réalisations d'un processus stochastique $(X_t)_{t\in T}$ à valeurs dans un espace de Hilbert H de fonctions définies sur un intervalle de temps T. On se contente ici du cas où H est un espace de fonctions à valeurs réelles. Pour les données fonctionnelles multivariées, c'est-à-dire où les fonctions de H sont à valeurs dans $\mathbb{R}^p$, $p \geq 2$, on pourra se référer par exemple à J. Jacques et C. Preda, 2013[18] pour un travail récent sur le clustering de données fonctionnelles multivariées.

Toutefois, lorsque nous parlons de données fonctionnelles, nous avons à l'idée que nos observations vivent dans un espace de dimension infinie, alors que dans la pratique, nous ne disposons que de courbes échantillonnées observées en un ensemble fini d'instants. C'est à dire qu'on fait un passage des données brutes

$$X_{ij} = X_i(t_{ij}), \quad t_{ij} \in [a,b], \quad i = 1, \cdots, n, j = 1, \cdots, J$$

$$\text{à des courbes lisses} \quad X_i(t), \quad t \in [a,b], \quad i = 1, \cdots, n, j = 1, \cdots, J.$$

La figure 2.2 ci-dessous illustre bien ce fait.

Pour cette raison, un premier travail consiste à reconstruire la forme fonctionnelle des données à partir d'observations discrètes, qui consiste à les exprimer au moyen d'une expansion de base. Cette manière de procéder est appelée "lissage des données".

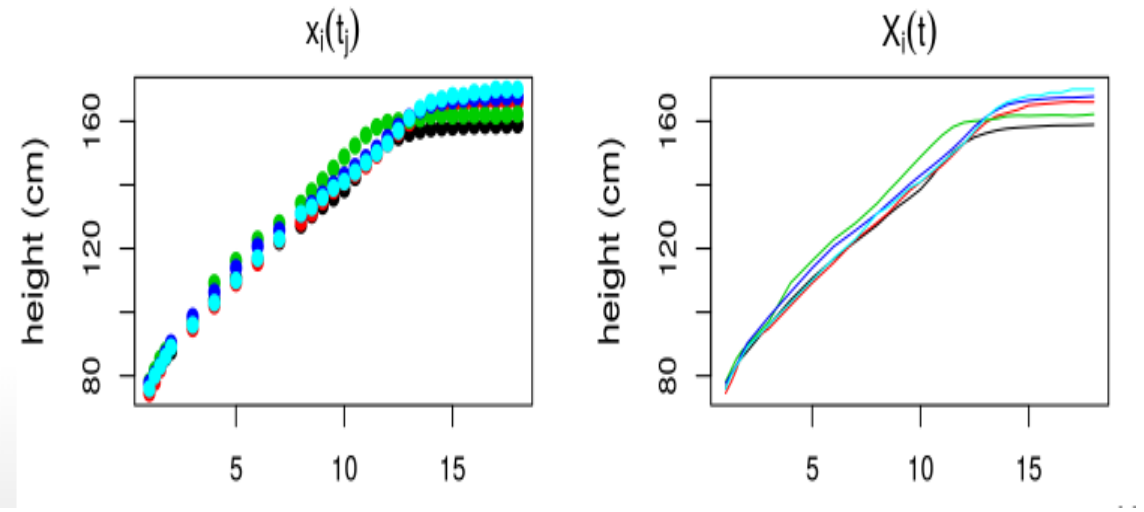


FIGURE 2.2 – Des données brutes aux donneés fonctionnelles

2.4 Lissage Paramétrique

2.4.1 Quelques définitions

Définition 2.4.1. ***Spline***
Une spline sur $T = [a,b]$ est une fonction polynomiale par morceaux, avec conditions de continuité sur la fonction et ses dérivées aux jointures. Elle est caractérisée par :

- des noeuds ("knots") $\tau_0 = a \leq \tau_1 \leq \ldots \leq \tau_L = b$, non nécessairement répartis régulièrement, non nécessairement distincts.

- un ordre m (= degré maximal des polynômes sur les sous-intervalles +1)

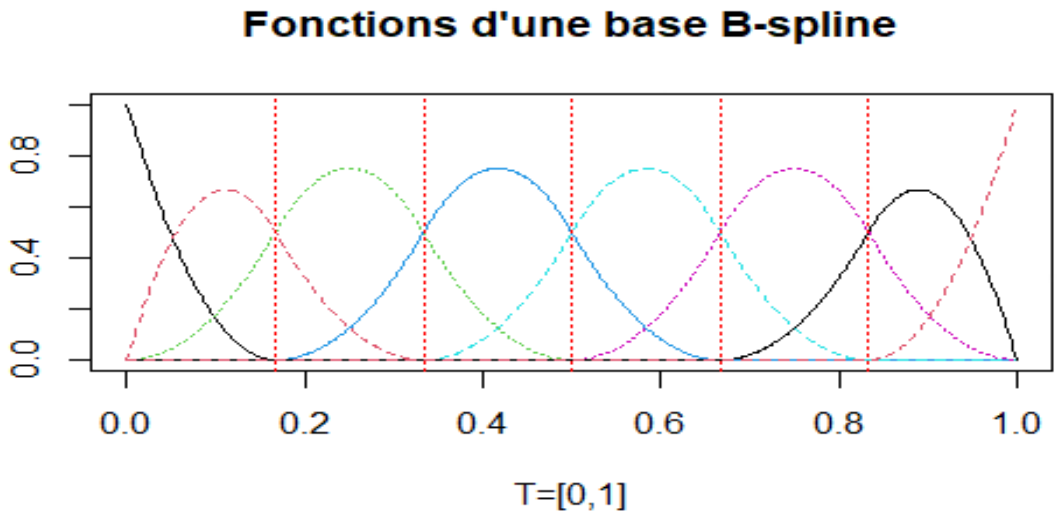
- des dérivées continues sur T jusqu'à l'ordre $m-2$.

Définition 2.4.2. ***Base de Splines***
Une base de spline d'ordre m et de séquence de noeuds τ est une famille de fonctions telles que :

(i) chaque fonction de base est une spline (toute combinaison linéaire de ces fonctions est donc encore une fonction spline) ;
(ii) toute spline d'ordre m et de séquence de noeuds τ peut s'exprimer comme combinaison linéaire de ces fonctions de base ;
(iii) les fonctions de bases sont linéairement indépendantes.

Il existe plusieurs bases classiques de splines, utilisées le plus souvent en analyse de données fonctionnelles dont la plus célèbre est la base de B-splines (de Boor, 2001[20]).
Voici un exemple de fonctions d'une base B-splines d'ordre 3

```
bspl_bf <- create.bspline.basis(rangeval=c(0,1), norder=3,
                                breaks=seq(0,1,len=7))
plot(bspl_bf, main="Base de fonctions B-spline", xlab="[a,b]=[0,1]")
```



Bien que les splines procurent une calculabilité facile surtout pour des polynômes changeant de comportements localement et une grande flexibilité, il y a la question de l'emplacement exacte des nœuds τ_k. Certaines applications suggèrent fortement des emplacements de nœuds spécifiques, mais ce choix se fait le plus souvent arbitrairement et dans d'autres cas il est suggéré commençer avec un ensemble dense de nœuds, puis éliminer les noeuds inutiles par un procédure algorithmique similaire aux techniques de sélection de variables utilisées dans la régression multiple.

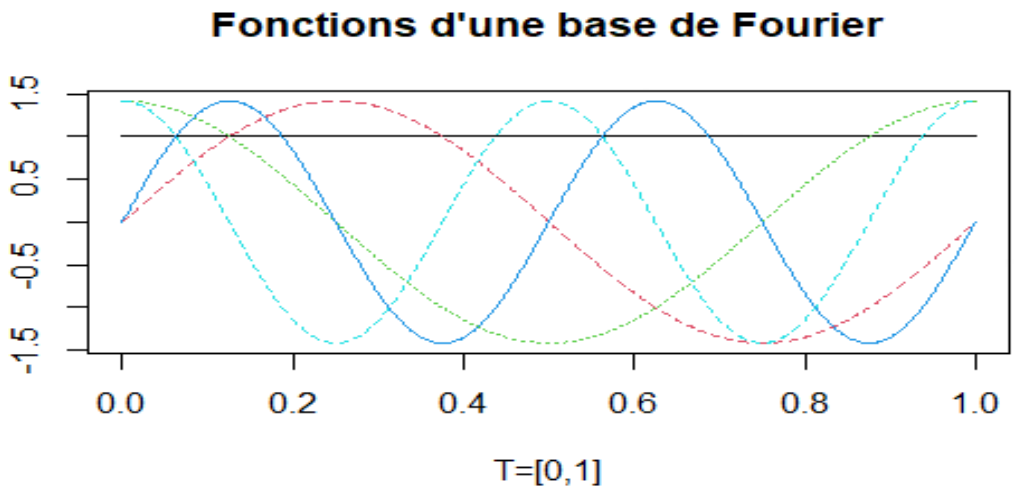
Définition 2.4.3. ***Base de Fourier***
On appelle base de Fourier, la base de fonctions définie de la manière suivante : $\forall t$,

$$\psi_0(t) = 1, \psi_{2k-1}(t) = \sin(k\omega t), \psi_{2k}(t) = \cos(k\omega t) \quad k \in \mathbb{N}\backslash\{0\}.$$

où la période $= 2\pi/\omega$ *et les coefficients de Fourier sont les coefficients du développement d'une fonction dans cette base.*

Voici un exemple d'une base de Fourier composée de 5 fonctions :

```
fourier_bf <- create.fourier.basis(rangeval=c(0,1), nbasis=5)
plot(fourier_bf, main="Functions d'une base de Fourier", xlab="T=[0,1]")
```



La base de Fourier est particulièrement utilisée pour des donnés périodiques. Mais elle est inappropriée pour des données présentant une certaine discontinuité en la fonction sous jacente ou dans ses dérivés d'ordre inférieur.

Définition 2.4.4. ***Base polynomiale***
La base monomial $\Phi_k(t) = (t-\omega)^k, \quad k = 1, \cdots, K$ *est très classique, où le paramètre* ω *est à bien choisir pour une meilleure expansion.*
Cette base est incapable de capter de vraies caractéristiques locales de fonctions sans utiliser un K très grand. Les polynômes ont tendance à bien s'adapter au centre des données, mais présentent un comportement plutôt peu attrayant dans les queues.

2.4.2 Lissage Paramétrique

Une méthode en générale utilisée pour le lissage de données X_i est de considérer sa décomposition dans une base de fonctions de la forme $\{e_k(t), \quad t \in T, 1 \leq k \leq K, \quad \text{pour un certain} \quad K \in \mathbb{N}\}$:

$$X_i(t) = \sum_{k=1}^{K} a_{i,k} f_k(t), \quad i = 1, \ldots, n \tag{2.1}$$

De façon équivalente, cela revient à considérer X_i sous la forme

$$X_i(t) = f(t) a_i$$

où les coefficients

$$a_i = (a_{i,1}, \ldots, a_{i,K})^\top$$

sont estimés à partir des observations discrètes dont on dispose, à l'aide d'une méthode numérique appropriée :

➢ Par interpolation si l'on suppose qu'on dispose d'observations sans termes d'erreur de la forme

$$X_{ij} = X_i\left(t_{ij}\right) \quad j = 1,\ldots,m_i,$$

➢ Sinon si on dispose d'observations avec des erreurs de la forme

$$X_{ij} = X_i\left(t_{ij}\right) + \varepsilon_{ij} \quad j = 1,\ldots,m_i,$$

dans ce cas on utlise une procédure de moindres carrés ordinaire ou pénalisée et on estime les coefficients a_{ik} en minimisant un critère des moindres carrés

$$\sum_{j=1}^{n}\left[x_{ij} - \sum_{k=1}^{K} a_{ik} f_k(t_{ij})\right]^2,$$

et ce critère est mininisé par la solution :

$$\widehat{a_i} = \left(\Theta_i'\Theta_i\right)^{-1}\Theta_i'\tilde{X}_i$$

avec $\widehat{a_i} = (\widehat{a_{i1}},\ldots,\widehat{a_{iK}})'$, $\Theta_i = \left(f_k\left(t_{ij}\right)\right)_{1\le j\le m_i, 1\le k\le K}$ et $\tilde{X}_i = (X_{i1},\ldots,X_{im_i})'$.
Maintenant suivant la nature de nos données, les fonctions $f_k(t)$ quant à elles peuvent être une collection standard de fonctions de base comme les B-splines, les ondelettes(Ramsay et Silverman (2005)[21]) , la base de Fourier, ou encore la base de polynômes.
Dans le cas d'une base de Fourier par exemple, on obtient d'après la définition en (2.4.3) l'expansion :

$$\hat{X}_i(t) = a_{i0} + a_{i1}sin(\omega t) + a_{i2}cos(\omega t) + a_{i3}sin(2\omega t) + a_{i4}cos(2\omega t) + \cdots$$

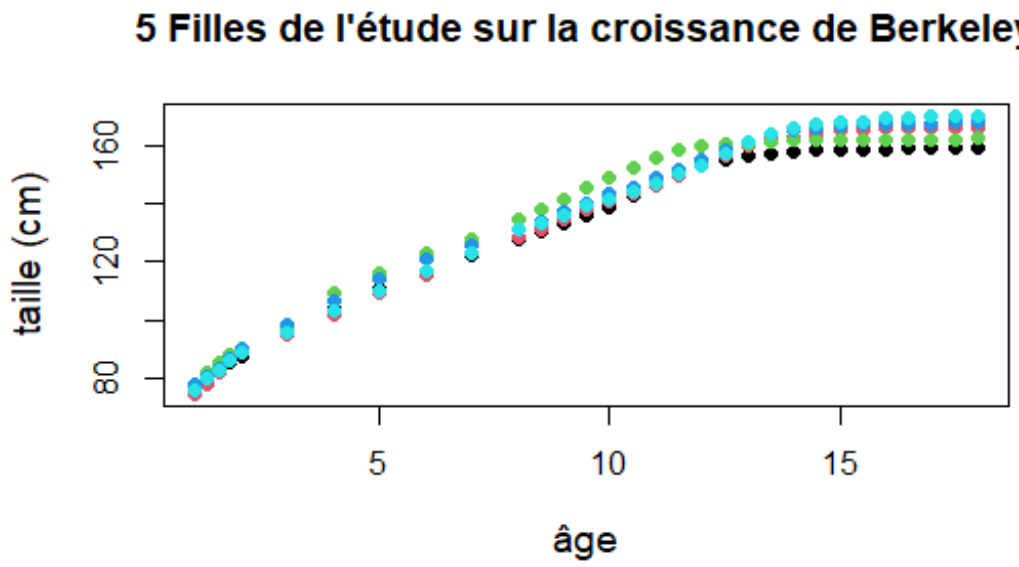
Voici un exemple court sur la base de données "Growth" du package "fda" , une liste contenant les tailles de 39 garçons et 54 filles âgés de 1 à 18 ans et les âges auxquels ils ont été recueillis. Nous nous intéressons par exemple à la vitesse et accélération des courbes de croissance des cinq premières filles .

Exemple 2.4.5. *Données brutes de l'étude sur la croissance de Berkeley*

```
matplot(x=growth$age, y=growth$hgtf[,1:5], type="p", lty=1, pch=21,
    xlab="âge", ylab="taille (cm)", cex.lab=1.2,col=1:5,bg=1:5,
main="5 Filles de l'étude sur la croissance de Berkeley")
```

Nous procédons dans la suite à un lissage des courbes de croissance à l'aide d'une base B-spline

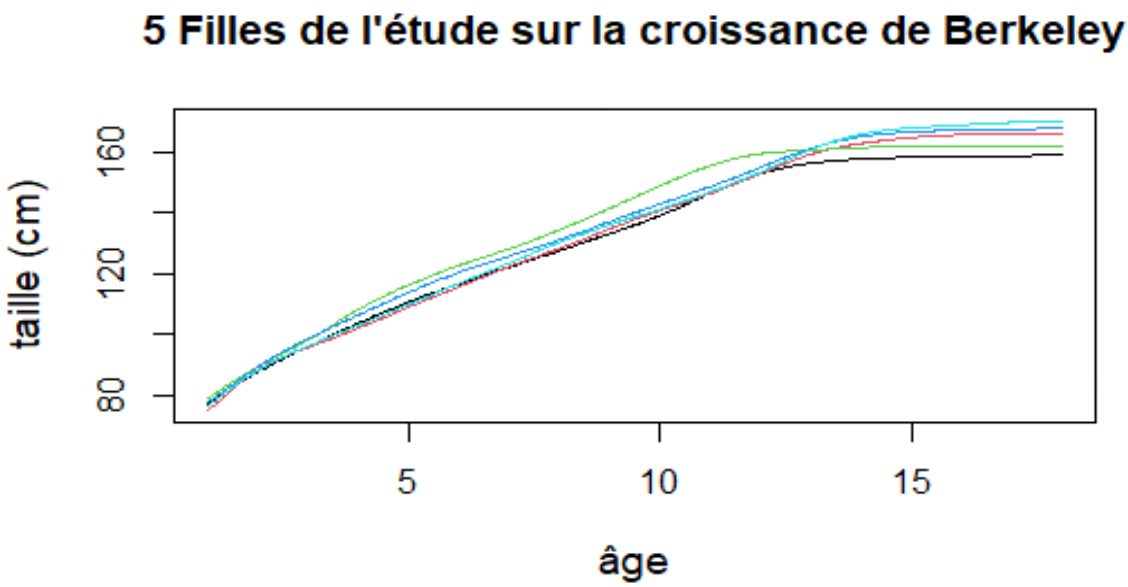
```
# observations rangées
rng <- c(1,18)
# base B-spline d'ordre 6 avec des nœuds =  âges
knots <- age
nordre <- 6
nbasis <- length(knots) + nordre - 2
hgtbasis <- create.bspline.basis(rng, nbasis, nordre, knots)
```

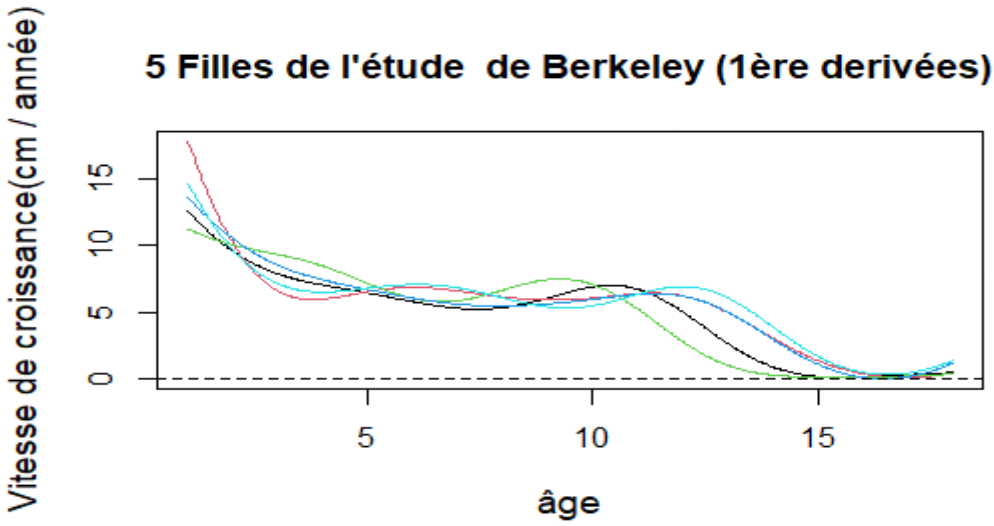


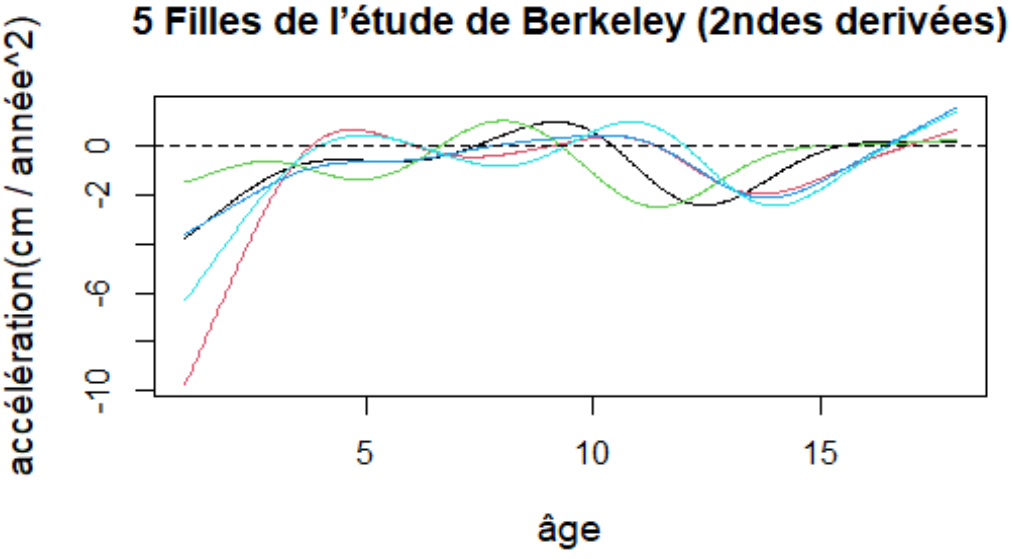

```
#  objet fonctionnel pour l'estimation des courbes
# La 4-èm derivée est pénalisée pour assurer
# une dérivée d'ordre 2 et une accélération lisses
Lfdobj <- 4
lambda <- 10^(-0.5)    # Valeur connue en avance.
growfdPar <- fdPar(hgtbasis, Lfdobj, lambda)
#  Données lissées: celles des filles correspondent à "hgtf".
hgtffd <- smooth.basis(age, growth$hgtf[,1:5], growfdPar)$fd
# Affichage des courbes des cinq premières filles
plot(hgtffd, xlab="âge", ylab="taille (cm)", cex.lab=1.2,
main="5 Filles de l'étude sur la croissance de Berkeley",lty=1)
```

La vitesse de croissance et l'accélération de ces courbes s'obtiennent ainsi comme suit :

```
# Vitesse
plot(deriv(hgtffd, 1), xlab="age",
ylab="Vitesse de croissance(cm / année)", cex.lab=1.2,
main="5 Filles de l'étude  de Berkeley (1ère derivées)",lty=1)

# Accélération
plot(deriv(hgtffd, 2), xlab="age", cex.lab=1.2,
ylab=" accélération-croissance (cm / année^2)",
main="5 Filles de l'étude  de Berkeley(2ndes derivées)",lty=1)
```


5 Filles de l'étude sur la croissance de Berkeley
taille (cm)
80
120
160
5
10
15
âge


5 Filles de l'étude de Berkeley (1ère derivées)
Vitesse de croissance(cm / année)
0
5
10
15
5
10
15
âge


5 Filles de l'étude de Berkeley (2ndes derivées)
accélération(cm / année^2)
-10
-6
-2
0
5
10
15
âge

2.5 Lissage Non paramétrique

Dans l'approche paramétrique de reconstruction des données X_i que nous avons présenté, la spécification d'une base de fonction appropriée était primordiale. Toutefois, il exite également une autre démarche nonparamétrique de lissage de données, ne nécessitant pas de spécification de base de fonctions et qui utilise principalement des méthodes de pondérations locales, comme les méthodes à noyau.

Principalement, il s'agit d'attribuer un poids w pour chaque temps t d'observation autour d'un point fixe t_0, qui soit de plus en plus grand au fur et à mesure que la distance entr t et t_0 devient petite, ces poids étant sous la forme

$$w(t) = \frac{1}{h}K(\frac{t-t_0}{h}),$$

où K est un noyau à valeurs positives et tel que $\int_{-\infty}^{+\infty} K(u)\mathrm{d}u = 1$, avec $h > 0$ le paramètre de lissage appelé "fenêtre". Ce faisant, on en déduit l'estimateur de Nadaraya-Watson (Nadaraya, 1964[19]) de X_i suivant la formule

$$\hat{X}_i(t) = \frac{\frac{1}{h}\sum_{j=1}^{m_i} X_{ij}K\left(\frac{t-t_j}{h}\right)}{\frac{1}{h}\sum_{j=1}^{m_i} K\left(\frac{t-t_j}{h}\right)}, \tag{2.2}$$

où t_j est le j-ièm temps d'observation de X_i. Cette procédure reste très sensible au choix de la fenêtre h qui se fait en général par validation croisée.

2.6 L'analyse en composantes principales pour données fonctionnelles(ACPF)

2.6.1 Un mot sur l'ACP Classique

Très souvent, nous désirons réduire la dimensionnalité d'énormes ensembles de données. C'est assez problématique, car chaque dimension stocke des informations que nous perdrions si nous les négligeons. Ceci dit, il faut une mesure d'information tout en réduisant la dimensionnalité, et dans le cas de l'ACP, cette mesure est la variation au sein de nos données car là où il y a variation, il y a aussi information.

L'objectif global de l'ACP consiste alors à trouver ce que l'on appelle les composantes principales (vecteurs), qui maximisent la variance des données le long de leur direction. Chaque composante principale explique une partie de la variance totale des données. Ils construisent une base orthonormée, ce qui signifie que chaque point de données est une combinaison linéaire de nos composantes principales. Ce faisant, il est alors possible de réduire la dimensionnalité de nos données en utilisant simplement un sous-ensemble de PC pour représenter nos points de données. Et ceci, en gardant par exemple le nombre minimale de composantes principales dont la variance cumulée dépasse 95%. De cette façon, nous réduisons la dimension des ensembles de données tout en préservant la plupart des informations.

2.6.2 Fonctionnement de l'ACP pour données fonctionnelles

A partir de l'ensemble des données fonctionnelles $X_1, \cdots, X_n$, on peut s'intéresser à la représentation optimale des courbes dans un espace fonctionnel de dimension réduite. Comme tâche principale, l'ACPF utilise des concepts similaires à ceux de l'ACP pour extraire les composantes principales. On calcule la matrice de covariance ou la matrice de corrélation entre les fonctions, puis on décompose cette matrice en valeurs propres et vecteurs propres, ce qui est possible grâce à la proposition 2.2.5. Les valeurs propres mesurent l'importance relative de chaque composante principale, tandis que les vecteurs propres représentent les poids attribués à chaque fonction dans chaque composante principale.
Formellement, nos données fonctionnelles sont sous la forme

$$X(t) = (X_1(t), \cdots, X_p(t), \quad t \in T), \quad \text{où T un intervalle quelconque de} \quad \mathbb{R} \quad et \quad X \in L^2(T).$$

Désignons par :

$$\mu(t) = \mathbb{E}(X(t)), \quad et \quad V(t,s) = Cov((X(t) - \mu(t))(X(s) - \mu(s))),$$

respectivement la fonction moyenne et l'opérateur de covariance de X et appelons $\lambda_1 \geq \lambda_2 \geq \cdots,$ les vecteurs propres obtenues par analyse spectrale de V(possible puisque l'opérateur de covariance étant compact), associée à une base orthonormée de fonctions propres $\left(C_j\right)_{j\in\mathbb{N}}$ appelées axes principaux fonctionnels et vérifiant :

$$\Gamma C_j = \lambda_j C_j, \tag{2.3}$$

$$\textbf{avec} \quad \int_T C_j(t) C_{j'}(t) dt = \delta_{jj'}$$

faisant de $\left(C_j\right)_{j\in\mathbb{N}}$ un système orthonormal de $L^2(T)$

Projection des fonctions sur les axes principaux : Les fonctions d'origine sont projetées sur les axes principaux sélectionnés pour obtenir les scores des individus dans l'espace des facteurs principaux. Cela permet de représenter chaque fonction par un ensemble de scores, réduisant ainsi la dimensionnalité des données et ce résultat constitue exactement ce qu'on désigne par décomposition de Karhunen-Loève(K. Karhunen., 1947[16], M. Loève., 1945[17]).

Définition 2.6.1. ***Décomposition de Karhunen-Loève***
On appelle développement de Karhunen-Loève de X, le développement de X dans la base de fonctions propres de l'opérateur de covariance associé de la forme :

$$X = \mathbb{E}[X] + \sum_{j=1}^{\infty} \langle X, C_j \rangle C_j = \mathbb{E}[X] + \sum_{j=1}^{\infty} R_j C j, \tag{2.4}$$

où

— C_j *est un vecteur propre de* Γ *associé à la valeur propre* $\lambda_j : \Gamma C_j = \lambda_j C_j$

— $\lambda_1 \geq \lambda_2 > \cdots \geq \cdots \geq 0.$

— *Les variables aléatoires* R_j *sont appelées composantes principales, projections ortho-*

gonales de X sur les fonctions propres C_j

$$R_j = \langle X, C_j \rangle = \int_T X(s)C_j(s)ds.$$

Ce sont des variables aléatoires centrées, décorellées et de variance égale à la valeur propre λ_j associée à la fonction propre C_j.

Troncature pour obtenir l'approximation : En tronquant (5.2) sur ses q premier termes par exemple, on obtient on obtient la meilleure approximation en norme L^2 de X par une somme de processus quasi-déterministes donnée par

$$X^q = \mathbb{E}[X] + \sum_{j=1}^{q} \langle X, C_j \rangle C_j = \mathbb{E}[X] + \sum_{j=1}^{q} R_j Cj, \tag{2.5}$$

2.6.3 Méthodes de calculs pour l'ACPF

Pour trouver des estimateurs de $\mu(t)$ et $V(s,t)$ pour $s,t \in T$, on part d'une réalisation $\{x_1,\ldots,x_n\}$ d'un échantillon $\{X_1,\ldots,X_n\}$, et on en déduit les estimateurs par les formules suivantes :

$$\hat{\mu}(t) = \frac{1}{n}\sum_{i=1}^{n} x_i(t) \quad \text{et} \quad \hat{V}(s,t) = \frac{1}{n-1}\sum_{i=1}^{n} \left(x_i(s) - \hat{\mu}(s)\right)\left(x_i(t) - \hat{\mu}(t)\right).$$

Et comme on l'a notifié prédemment, on observe nos données fonctionnelles qu'à des instants discrtes de temps et on procède à une reconstruction fonctionnelles de ces données en les décomposant dans une base de fonctions d'un espace de dimension finie.
Ainsi si $a_i = (a_{i1},\ldots,a_{iL})'$ désignent les coefficients dans la décompositions des courbes observées x_i dans la base $F = \{f_1,\ldots,f_L\}$, on aura que :

$$x_i(t) = F'(t)a_i, \quad \text{avec} \quad F(t) = (f_1(t),\ldots,f_L(t))'. \tag{2.6}$$

Appelons $\tilde{A}$ la matrice de taille $n \times L$ dont les lignes correspondent aux vecteurs a_i', et $M(t) = (x_1(t),\ldots,x_n(t))'$ le vecteurs des valeurs $x_i(t)$ de nos fonctions x_i au temps $t \in T\,(1 \leq i \leq n)$. On a que :

$$M(t) = \tilde{A}F(t). \tag{2.7}$$

Et en utilisant la décomposition donnée en (2.1), l'estimateur $\hat{V}$ de l'opérateur de covariance V va s'écrire :

$$\hat{V}(s,t) = \frac{1}{n-1}(M(s) - \hat{\mu}(s))'(M(t) - \hat{\mu}(t)) = \frac{1}{n-1}F(s)'A'AF(t), \tag{2.8}$$

où $A = (I_n - \mathbb{I}_n(1/n,\ldots,1/n))\tilde{A}$, avec I_n et $\mathbb{I}_n$ respectivement la matrice unité de taille $n \times n$ et le vecteur colonne unitaire de taillee n.

Maintenant, si nous utilisons le résultat (5.1) et (4.7), on parvient au fait que chaque C_j s'écrit :

$$C_j(t) = F(t)'b_j, \quad \text{où l'on a noté} \quad b_j = \left(b_{j1},\ldots,b_{jL}\right)'. \tag{2.9}$$

En utilisant la nouvelle expression de l'estimateur $\hat{V}$ de V, on remarque que l'équation aux valeurs propres (5.1) devient :

$$\int_0^T \hat{V}(s,t)C_j(t)dt = \lambda_j C_j(s),$$

ce qui revient en remplaçant $\hat{V}(s,t)$ et $C_j(t)$ par leurs expressions en (4.7) et (2.9) à écrire que :

$$\frac{1}{n-1}\Phi(s)'A'A\underbrace{\int_0^T \Phi(t)\Phi(t)'dt}_{W} b_j = \lambda_j \Phi(s)'b_j,$$

avec $W = \int_0^T \Phi(t)\Phi(t)'dt$ la matrice symétrique de taille $L \times L$ constituée des produits scalaires entre les fonctions de base. Comme on a ce résultat pour tout s, on en arrive à

$$\frac{1}{n-1}A'AWb_j = \lambda_j b_j.$$

Si nous posons $u_j = W^{1/2}b_j$, on effectue une analyse en composantes principales classisques de la matrice $\frac{1}{\sqrt{n-1}}AW^{1/2}$pour obetnir les composantes principales multivariées :

$$\frac{1}{n-1}W^{1/2'}A'AW^{1/2}u_j = \lambda_j u_j, \tag{2.10}$$

où les $b_j, j \geq 1$, associés aux fonctions propres f_j sont donnés par la formule $b_j = \left(W^{1/2}\right)^{-1} u_j$, et les scoores des composantes principales par :

$$R_j = AWb_j \quad j \geq 1, \tag{2.11}$$

qu'on peut également regarder comme solutions de l'équation aux valeurs propres

$$\frac{1}{n-1}AWA'\zeta_j = \lambda_j R_j.$$

2.7 Clustering pour données fonctionnelles

Le clustering des données fonctionnelles a fait l'objet d'une attention particulière de la part des statisticiens au cours de la dernière décennie. Nous présentons dans cette section une classification des différentes approches de clustering des données fonctionnelles en trois groupes : le clustering de données brutes, le clustering non paramétrique et les techniques de clutering basées sur des modèles, qui supposent une distribution de probabilité sous-jacente aux données.

2.7.1 Approche en deux étapes

Cette technique consiste premièrement en une étape de réduction de la dimension des données, où chaque courbe fonctionnelle est réduite en un nombre réduit de caractéristiques appelées descripteurs fonctionnels, et ensuite en une étape de clustering classique en dimension finie.
Dans la première étape essentiellement, on approxime les courbes de données en une base finie de fonctions (voir Section 1.5), par exemple la base de splines, choix le plus couramment utilisé en raison de leurs opitmalités (Voir par exemple Abraham et al (2003)[22], Rossi et al (2004)

[23] pour l'utilisation des B-splines). Une autre alternative à ce niveau consiste à passer par ACPF (Voir Section 1.7) afin de réduire la dimensionalité des données. Et une fois cela fait, on fait appel en deuxième position aux algorithmes de clustering usuels pour données fonctionnelles, désormais résumées soit par leurs coefficients dans une base de fonctions, soit par leurs premiers scores en composantes principales. Abraham et al (2003)[22] ont par exemple utilisé l'algorithme K-means sur les coefficients B-splines et Peng et al (2008)[24] sur un nombre de donné de scores des composantes principales .

Cette méthode a l'avantage de réduire effectivement la complexité des données, l'adaptation aux méthodes de clustering traditionnelles, mais se heurte aux problèmes de perte d'information en résumant les courbes en descripteurs, de dépendance de la qualité des descripteurs choisis.

2.7.2 Approches non paramétriques

Les approches non paramétriques fonctionnelles de clustering se divisent en deux, l'une utilisant les techniques de clustering non paramétriques classiques telles que les K-means, l'algorithme CAH et avec des distances spécifiques ou des dissimilarités, la seconde méthode proposant plutôt de nouveaux critères géométriques de clustering.
Concrètement, la première méthode utilise généralement une mesure de proximité définie entre deux courbes x_i et $x_{i'}$ par :

$$d_\ell(x_i, x_{i'}) = \left(\int_T \left(x_i^{(\ell)}(t) - x_{i'}^{(\ell)}(t) \right)^2 dt \right)^{1/2}, \tag{2.12}$$

où $x^{(\ell)}$ désigne la l-ième dérivée de x. On peut trouver par exemple dans Ferraty et al (2006)[25], une combianiason de l'algorithme CAH avec la distance d_0 (distance L^2), et dans Ieva et al (2012)[26] une utilisation couplée des K-means avec les distances d_0, d_1 et $(d_0 + d_1)^{\frac{1}{2}}$.
La seconde méthode utilise de nouvelles heuristiques pour le clustering fonctionnel. Hébrail et al (2010)[28] ont proposé par exemple une stratégie qui effectue simultannément du clustering et de l'estimation par morceaux des centres des clusters, et très récemment Yamamoto et al (2012)[27] ont développé une nouvelle procédure pour identifier simultanément les clusters optimaux de fonctions et les sous-espaces optimaux pour le clustering : à cette fin, une fonction objectif est définie comme la somme des distances entre les observations et leurs projections, plus les distances entre les projections et les moyennes de cluster (dans l'espace de projection) et un autre algorithme est utilisé pour optimiser la fonction objectif.

L'avantage d'utiliser cette méthode réside surtout en sa flexibilité pour modéliser des formes de courbes complexes (pas de contraintes sur les distributions). Elle est cependant sensibile aux paramètres de réglage, des difficultés de choisir les mesures de dissimilarité appropriées.

2.7.3 Méthodes basées sur les modèles

Les techniques de clustering fonctionnelles qui sont basées sur l'utilisation de modèles consistent principalement à supposer une densité de probabilité décrivant nos courbes. Ces techniques se distinguent de la méthode en deux étapes, où l'estimation des coefficients se fait préablement avant le clustering, alors qu'ils sont simultanément exécutés dans le cas que nous considérons actuellement.
Selon que la modélistion est faite sur les scores d'une ACPF (Voir Section 1.8) ou directement

sur les coefficients de l'expansion dans une base de fonctions de dimension finie (2.1), ces techniques de clustering se divisent en deux groupes.

2.7.3.1 Techniques de clustering utilisant la modélissation des composantes principales

Ces techniques s'appuient sur une idéé d'approximation de la densité de probabilité pour des variables fonctionnelles , proposée par Delaigle et al (2010) [34].
S'appuyant sur cette notion d'approximation d'une densité de probabilité pour une variable fonctionnelle, Bouveyron et al (2011) [35] et Jacques et al (2013) [36] ont fait l'hypothèse d'une distribution gaussienne sur les composantes principales C_j, ce qui leur a permis de dériver une technique de clustering utilisant la moyenne d'un modèle de mélange gaussien de la forme :

$$f_X^{(q)}(x;\theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{q_k} f_{C_j/Z_k=1}\left(c_{jk}(x);\lambda_{jk}\right), \tag{2.13}$$

où $\theta = \left(\pi_k, \lambda_{1k}, \ldots, \lambda_{q_k k}\right)_{1\leq k\leq K}$ sont les paramètres du modèle et q_k désigne l'odre de troncature dans le dévéloppement de Karhumen-Loève ([16], et spécifique à chaque cluster.
Le principal intérêt de cette approche réside dans le fait que les scores des composantes principales sont calculés par cluster (grâce à un algorithme de type EM (Voir Chapitre 2), qui calcule itérativement les probabilités conditionnelles d'appartenance des courbes à chaque cluster), effectue des ACPF par cluster en pondérant les courbes selon ces probabilités conditionnelles, et calcule les ordres de troncature q_k grâce au scree-test de Cattell [33].

2.7.3.2 Techniques de clustering utilisant la modélisation des coefficients d'expansion de base

On peut citer principalement l'algorithme nommé *fclust* de James et al (2013) [37], qui consiste à supposer que les coefficients de l'expansion des courbes en une base spline sont distribués selon un mélange de processus gaussiens de moyennes μ_k propre à chaque cluster, et variance commune Σ de la forme :

$$a_i \sim \mathcal{N}(\mu_k, \Sigma). \tag{2.14}$$

Une autre approche intéressante a également été proposée par Samé et al (2011) [38] en supposant que les courbes sont issues d'un mélange de regressions sur une base de fonctions polynomiales, avec de possibles changements de régime à chaque instant du temps. Ainsi, à chaque instant t_{ij}, l'observation X_{ij} est supposé provenir de l'un des modèles de régression polynomiale spécifique au cluster auquel appartient X_i.

2.7.4 Clustering de données fonctionnelles irrégulières à l'aide de GP

2.7.4.1 Rappels sur les Processus gaussiens (GP)

Dans cette partie de notre travail, nous nous attardons sur quelques rappels utiles sur les processus gaussiens, avec un bref aperçu sur la loi Gaussienne et sa généralisation en dimension d.

Définition 2.7.1. ***Loi Gaussienne multi-dimensitionnelle***
Une variable aléatoire X suit une loi Gaussienne multidimensionnelle de dimension d si sa loi de probabilité a pour densité $p(x) \in \mathbb{R}^d$ définie par :

$$p(x) := \mathcal{N}(x,\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} e^{-\frac{1}{2}\left((x-\mu),\Sigma^{-1}(x-\mu)\right)},$$
$$\text{avec } \mathbb{E}[X] = \mu \in \mathbb{R}^d \text{ et } \mathrm{Cov}[X] = \Sigma \in \mathcal{M}_{d,d}(\mathbb{R}).$$

Définition 2.7.2. ***Processus stochastique***
Un processus stochastique est une famille de variables aléatoires définies sur le même espace de probabilité, indexée par un ensemble T (on parle de processus discret dénombrable et continu sinon), et à valeurs dans un espace métrique M. On le note souvent par par $\{X_t\}_{t\in T}$. *En général on prend $T = \mathbb{N}$ ou $\mathbb{R}_+$.*

Définition 2.7.3. ***Processus Gaussien.***
Un processus stochastique $\{X_t\}_{t\in T}$ est dit Gaussien si chaque collection finie de variables aléatoires appartenant au processus suit une loi gaussienne multidimensionnelle. Cela revient à dire que $\{X_t\}_{t\in T}$ est gaussien si toute combinaison linéaire

$$a_1 X_{t_1} + \cdots + a_n X_{t_n}$$

suit une loi gaussienne (pour tout $n \in \mathbb{N}$, $\quad t_1,\ldots,t_n \in T$ et $a_1,\ldots,a_n \in \mathbb{R}$).

Ainsi lorqu'on parle de distribution d'un processus gaussien, on fait référence à la loi jointe de toutes ces variables aléatoires.
Par ailleurs, on sait que la loi d'un vecteur gaussien $(X_{t_1},\ldots,X_{t_n})$ est totalement déterminée si on connaît le vecteur moyenne

$$(\mathbb{E}[X_{t_1}],\ldots,\mathbb{E}[X_{t_n}])$$

et la matrice de covariance

$$\left(\mathrm{Cov}\left(X_{t_i},X_{t_j}\right)_{1\le i,j\le n}\right),$$

on en déduit alors que la loi d'un processus gaussien est déterminée complètement si on connaît sa fonction moyenne $\mu(t) = \mathbb{E}[X_t]$ et son opérateur de covariance $\Sigma(s,t) = \mathrm{Cov}(X_s,X_t)$. De plus, la loi finie-dimensionnelle de $(X_{t_1},\ldots,X_{t_n})$ est la loi normale de dimension n et de paramètres $\mathcal{N}(\mu_n,\Sigma_n)$ avec $\mu_n = (\mu(t_1),\ldots,\mu(t_n))$ et $\Sigma_n = \left(\Sigma\left(t_i,t_j\right)\right)_{1\le i,j\le n}$.

Le processus gaussien le plus connu est le **Mouvement Brownien** qui est obtenu avec $T = \mathbb{R}^+$, $\quad \mu(t) =$ et $K(s,t) = \min(s,t)$. On peut aussi citer le processus de d'Ornstein-Uhlenbeck qui est défini sur $T = \mathbb{R}_+$, avec pour paramètres $\mu(t) =$ et $K(s,t) = e^{-\frac{|s-t|}{2}}$.

2.7.4.2 Modélisation de données fonctionnelles à l'aide de processus gaussiens

Dans le cadre de notre travail, les données dont nous disposons sont le plus souvent affectées par un lot de difficulés qui sort souvent du cadre classique des données longitudinales, et forcent la création de modèles spécifiques à cette problématique. En effet, deux caractéristiques principales orientent les choix de modélisation dans notre étude :

➢ Une seule variable est observée , avec peu d'occurences par individus.
➢ Le nombre d'observations est différent d'un individu à l'autre et les instants d'observations sont différents.
Ces contraintes sont la raison principale du choix de considérer les observations comme fonctionnelles. Par ailleurs, la nécessité de quantifier l'incertitude des prédictions, notamment pour être utilisable en pratique, pousse à adopter une approche probabiliste, via la modélisation par processus Gaussiens, une approche récente qui permet de prendre en compte cette incertitude, proposée par exemple dans le livre de Rasmussen et Williams (2006) ou les articles de J. Q. Shi and Wang (2008) [13], Yang et al. (2017)[14].
C'est un cadre de travail dans lequel nos données fonctionnelles sont modélisées comme des réalisations d'un processus gaussien, processus défini par sa moyenne et sa covariance. La moyenne représente la tendance générale des données, tandis que la covariance mesure la dépendance entre les observations à différents points dans le temps ou l'espace.
Formellent, en notant $Y_i(t)$ la courbe de progression de l'individu i, on définit :

$$Y_i(t) = \mu(t) + X_i(t) + \varepsilon_i,$$

Avec :
— $\mu(t)$ une fonction moyenne commune à tous les individus,
— $X_i(t) \sim GP(0, \Sigma_i(\cdot,\cdot))$, un processus gaussien de moyenne nulle et de noyau de covariance $\Sigma_i(\cdot,\cdot)$,
— $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, un bruit blanc gaussien.
Ainsi, Y_i est également un processus Gaussien, de moyenne μ, calculé sur tous les individus, et de structure de covariance $\Sigma_i(\cdot,\cdot) + \sigma^2 Id$ estimée via les observations spécifiques à l'individu i.

2.7.4.3 Clustering de données fonctionneles à l'aide d'un mélange gaussien

L'idée générale de la méthode consiste à supposer le processus Y_i comme un mélange de processus Gaussiens $Y_i = \sum_{k=1}^{K} \pi_k GP_{ik}(\mu_k, \Sigma_{ik}(\cdot,\cdot))$, chaque processus ayant une moyenne et un noyau de covariance spécifique, chacun représentant un des K clusters de nos données. Dit autrement, on suppose que les données fonctionnelles d'un même groupe (cluster) proviennent d'une distribution de probabilité gaussienne spécifique à ce groupe. Et l'objectif est de trouver les paramètres optimaux de ces distributions gaussiennes pour chaque cluster, ainsi que les proportions optimales de chaque cluster π_k. De telles idées de mélange pour les processus Gaussiens ont été étudiées dans Rasmussen et Ghahramani (2002)[11] et plus récemment dans Bouveyron et al (2015)[9], Leroy et al (2022)[1] pour le clustering.

Cadre formel (Voir Leroy et al (2022) [1]

Le cadre formel de cette procédure s'articule comme suit : Si nous fixons le nombre de clusters à K, la taille de l'échantillon à n, et supposons que nous disposons de n_i observations sur chaque individu i, alors pour T un intervalle de $\mathbb{R}$, en notant :
➢ $t_i = (t_{i1}, \ldots, t_{in_i}) \in T^{n_i}$: le vecteur des instants d'observations pour l'individu i, et $t = (t_i)_{1 \leq i \leq n}$, le vecteur commun des temps d'observations,
➢ $y_{ij} = y_i(t_{ij}) \in \mathbb{R}, y_i = (y_{ij})_{1 \leq j \leq n_i}$, les vecteurs de sorties de l'individu i,
➢ $Z_i = (Z_{i1}, \ldots, Z_{iK})$ le vecteur des variables latentes pour chaque individu i , codant son appartenance à un cluster k donné, supposés distribués selon une distribution multinomiale commune : $Z_i \sim \mathcal{M}(1, \pi), 1 \leq i \leq n$, avec $\pi = (\pi_1, \cdots, \pi_K)^t$ et $\sum_{k=1}^{K} \pi_k = 1$,
alors l'approche, dans le cas univarié, par mélange gaussien consiste à modéliser la donnée Y_i d'un individu i appartenant à une classe k à l'instant t comme une somme d'un processus moyen spécifique au cluster k et d'un processus centré spécifique à l'individu i :

$$y_i(t) = \mu_k(t) + f_i(t) + \varepsilon_i(t), t \in \mathcal{T}. \tag{2.15}$$

avec pour $1 \leq i \leq n$, et $1 \leq k \leq K$,

— $\mu_k(\cdot) \sim \mathcal{GP}\left(m_k(\cdot), c_{\gamma_k}(\cdot,\cdot)\right)$: le processus moyen commun spécifique au cluster k,

— $f_i(\cdot) \sim \mathcal{GP}\left(0, \xi_{\theta_i}(\cdot,\cdot)\right)$: le processus spécifique à l'individu i,

— $\varepsilon_i(\cdot) \sim \mathcal{GP}(0, Id)$: le bruit associé au processsus de l'individu i,

— $\Theta = \left\{(m_k(\cdot))_{1 \leq k \leq K}, (\gamma_k)_{1 \leq k \leq K}, (\theta_i)_{1 \leq i \leq n}, \left(\sigma_i^2\right)_{1 \leq i \leq n}, \pi\right\}$: le vecteur des paramètres du modèle.

Hypothèses de la méthode
— $\mathbf{H}_1$: $\{\mu_k\}_{1 \leq k \leq K}$ sont indépendants,
— $\mathbf{H}_2$: $\{f_i\}_{1 \leq i \leq n}$ sont indépendants,
— $\mathbf{H}_3$: $\{Z_i\}_{1 \leq i \leq n}$ sont indépendants,
— $\mathbf{H}_4$: $\{\varepsilon_i\}_{1 \leq i \leq n}$ sont indépendants,
— $\mathbf{H}_5$: Pour tout $1 \leq i \leq n, \quad 1 \leq k \leq K, \quad \mu_k, \quad f_i, \quad Z_i$ sont indépendants.

2.7.4.4 Avantages et inconvenients

Les processus gaussiens (GP), adaptés pour résoudre certains problèmes de régression et de classification probabiliste, peuvent s'avérer très utiles et aider à founir de bonnes prédictions. Les avantages d'utiliser des processus gaussiens sont les suivants :
➢ La prédiction interpole les observations (au moins pour les noyaux réguliers).

➢ La prédiction est probabiliste (gaussienne), ce qui permet de calculer des intervalles de confiance empiriques et de décider, sur la base de ces intervalles, si l'on doit réajuster la prédic-

tion dans une région d'intérêt.

➢ Polyvalent : différents noyaux peuvent être spécifiés. Des noyaux communs sont fournis, mais il est également possible de spécifier des noyaux personnalisés.

Les quelques inconvénients des processus gaussiens généralement rencontrées sont entre autres :

➢ Ils ne sont pas éparses, c'est-à-dire qu'ils utilisent l'ensemble des échantillons/caractéristiques pour effectuer la prédiction.

➢ Ils perdent de leur efficacité dans les espaces de grande dimension, notamment lorsque le nombre d'entités dépasse quelques dizaines.

Dans ce chapitre, nous avons rappelé le cadre fonctionnel pour les données statistiques et passé en revue quelques techniques de clustering de données fonctionnelles. Le cadre qui nous intéresse reste le cas du clustering de courbes via des mélanges de processus gaussiens, qui requiert pour la pluspart du temps de nouvelles approches d'estimations autres que les méthodes classiques telles que les moindres carré ordinaires ou encore la méthode de maximum de vraisemblance.
Ainsi dans le chapitre suivant, nous discutons de nouveaux algorithmes d'estimations pour ces modèles de clustering, à savoir l'algorithme Espérance-Maximisation (EM) et ses variantes.

LES ALGORITHMES EM

3.1 Contexte

Dans cette partie, nous nous attardons un tout peu sur la théorie de l'estimation par maximum de vraisemblance et exposons sur l'algorithme d'estimation EM et ses variantes.

On part de n observations que l'on considère comme des réalisations de n variables aléatoires indépendantes et identiquement distribuées $(X_1, \cdots, X_n)$, de modèle statistique paramétrique sous-jacent $\left(E_X, \mathcal{E}, (f_\theta)_{\theta \in \Theta}\right)$, dominé par une mesure de référence μ : chaque $X_i, i = 1, \cdots, n$, suit une loi gouvernée par le vecteur de paramètres $\theta \in \mathbb{R}^d$ et admettant une densité de probabilité par rapport à la mesure de référence μ qui domine toutes les autres mesures possibles du modèle.

On note $p(x_i; \theta)$ ou $p(x_i \mid \theta)$, la densité de X_i par rapport à μ et on appelle vraisemblance de l'échantillon, la densité jointe des $X_1, \cdots, X_n$ par rapport à $\mu^{\otimes n}$ donnée par :

$$p_n(x_1, \cdots, x_n; \theta) = \prod_{i=1}^{n} p(x_i; \theta),$$

et notre objectif est de trouver l'estimateur par maximisation de la vraisemblance de θ donné par :

$$\widehat{\theta} \in \arg\max_{\theta} p_n(x_1, \cdots, x_n; \theta)$$

En gros pour un échantillon fixé, cet estimateur est en fait la valeur du vecteur de paramètres θ qui rend aussi vraisemblables que possible les observations que l'on dispose.
Mais très souvent, ce problème de maximisation est très complexe et pénible, soit parceque les seules données dont on dispose ne permettent pas l'estimation des paramètres, ou soit parce que l'expression de la vraisemblance est analytiquement impossible à maximiser. Et justement,l'algorithme EM constitue un outil puissant dans ce genre de situations. Dit de façon resumée, il vise à fournir un estimateur lorsque cette impossibilité provient de la présence de données cachées ou manquantes.

3.1.1 Espérance-Maximisation

L'idée générale consiste tout d'abord à supposer qu'une maximisation directe de la vraisemblance $p(x;\theta)$ est difficile, mais qui serait facile si l'on considère plutôt la vraisemblance des données complètes $p(x,z;\theta)$. Et définissant une distribution $q(z)$ sur le vecteur des variables latentes, on commence par décomposer la vraisemblance de la daçon suivante :

$$\begin{aligned}
\ln p(x;\theta) &= \int q(z) \ln p(x;\theta) dz \\
&= \int q(z) \ln \left(\frac{p(x;\theta) p(z \mid x;\theta)}{p(z \mid x;\theta)} \right) dz \\
&= \int q(z) \ln \left(\frac{p(x,z;\theta)}{p(z \mid x;\theta)} \right) dz \\
&= \int q(z) \ln \left(\frac{p(x,z;\theta) q(z)}{p(z \mid x;\theta) q(z)} \right) dz \\
&= \int q(z) \ln \left(\frac{p(x,z;\theta)}{q(z)} \right) dz - \int q(z) \ln \left(\frac{p(z \mid x;\theta)}{q(z)} \right) dz \\
&= \mathcal{L}(q,\theta) + KL(q \| p).
\end{aligned} \tag{3.1}$$

où $KL(\cdot\|\cdot)$ est la divergence Kullback-Leibler entre q(z) et la loi a posteriori q(z|x,θ), qui est une quantité positive(par Jensen appliquée à la concavité de la fonction x$\rightarrow \log(x)$), et $\mathcal{L}(q,\theta)$ est une fonctionnelle de q(z) et de θ, connue sous le nom de borne inférieure évidente(evidence lower bound ou ELBO en anglais) sur la vraisemblance. Il s'en suit donc

$$\text{que :} \quad \ln p(x;\theta) \geq \mathcal{L}(q,\theta) \tag{3.2}$$

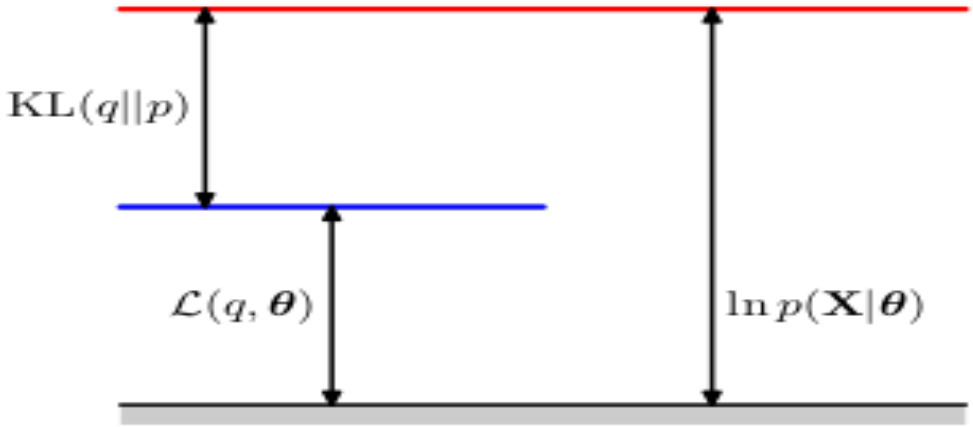


FIGURE 3.1 – Confert Christopher M. Bishop., 2006[5], illustration de la décomposition de la vraisemblance qui vaut pour tout choix de distribution q(z). Parce que le La divergence de Kullback-Leibler satisfait $KL(q\|p) \geq 0$, on voit que la quantité $\mathcal{L}$(q,θ) est une borne inférieure sur le logarithme de la vraisemblance p(x ;θ).

Ainsi, au lieu de maximiser directement la méthode EM s'intéresse plutôt à la maximisation de la cette borne inférieure établie sur la vraisemblance.

3.1.2 Démarche de la méthode

Supposons que la valeur actuelle du vecteur paramètre soit θ^{Anc}. À l'étape E, le la borne inférieure $\mathcal{L}(q,\theta)$ est maximisée par rapport à $q(z)$ tout en maintenant θ^{Anc} fixe. On trouve la solution à ce problème de maximisation en remarquant que la valeur de $\log p(x|\theta^{Anc})$ (où θ^{Anc} fait allusion au vecteur des anciens paramètres) ne dépend pas de $q(z)$ et donc que la plus grande valeur de $\mathcal{L}(q,\theta)$ sera atteinte lorsque la divergence de Kullback-Leibler s'annulera, c'est-à-dire lorsque $q(z)$ sera égalé à la distribution postérieure $p(z|x,\theta^{Anc})$.

A l'étape M pour $q(z)$ fixée, on maximise la borne inférieure $\mathcal{L}(q,\theta)$ par rapport à θ pour en déduire une nouvelle valeur θ^{Nouv}. En supposant par exemple une expression analytique de $p(z|x,\theta^{Anc})$ obtenue à l'étape E précédente, on obtient en substituant simplement $q(z) = p(z|x,\theta^{Anc})$ dans la borne inférieure ci-dessus :

$$\begin{aligned}
\mathcal{L}(q,\theta) &= \int q(z)\ln\left(\frac{p(x,z;\theta)}{q(z)}\right)dz \\
&= \int q(z)\ln p(x,z;\theta)dz - \int q(z)\ln q(z)dz \\
&= \int p\left(z \mid x;\theta^{Anc}\right)\ln p(x,z;\theta)dz \\
&\quad - \int p\left(z \mid x;\theta^{Anc}\right)\ln p\left(z \mid x;\theta^{Anc}\right)dz \\
&= Q\left(\theta,\theta^{Anc}\right) + H(q).
\end{aligned}$$

Le second terme $H(q)$ est simplement l'entropie négative de q, une constante par rapport à θ et n'est donc pas prise en compte dans le processus de maximisation de la borne inférieure à l'étape M.

L'algorithme EM se résume donc essentiellement en une maximisation de $Q\left(\theta,\theta^{Anc}\right)$ de la façon suivante :

— Étape E : Calculer $p(z|x,\theta^{Anc})$ et $\int p\left(z \mid x;\theta^{Anc}\right)\ln p(x,z;\theta)dz$

— Étape M : Trouver un nouveau vecteur des paramètres θ^{Nouv} vérifiant :

$$\theta^{Nouv} = \text{argmax}_\theta \int p\left(z \mid x;\theta^{Anc}\right)\ln p(x,z;\theta)dz,$$

ce qui entraîne une augmentation de la borne inférieure, et par suite nécessairement une augmentation de la fonction de log de vraisemblance correspondante.

3.1.2.1 Garanties et faiblesses de la méthode

L'utilisation de la méthode EM nous garanti une convergence vers un maximum local de la fonction de vraisemblance(Dempster et al., 1977)[9]), même si ce maximum n'est pas global. L'algorithme EM fonctionne généralement assez efficacement et converge rapidement(J.Q. Shi

· B. Wang, 2008[13]. On trouve par exemple une utilisation efficace de cette méthode dans Arthur L. et al., 2022[10], dans le cadre d'un modèle de prédiction à l'aide de GP multitâches qui souligne la convergence de la méthode la plupart du temps après quelques itérations vers des maximums locaux.
Cependant, La dépendance en la condition initiale θ_0 choisie arbitrairement est forte : pour certaines mauvaises valeurs, l'algorithme peut rester gelé en un point selle, alors qu'il convergera vers le maximum global pour d'autres valeurs initiales plus pertinentes.

L'astuce souvent utilisée consiste à choisir un ensemble de valeurs de départ différentes (en fonction de la complexité du problème), et comparerles valeurs des log-vraisemblance et valeurs des estimations pour faire un choix raisonnable de valeurs de départ. Cette méthode conduit généralement à des résultats très robustes.

3.1.3 EM Variationnel

Nous avons vu précédemment avec l'algorithme EM, que nous avions besoin évaluer l'espérance de la vraisemblance des données complètes par rapport à la distribution a posteriori des variables latentes. Mais dans plusieurs cas, on n'est pas en mesure d'évaluer cette distribution a priori(dû par exemple à la dimension du vecteur des variables latentes, ou parce que la distribution postérieure a une forme très complexe pour laquelle des calculs d'espérances ne sont pas analytiquement traitables).
L' EM variationnel consiste justement à dériver une approximation optimale de cette distribution afin de pouvoir appliquer la théorie qu'on connait déjà sur la méthode. Une démarche habituelle consiste à se restreindre à une famille de distributions puis de chercher l'élément de cette famille qui minimise la divergence de Kullback, et l'objectif c'est de pouvoir le faire assez suffisamment pour qu'elle ne comporte que des distributions traitables, mais tout en permettant à la famille d'être suffisamment riche pour pouvoir fournir une bonne approximation de la vraie distribution a posteriori.
Dans le cas où par exemple $q(z) = \prod_i q(z_i)$, la borne inférieure se factorise en :

$$
\begin{aligned}
\mathcal{L}(q,\theta) &= \int q(z) \ln\left(\frac{p(x,z;\theta)}{q(z)}\right) dz \\
&= \int \prod_i q(z_i) \ln p(x,z;\theta) dz - \sum_i \int q(z_i) \ln q(z_i)\, dz_i \\
&= \int q(z_j) \int \left(\prod_{i\neq j} q(z_i) \ln p(x,z;\theta)\right) \prod_{i\neq j} dz_i dz_j \\
&\quad - \int q(z_j) \ln q(z_j)\, dz_j - \sum_{i\neq j} \int q(z_i) \ln q(z_i)\, dz_i \\
&= \int q(z_j) \ln\left(\frac{\exp\left(\langle \ln p(x,z;\theta)\rangle_{i\neq j}\right)}{q(z_j)}\right) dz_j \\
&\quad - \sum_{i\neq j} \int q(z_i) \ln q(z_i)\, dz_i \\
&= \int q(z_j) \ln\left(\frac{\tilde{p}_{i\neq j}}{q(z_j)}\right) dz_j + H\left(z_{i\neq j}\right) + c \\
&= -KL\left(q_j \| \tilde{p}_{i\neq j}\right) + H\left(z_{i\neq j}\right) + c,
\end{aligned}
$$

,

où l'on a définit une nouvelle distribution $\widetilde{p}\left(\mathbf{x},\mathbf{z}_j\right)$ par la relation

$$\ln \widetilde{p}\left(\mathbf{x},\mathbf{z}_j\right) = \langle \ln p(x,z;\theta)\rangle_{i\neq j} + C = \mathbb{E}_{i\neq j}[\ln p(\mathbf{x},\mathbf{z};\theta)] + c,$$

avec $\langle . \rangle_{i\neq j} = \mathbb{E}_{i\neq j}[\cdots]$ l'espérance par rapport a la distribution de q sous toutes les variables $\mathbf{z}_i$ pour $i \neq j$, de sorte que

$$\mathbb{E}_{i\neq j}[\ln p(\mathbf{x},\mathbf{z};\theta)] = \int \ln p(\mathbf{x},\mathbf{z};\theta) \prod_{i\neq j} q_i \, \mathrm{d}\mathbf{z}_i.$$

Puisque $\exp\left(\langle \ln p(x,z;\theta)\rangle_{i\neq j}\right)$ n'est pas une densité de probabilité proprement dite, la constante c est ajoutée pour l'ajuster afin qu'elle devienne une densité de probabilité appropriée. Comme la divergence KL n'est pas négative, la borne inférieure est maximisé lorsque $KL(\cdot\|\cdot) = 0$, ce qui se produit lorsque

$$q\left(z_j\right) = \tilde{p}_{i\neq j} = \exp\langle \ln p(x,z;\theta)\rangle_{i\neq j}. \tag{3.3}$$

Cette équation dit donc que le logarithme de la solution optimale pour le facteur q_j est obtenu simplement en considérant le logarithme de la distribution conjointe sur toutes les variables cachées et visibles, puis en prenant l'espérance par rapport à tous les autres facteurs.

Ainsi de façon similaire, pour l'EM variationnelle on a les étapes suivantes :

- Étape E : Calculer $q^*\left(z_j\right) = \exp\langle \ln p(x,z;\theta)\rangle_{i\neq j} \quad \forall j, \quad$ et poser $\quad q^{Nouv} = \prod_i q_i^*$
- Étape M : Trouver $\theta = \mathrm{argmax}_\theta \, \mathcal{L}\left(q^{Nouv},\theta\right)$

3.1.3.1 Garanties et faiblesses de la méthode

De même pour cette méthode on a la garantie d'une convergence vers un minimum local (Boyd et Vandenberghe, 2004[2]) de la fonction d'approximation, garantissant ainsi un résultat stable et elle est souvent plus rapide que les méthodes d'inférence exactes, telles que l'échantillonnage de Gibbs ou la méthode de Monte-Carlo par chaînes de Markov. Elle a été par exemple utilisée dans Arthur Leroy et al.[1] et dans Hensman et al., 2013a [7] et a produit de bonnes performances d'estimation. De même Titsias, 2009[6] l'a utilisé pour introduire une formultion variationnelle d'approximation permettant d'estimer conjointement les hyperparamètres du noyau et les entrées d'un modèle GP parcimonieux en maximisant une borne inférieure du logarithme de la varie vraisemblance, ce qui s'est révélé bien meilleure que les méthodes précédemment utilisées par exemple dans Seeger et al., 2003.[8].
Par contre elle n'offre pas de garanties sur la qualité de l'approximation de la distribution postérieure, même si elle est souvent précise en pratique et peut même introduire un biais dans l'estimation des paramètres du modèle, sensible aux valeurs initiales des paramètres, ce qui peut conduire à des résultats différents en fonction des points de départ.

3.1.4 EM stochastique

C'est une variante de l'EM utilisée lorsqu'on a des données massives ou pour des modèles avec un grand nombre de paramètres. Cette méthode propose d'intercaler une étape stochastique de classification entre les étapes E et M afin de réduire le risque de tomber dans un maximum local de vraisemblance. Concrètement, après avoir calculé les probabilités $t_{ik} = \frac{\pi_k f(x_i, \theta_k)}{\sum_{\ell=1}^{g} \pi_\ell f(x_i, \theta_\ell)}$, il s'agit de tirer l'appartenance z_{ik} d'un individu i à une classe k donnée selon une loi multinomiale $\mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \boldsymbol{\tau}_{iK})^\top\right)$.

3.1.4.1 Garanties et faiblesses de la méthode

Cette méthode est bien adaptée aux problèmes de grande taille, car elle permet d'économiser en temps de calcul et de la mémoire en ne travaillant qu'avec des mini-lots de données à chaque itération et le plus important c'est qu'elle peut être plus robuste que la méthode EM classique, car elle évite certains problèmes de convergence liés à la sensibilité des estimations initiales et des valeurs aberrantes.

Dans la suite, nous exposons sur un exemple d'utilisation de la stratégie EM variationnel dans Arthur Leroy et al.[1].

EMV POUR UN MODÈLE DE CLUSTERING AVEC GP MULTITÂCHES

4.1 Contexte

Tout au long de notre travail, nos variables d'entrées se référerons à des horodatages et le terme individu désignera une tâche. L'ensemble de tous les indices est désigné par $I \in \mathbb{N}$, contenant notamment $\{1,\cdots,M\}$, les indices des individus observés (qui constitueront notre échantillon d'apprentissage). Comme les variables d'entrées sont continues, nous notons également T, l'espace où ces entrées prennent leurs valeurs(avec par exemple $T \subset \mathbb{R}$). Maintenant, puisque nous sommes dans un cadre de classification, on notera l'ensemble des indices des K-différents clusters par $\{1,\cdots,K\}$, et pour simplifier on adopte la notation $\{x_i\}_i = \{x_1,\ldots,x_M\}$ et également $\{x_k\}_k = \{x_1,\ldots,x_K\}$
Nous sommes dans le cadre où nos données nous proviennent de M différents individus et pour chaque individu, nous disposons de N_i données $\left\{\left(t_i^1, y_i\left(t_i^1\right)\right),\ldots,\left(t_i^{N_i}, y_i\left(t_i^{N_i}\right)\right)\right\}$.
Conventions de notations : on notera par

— $\mathbf{t}_i = \left\{t_i^1,\ldots,t_i^{N_i}\right\}$, l'ensemble des horodatage d'un individu i

— $\mathbf{y}_i = y_i(\mathbf{t}_i)$, le vecteur des sorties d'un individu i,

— $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, la grille commune des horodatages de tous les individus de notre échantillon,

— $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, le nombre totale des horodatages en lesquels nous disposons d'observations.

Bien-sûr les valeurs d'entrée peuvent varier à la fois en nombre et en emplacement d'un individu à un autre et dans le cas où par exemple $\mathbf{t}_i = \mathbf{t}, \forall i \in \mathcal{I}$, nous dirons que la grille d'horodatages est commune, dans le cas contraire elle sera dite non commune.
Dans le cadre d'un mélange gaussien , on introduit un vecteur aléatoire binaire latent pour chaque individu i noté $Z_i = (Z_{i1},\ldots,Z_{iK})^\top$ qui permet d'indiquer la classe dans laquelle se trouve l'individu i de l'échantillon. Dans le cadre de notre travail, les variables latentes seront supposées multinomiales : $Z_i \sim \mathcal{M}(1,\pi), \forall i \in \mathcal{I}$, où π est le vecteur des proportions $(\pi_1,\ldots,\pi_K)^\top$, avec la condition supplémentaire $\sum_{k=1}^{K} \pi_k = 1$.

4.2 Le modèle

Pour un individu i appartenant à une classe k donné, son expression fonctionnelle s'écrit :

$$y_i(t) = \mu_k(t) + f_i(t) + \varepsilon_i(t), \forall t \in T$$

où : $\mu_k(\cdot) \sim \mathcal{GP}\left(m_k(\cdot), C_{\gamma_k}(\cdot,\cdot)\right)$ désigne le processus moyen commun de la classe k, $f_i(\cdot) \sim \mathcal{GP}\left(0, \xi_{\theta_i}(\cdot,\cdot)\right)$ est le processus spécifique à l'individu i et $\varepsilon_i(\cdot) \sim \mathcal{GP}\left(0, \sigma_i^2 I\right)$ désigne le terme d'erreur. Les hyper-paramètres du modèle dans ce cadre étant :

➢ $\forall k = 1, \cdots, K, m_k(\cdot)$ est la fonction a priori sur la moyenne de la classe k,
➢ $\forall k = 1, \cdots, K, C_{\gamma_k}(\cdot,\cdot)$ le noyau de covariance associé à la classe k et paramétré par γ_k,
➢ $\forall i \in I, \xi_{\theta_i}(\cdot,\cdot)$ est le noyau associé à l'individu i, de paramètre paramétré par θ_i,
➢ $\forall i \in I, \sigma_i^2 \in \mathbb{R}$ est le bruit associé à l'individu i,
➢ $\forall i \in I$, on pose $\psi_{\theta_i,\sigma_i^2}(\cdot,\cdot) = \xi_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I$,
➢ $\Theta = \left\{\{\gamma_k\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i, \boldsymbol{\pi}\right\}$, est le vecteur des paramètres du mélange global.

De plus , on émet les hypothèses suivantes :

— Les $\{\mu_k\}_k$ sont indépendants et les $\{f_i\}_i$ sont indépendants

— Les $\{\mathbf{Z}_i\}_i$ sont indépendants

— Les $\{\varepsilon_i\}_i$ sont indépendants

— $\forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \mu_k, f_i, \mathbf{Z}_i, \varepsilon_i$ ainsi que toute combinaison deux à deux sont indépendants.

Et maintenant on obtient la distribution a priori conditionnelle des $y_i()$ en intégrant sur les f_i qui donne la formulation du type :

$$y_i(\cdot) \mid \{Z_{ik} = 1, \mu_k(\cdot)\} \sim \mathcal{GP}\left(\mu_k(\cdot), \psi_{\theta_i,\sigma_i^2}(\cdot,\cdot)\right), \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{4.1}$$

Ce qui voudrait dire que, les processus de sorties $\{y_i(\cdot) \mid \{\mathbf{Z}_i\}_i, \{\mu_k(\cdot)\}_k\}_i$ sont également indépendants, conditionnellement aux variables latentes.
D'autre part la vraisemblance du modèle s'écrit, si par exemple on a un ensemble fini d'observations $\{\mathbf{t}_i, \mathbf{y}_i\}_i$:

$$\begin{aligned} p\left(\{\mathbf{y}_i\}_i \mid \{\mathbf{Z}_i\}_i, \{\mu_k(\mathbf{t})\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i\right) &= \prod_{i=1}^{M} p\left(\mathbf{y}_i \mid \mathbf{Z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i, \sigma_i\right) \\ &= \prod_{i=1}^{M}\prod_{k=1}^{K} p\left(\mathbf{y}_i \mid Z_{ik} = 1, \mu_k(\mathbf{t}_i), \theta_i, \sigma_i\right)^{Z_{ik}} \\ &= \prod_{i=1}^{M}\prod_{k=1}^{K} \mathcal{N}\left(\mathbf{y}_i; \mu_k(\mathbf{t}_i), \Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)^{Z_{ik}}, \end{aligned} \tag{4.2}$$

où Pour tout i $\in \mathcal{I}, \Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2} = \psi_{\theta_i,\sigma_i^2}(\mathbf{t}_i, \mathbf{t}_i) = \left[\psi_{\theta_i,\sigma_i^2}(u,v)\right]_{u,v\in\mathbf{t}_i}$ désigne une matrice de variance-covariance de dimension $N_i \times N_i$.
Les distributions a priori des processus moyens pour chaque classe évaluées sur la grille groupée

des horodatatges s'écrit également :

$$\begin{aligned} p\left(\{\mu_k(\mathbf{t})\}_k \mid \{\gamma_k\}_k\right) &= \prod_{k=1}^{K} p\left(\mu_k(\mathbf{t}) \mid \gamma_k\right) \\ &= \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right), \end{aligned} \tag{4.3}$$

où $\mathbf{C}_{\gamma_k}^{\mathbf{t}} = c_{\gamma_k}(\mathbf{t},\mathbf{t}) = \left[c_{\gamma_k}(k,\ell)\right]_{k,l\in\mathbf{t}}$ est une matrice de variance-covariance de dimension $N \times N$, tout comme celles des variables latentes d'affectation qui elles se factorisent sur les individus de la manière :

$$\begin{aligned} p\left(\{\mathbf{Z}_i\}_i \mid \boldsymbol{\pi}\right) &= \prod_{i=1}^{M} p\left(\mathbf{Z}_i \mid \boldsymbol{\pi}\right) \\ &= \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\pi}\right) \\ &= \prod_{i=1}^{M}\prod_{k=1}^{K} \pi_k^{Z_{ik}}. \end{aligned} \tag{4.4}$$

De tout ce qui précède, on peut reprendre la vraisemblance des données complètes du modèle de la façon suivante :

$$\begin{aligned} p\left(\{\mathbf{y}_i\}_i, \{\mathbf{Z}_i\}_i, \{\mu_k(\mathbf{t})\}_k \mid \Theta\right) &= p\left(\{\mu_k(\mathbf{t})\}_k \mid \gamma_k\right) \prod_{i=1}^{M} p\left(\mathbf{y}_i \mid \mathbf{Z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i, \sigma_i^2\right) p\left(\mathbf{Z}_i \mid \boldsymbol{\pi}\right) \\ &= \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right) \prod_{i=1}^{M} \left(\pi_k \mathcal{N}\left(\mathbf{y}_i; \mu_k\left(\mathbf{t}_i\right), \Psi_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right)\right)^{Z_{ik}}. \end{aligned} \tag{4.5}$$

Ici, comme on peut le voir, elle dépend de variables latentes qui ne peuvent pas être évaluées directement. Aussi, même si les lois a priori sur $\{\mathbf{Z}_i\}_i$ et $\{\mu_k(\mathbf{t})\}_k$ sont indépendantes, les expressions de leurs a posteriori respectifs dépendraient inévitablement l'une de l'autre. Ce qui nous reste à faire, c'est d'utiliser la méthode EM varaiationel exposée précédemment pour faire cette maximisation et trouver les paramètres optimaux du modèle.

On va donc procéder à une maximisation de la borne inférieure $\mathcal{L}(q,\theta)$, qui dépend de la distribution approximative q(.) et de θ, pour obtenir de paramètres optimaux pour notre modèle. Et comme précédemment, on suppose la factorisation suivante pour la distribution approximative :

$$q(\mathbf{Z},\boldsymbol{\mu}) = q_{\mathbf{Z}}(\mathbf{Z}) q_{\boldsymbol{\mu}}(\boldsymbol{\mu}) \tag{4.6}$$

On va donc dire que la propriété d'indépendance qui manquait pour calculer explicitement les distributions hyperpostérieures est imposée. Une telle condition restreint la famille de distribution parmi lesquelles nous choisissons q(·), et nous cherchons maintenant des approximations au sein de cette famille qui sont aussi proches que possible des vraies distributions hyper-postérieurs.

☞**Etape E** : Il s'agit donc de maximiser $\mathcal{L}(q,\theta)$ par rapport à q(.), avec un vecteur θ probablement estimé ultérieurement ou alors initialisé. Ce faisant, on pourra endéduire avec l'hypothèse faite en (4.6), des expressions analytiques des disbributions postérieures optimales pour $q_{\mathbf{Z}}(\mathbf{Z})$ et $q_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. Mais ici dans notre cas, le calcul de chaque distribution implique de prendre une espérance par rapport à l'autre, cela suggère donc une procédure itérative, ce que détaillent les propositions suivantes.

Proposition 4.2.1. *Supposons que* $\widehat{\Theta}$ *et la distribution variationelle* $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right)$ *sont connus. L'approximation variationelle optimale* $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$ *de la vraie distribution hyper-postérieure* $p\left(\mathbf{Z} \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}\right)$ *se factorise en un produit de distributions multinomiales :*

$$\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \boldsymbol{\tau}_{iN})^\top\right)$$

où :

$$\tau_{ik} = \frac{\widehat{\pi}_k \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}\right) \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}{}^{-1} \widehat{\mathbf{C}}_k^{\mathbf{t}_i}\right)\right)}{\sum_{l=1}^{K} \widehat{\pi}_l \mathcal{N}\left(\mathbf{y}_i; \widehat{m}_l(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}\right) \exp\left(-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}{}^{-1} \widehat{\mathbf{C}}_k^{\mathbf{t}_i}\right)\right)}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}.$$

Preuve. Nous utiliserons les résultats suivants dans la preuve :

Lemme 4.2.2. *Soit* $X \in \mathbb{R}^N$ *un vecteur gaussien* $X \sim \mathcal{N}(m, \mathbf{K})$*, où* $\mathbb{E}_X$ *désigne l'espérance,* $\mathbb{V}_X$ *la variance par rapport à la loi de X. Et considérons* $b \in \mathbb{R}^N$ *un vecteur arbitraire et* $\mathbf{S}$ *une matrice de variance-covariance de dimension* $N \times N$*, alors on a :*

$$\mathbb{E}_X\left[(X-b)^\top \mathbf{S}^{-1}(X-b)\right] = (m-b)^\top \mathbf{S}^{-1}(m-b) + \operatorname{tr}\left(\mathbf{K}\mathbf{S}^{-1}\right)$$

Preuve. On a en effet :

$$\begin{aligned}
\mathbb{E}_X\left[(X-b)^\top \mathbf{S}^{-1}(X-b)\right] &= \mathbb{E}_X\left[\operatorname{tr}\left(\mathbf{S}^{-1}(X-b)(X-b)^\top\right)\right] \\
&= \operatorname{tr}\left(\mathbf{S}^{-1}(m-b)(m-b)^\top\right) + \operatorname{tr}\left(\mathbf{S}^{-1}\mathbb{V}_X[X]\right) \\
&= (m-b)^\top \mathbf{S}^{-1}(m-b) + \operatorname{tr}\left(\mathbf{K}\mathbf{S}^{-1}\right).
\end{aligned}$$

■

Rappel :
La densité de probabilité d'un échantillon gaussien multi-dimensionnel de vecteur moyenne μ et de matrice de covariance Σ est donnée par l'expression suivante :

$$f(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right) \tag{4.7}$$

En se rappelant de la décomposition en (4.5) de la distribution jointe de toutes les variables donnée par :

$$p\left(\{\mathbf{y}_i\}_i, \{\mathbf{Z}_i\}_i, \{\mu_k(\mathbf{t})\}_k \mid \Theta\right) = p\left(\{\mu_k(\mathbf{t})\}_k \mid \gamma_k\right) \prod_{i=1}^{M} p\left(\mathbf{y}_i \mid \mathbf{Z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i, \sigma_i^2\right) p\left(\mathbf{Z}_i \mid \boldsymbol{\pi}\right),$$

alors le log de l'approximation optimale de q(**Z**) d'après (3.3) et en tenant compte du fait que que nous ne nous intéressons qu'à la dépendance fonctionnelle du membre de droite vis-à-vis de la variable Z et qu'ainsi tout terme qui ne dépend pas de Z peut être absorbé dans la constante de normalisation additive, donnant :

$$\begin{aligned}
\log \widehat{q}_{\mathbf{Z}}(\mathbf{Z}) =& \mathbb{E}_{\boldsymbol{\mu}}\left[\log p\left(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \widehat{\Theta}\right)\right] + C_1 \\
=& \mathbb{E}_{\boldsymbol{\mu}}\left[\log p\left(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i\right) + \log p(\mathbf{Z} \mid \widehat{\boldsymbol{\pi}}) + \log p\left(\boldsymbol{\mu} \mid \{\widehat{\gamma}_k\}_k\right)\right] + C_1, \\
& \text{où l'on a utilisé la décomposition (4.5) et en absorbant les termes indépendants de Z} \\
& \text{dans la consatnte } C_1\text{, on a :} \\
=& \mathbb{E}_{\boldsymbol{\mu}}\left[\log p\left(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i\right)\right] + \log p(\mathbf{Z} \mid \widehat{\boldsymbol{\pi}}) + C_2 \\
=& \mathbb{E}_{\boldsymbol{\mu}}\left[\sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik} \log p\left(\mathbf{y}_i \mid Z_{ik}=1, \mu_k(\mathbf{t}_i), \widehat{\theta}_i, \widehat{\sigma}_i^2\right)\right] + \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\log\left(\widehat{\pi}_k\right) + C_2, \\
& \text{cette fois en utilisant les formules (5.2) et (5.3). En appliquant l'espérance, on a par linéarité :} \\
=& \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log\left(\widehat{\pi}_k\right) + \mathbb{E}_{\mu_k}\left[\log p\left(\mathbf{y}_i \mid Z_{ik}=1, \mu_k(\mathbf{t}_i), \widehat{\theta}_i, \widehat{\sigma}_i^2\right)\right]\right] + C_2 \\
=& \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log\left(\widehat{\pi}_k\right) - \frac{1}{2}\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right| - \frac{1}{2}\mathbb{E}_{\mu_k}\left[\left(\mathbf{y}_i - \mu_k(\mathbf{t}_i)\right)^\top \boldsymbol{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\left(\mathbf{y}_i - \mu_k(\mathbf{t}_i)\right)\right]\right] + C_3,
\end{aligned}$$

où l'on a utilisé dans la dernière égalité l'expression quadratique en (4.7) de la densité de probabilité d'un échantillon gaussien multi-dimensionnel.
Maintenant en appliquant notre lemme 4.2.2 à cette espérance on obtient que :

$$\log \widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log\left(\widehat{\pi}_k\right) - \frac{1}{2}\left(\log\left|\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right| + \left(\mathbf{y}_i - \widehat{m}_k\left(\mathbf{t}_i\right)\right)^{\top}\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\left(\mathbf{y}_i - \widehat{m}_k\left(\mathbf{t}_i\right)\right)\right)\right.$$
$$\left. - \frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}_i}\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}{}^{-1}\right)\right] + C_3$$
$$= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log\tau_{ik}\right],$$

où en inspectant des formes de distributions gaussiennes et multinomiales, l'on a posé :

$$\tau_{ik} = \frac{\widehat{\pi}_k\mathcal{N}\left(\mathbf{y}_i;\widehat{m}_k\left(\mathbf{t}_i\right),\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right)\exp\left(-\frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}_i}\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2} - 1\right)\right)}{\sum_{l=1}^{K}\widehat{\pi}_l\mathcal{N}\left(\mathbf{y}_i;\widehat{m}_l\left(\mathbf{t}_i\right),\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\right)\exp\left(-\frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}_l^{\mathbf{t}_i}\mathbf{\Psi}^{\mathbf{t}_i}_{\widehat{\theta}_i,\widehat{\sigma}_i^2} - 1\right)\right)}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{4.8}$$

qu'on peut interpréter comme étant la responsabilité que la classe k assume dans l' explication des observations x . Et ce faisant, la solution optimale peut être écrite sous une forme factorisée de distributions multinomiales de la forme :

$$\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M}\mathcal{M}\left(\mathbf{Z}_i;1,\boldsymbol{\tau}_i = \left(\tau_{i1},\ldots,\boldsymbol{\tau}_{iK}\right)^{\top}\right) \tag{4.9}$$

■

De même l'approximation variationnelle optimale de la distribution $q_{\mathbf{Z}}(\mathbf{Z})$ est donnée par la proposition suivante :

Proposition 4.2.3. *Supposons que* $\widehat{\Theta}$ *et la distribution variationelle* $\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M}\mathcal{M}\left(\mathbf{Z}_i;1,\boldsymbol{\tau}_i\right)$ *sont connus. L'approximation variationelle optimale* $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ *de la vraie distribution hyper-postérieure* $p\left(\boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \widehat{\Theta}\right)$ *se factorise en un produit de distributions multinomiales gaussiennes :*

$$\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K}\mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathrm{t}}\right),$$

avec :

- $\widehat{\mathbf{C}}_k^{\mathbf{t}} = \left(\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}{}^{-1} + \sum_{i=1}^{M}\tau_{ik}\tilde{\mathbf{\Psi}}_i^{-1}\right)^{-1}, \forall k \in \mathcal{K}$
- $\widehat{m}_k(\mathbf{t}) = \widehat{\mathbf{C}}_k^{\mathbf{t}}\left(\mathbf{C}_{\widehat{\gamma}_k}{}^{-1}m_k(\mathbf{t}) + \sum_{i=1}^{M}\tau_{ik}\tilde{\mathbf{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i\right), \forall k \in \mathcal{K}$

et où l'on a définit :

- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t\in\mathbf{t}_i]} \times y_i(t)\right)_{t\in\mathbf{t}}$*(un vecteur de dimension N),*
- $\tilde{\mathbf{\Psi}}_i = \left[\mathbb{1}_{[t,t'\in\mathbf{t}_i]} \times \psi_{\widehat{\theta}_i,\widehat{\sigma}_i^2}\left(t,t'\right)\right]_{t,t'\in\mathbf{t}}$*(Une matrice N× N).*

Preuve. La preuve est similaire comme pour l'approximation variationnelle de $q_{\mathbf{Z}}(\mathbf{Z})$, et ici on a toujours d'après la solution optimale donnée en (4.7 :

$$\begin{aligned}
\log \widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) =& \mathbb{E}_{\mathbf{Z}}\left[\log p\left(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \widehat{\Theta}\right)\right] + C_1 \\
=& \mathbb{E}_{\mathbf{Z}}\left[\log p\left(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i\right) + \log p(\mathbf{Z} \mid \widehat{\boldsymbol{\pi}}) + \log p\left(\boldsymbol{\mu} \mid \{\widehat{\gamma}_k\}_k\right)\right] + C_1 \\
& \text{où l'on a utilisé la décomposition } (4.5) \text{ et en absorbant les termes indépendants de } \mu \\
& \text{dans la consatnte } C_1, \text{ on a :} \\
=& \mathbb{E}_{\mathbf{Z}}\left[\log p\left(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\widehat{\theta}_i\right\}_i, \{\widehat{\sigma}_i^2\}_i\right)\right] + \log p\left(\boldsymbol{\mu} \mid \{\widehat{\gamma}_k\}_k\right) + C_2 \\
=& \sum_{i=1}^{M} \mathbb{E}_{\mathbf{Z}_i}\left[\log p\left(\mathbf{y}_i \mid \mathbf{Z}_i, \boldsymbol{\mu}, \widehat{\theta}_i, \widehat{\sigma}_i^2\right)\right] + \sum_{k=1}^{K} \log p\left(\mu_k(\mathbf{t}) \mid \widehat{\gamma}_k\right) + C_2 \\
=& \sum_{i=1}^{M}\sum_{k=1}^{K} \mathbb{E}_{\mathbf{Z}_i}[Z_{ik}] \log p\left(\mathbf{y}_i \mid Z_{ik}=1, \mu_k(\mathbf{t}_i), \widehat{\theta}_i, \widehat{\sigma}_i^2\right) + \sum_{k=1}^{K} \log p\left(\mu_k(\mathbf{t}) \mid \widehat{\gamma}_k\right) + C_2, \\
& \text{cette fois en utilisant les formules (5.2) et (5.3).} \\
=& -\frac{1}{2}\sum_{k=1}^{K}\left[\left(\mu_k(\mathbf{t}) - m_k(\mathbf{t})\right)^{\top} {\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} \left(\mu_k(\mathbf{t}) - m_k(\mathbf{t})\right)\right. \\
& \left. + \sum_{i=1}^{M} \tau_{ik} \left(\mathbf{y}_i - \mu_k(\mathbf{t}_i)\right)^{\top} {\boldsymbol{\Psi}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}}^{-1} \left(\mathbf{y}_i - \mu_k(\mathbf{t}_i)\right)\right] + C_3,
\end{aligned}$$

où l'on a utilisé dans la dernière égalité l'expression quadratique en (4.7) de la densité de probabilité d'un échantillon gaussien multi-dimensionnel et en absorbant les termes constants dans la constante C_2, donnant la constante C_3.

Maintenant, on va élargir les y_i et $\Psi^{\mathbf{t}_i}_{\widehat{\theta}_i, \widehat{\sigma}_i^2}$, en tout $t \in\in \mathbf{t},$, avec des zéro pour des $t \notin \mathbf{t}_i$, et en définissant :

★ - $\forall i \in \mathcal{I}, \tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t\in\mathbf{t}_i]} \times y_i(t)\right)_{t\in\mathbf{t}}$,un vecteur de dimension N,

★ $\forall i \in \mathcal{I}, \tilde{\Psi}_i = \left[\mathbb{1}_{[t,t'\in\mathbf{t}_i]} \times \psi_{\widehat{\theta}_i, \widehat{\sigma}_i^2}\left(t, t'\right)\right]_{t,t'\in\mathbf{t}}$, une matrice $N \times N$.

Ensuite, en rassemblant les facteurs en $\boldsymbol{\tau}_{ik}\tilde{\boldsymbol{\Psi}}_i^{-1}$, on va reconnaître deux termes quadratiques de vraisemblances gaussiennes en les variables $\mu_k(\cdot)$. Ce qui donne en rassemblant les termes contants dans la constante C_3, qui nous donne C_4 :

$$\begin{aligned}
\log \widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) =& -\frac{1}{2}\sum_{k=1}^{K} \mu_k(\mathbf{t})^{\top} \left({\mathbf{C}_{\widehat{\gamma}_k}}^{-1} + \sum_{i=1}^{M} \boldsymbol{\tau}_{ik}\tilde{\boldsymbol{\Psi}}_i^{-1}\right) \mu_k(\mathbf{t}) \\
& + \mu_k(\mathbf{t})^{\top} \left({\mathbf{C}_{\widehat{\gamma}_k}}^{-1} m_k(\mathbf{t}) + \sum_{i=1}^{M} \boldsymbol{\tau}_{ik}\tilde{\boldsymbol{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i\right) + C_4.
\end{aligned}$$

Et on reconnaît bien d'après la formule (4.7), une somme de vraissemblances gaussiennes.

Ce qui donne en rassemblant les termes contants dans la constante C_3, qui nous donne C_4 :

$$\log \widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = -\frac{1}{2}\sum_{k=1}^{K} \mu_k(\mathbf{t})^\top \left({\mathbf{C}_{\widehat{\gamma}_k}}^{-1} + \sum_{i=1}^{M} \boldsymbol{\tau}_{ik} \tilde{\boldsymbol{\Psi}}_i^{-1} \right) \mu_k(\mathbf{t}) + \mu_k(\mathbf{t})^\top \left({\mathbf{C}_{\widehat{\gamma}_k}}^{-1} m_k(\mathbf{t}) + \sum_{i=1}^{M} \boldsymbol{\tau}_{ik} \tilde{\boldsymbol{\Psi}}_i^{-1} \tilde{\mathbf{y}}_i \right) + C_4$$

Et on en déduit donc la forme inspectée

$$\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right),$$

où l'on a posé :

- $\widehat{\mathbf{C}}_k^{\mathbf{t}} = \left({\mathbf{C}_{\widehat{\gamma}_k}}^{-1} + \sum_{i=1}^{M} \tau_{ik} \tilde{\Psi}_i^{-1} \right)^{-1}, \forall k \in \mathcal{K}$
- $\widehat{m}_k(\mathbf{t}) = \widehat{\mathbf{C}}_k^{\mathbf{t}} \left({\mathbf{C}_{\widehat{\gamma}_k}^{\mathbf{t}}}^{-1} m_k(\mathbf{t}) + \sum_{i=1}^{M} \tau_{ik} \tilde{\Psi}_i^{-1} \tilde{\mathbf{y}}_i \right), \forall k \in \mathcal{K}$

■

Une fois ces approximations faites, on est alors prêt à procéder à l'étape M de la méthode EM, qui consiste à maximiser la vraisemblance $\mathcal{L}(\widehat{q}; \Theta)$ pour trouver $\hat{\Theta}$. C'est une étape qui dépend des hypothèses initiales du modèle génératif, ce qui donne quatre versions différentes pour l'algorithme VEM.
Remarquons avant, que selon les hypothèses faites sur les hyper-paramètres du modèle, on obtient 4 différentes situations suivantes :

	$\theta_0 = \theta_i, \forall i \in \mathcal{I}$		$\theta_i \neq \theta_j, \forall i \neq j$	
	Notation	Nb of HPs	Notation	Nb of HPs
$\gamma_0 = \gamma_k, \forall k \in \mathcal{K}$	$\mathcal{H}_{00}$	2	$\mathcal{H}_{0i}$	M + 1
$\gamma_k \neq \gamma_l, \forall k \neq l$	$\mathcal{H}_{k0}$	K + 1	$\mathcal{H}_{ki}$	M + K

Ainsi $\hat{\Theta}$ est donné par le théorème suivant :

Proposition 4.2.4. *Supposons que les distributions variationnelles* $\widehat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i)$ *et* $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \widehat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right)$ *sont connues. Alors pour un vecteur d'hyper-paramètres* $\Theta = \left\{ \{\gamma_k\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i, \boldsymbol{\pi} \right\}$*, les valeurs optimales sont données par*

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \mathbb{E}_{\{\mathbf{Z}, \boldsymbol{\mu}\}} \left[\log p\left(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \Theta \right) \right]$$

avec $\mathbb{E}_{\{\mathbf{Z}, \boldsymbol{\mu}\}}$ *l'espérance sous la distribution* $\widehat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ *et* $\widehat{q}_{\mathbf{Z}}(\mathbf{Z})$.
Le calcul des valeurs optimales de $\boldsymbol{\pi}$ *se fait de la façon :*

$$\widehat{\pi}_k = \frac{1}{M}\sum_{i=1}^{M} \tau_{ik}, \forall k = 1, \cdots, K.$$

Et en notant

$$\mathcal{L}_k(\mathbf{x}; \mathbf{m}, S) = \log \mathcal{N}(\mathbf{x}; \mathbf{m}, S) - \frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}} S^{-1}\right)$$

$$\mathcal{L}_i(\mathbf{x}; \mathbf{m}, S) = \sum_{k=1}^{K} \tau_{ik}\left(\log \mathcal{N}(\mathbf{x}; \mathbf{m}, S) - \frac{1}{2}\operatorname{tr}\left(\widehat{\mathbf{C}}_k^{\mathbf{t}_i}, S^{-1}\right)\right)$$

les autres paramètres en résolvant numériquement les problèmes d'optimisation suivants, selon qu'on est dans chacun des 4 situations :

- $\widehat{\gamma}_k = \underset{\gamma_k}{\operatorname{argmax}} \mathcal{L}_k\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right), \forall k = 1, \cdots, K,$
- $\left(\widehat{\theta}_i, \widehat{\sigma}_i^2\right) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \mathcal{L}_i\left(\mathbf{y}_i; \widehat{m}_k\left(\mathbf{t}_i\right), \mathbf{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right), \forall i \in I.$

Pour $\mathcal{H}_{k0}$:

- $\widehat{\gamma}_k = \underset{\gamma_k}{\operatorname{argmax}} \mathcal{L}_k\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right), \forall k = 1, \cdots, K,$
- $\left(\widehat{\theta}_0, \widehat{\sigma}_0^2\right) = \underset{\theta_0, \sigma_0^2}{\operatorname{argmax}} \sum_{i=1}^{M} \mathcal{L}_i\left(\mathbf{y}_i; \widehat{m}_k\left(\mathbf{t}_i\right), \mathbf{\Psi}_{\theta_0, \sigma_0^2}^{\mathbf{t}_i}\right).$

Pour $\mathcal{H}_{0i}$:

- $\widehat{\gamma}_0 = \underset{\gamma_0}{\operatorname{argmax}} \sum_{k=1}^{K} \mathcal{L}_k\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_0}^{\mathbf{t}}\right),$
- $\left(\widehat{\theta}_i, \widehat{\sigma}_i^2\right) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \mathcal{L}_i\left(\mathbf{y}_i; \widehat{m}_k\left(\mathbf{t}_i\right), \mathbf{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right), \forall i \in I.$

Pour $\mathcal{H}_{00}$:

- $\widehat{\gamma}_0 = \underset{\gamma_0}{\operatorname{argmax}} \sum_{k=1}^{K} \mathcal{L}_k\left(\widehat{m}_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_0}^{\mathbf{t}}\right),$
- $\left(\widehat{\theta}_0, \widehat{\sigma}_0^2\right) = \underset{\theta_0, \sigma_0^2}{\operatorname{argmax}} \sum_{i=1}^{M} \mathcal{L}_i\left(\mathbf{y}_i; \widehat{m}_k\left(\mathbf{t}_i\right), \mathbf{\Psi}_{\theta_0, \sigma_0^2}^{\mathbf{t}_i}\right).$

4.3 Un exemple numérique d'utilisation de l'algoritme EM variationnel

Dans cette partie, nous illustrons à travers un exemple comment l'algoritme EM variationnel est utilisé dans le cadre du modèle MagmaClust pour estimer les paramètres optimaux du modèle.

Dans un premier temps, nous simulons des données numériques en dimension 2, suivant le format adapté pour Magmaclust et grâce à la fonction *simuldb*, puis nous divisons ces données en données d'entrainement et données de validation à l'aide de la fonction *datamagmaclust*.

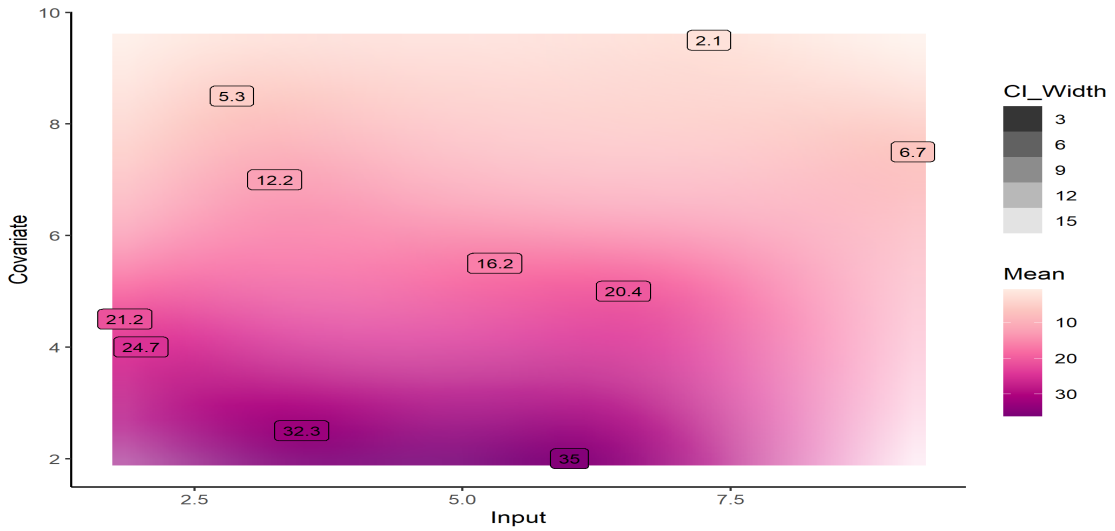
```
## Dataset with 11 individuals, 10 reference input locations
   and a covariate
set.seed(5)
data_dim2 <- simu_db(M = 11, N = 10, covariate = TRUE)
## Split individuals into training and prediction sets,
   and define test points
dim2_train <- data_dim2 %>% subset(ID %in% 1:10)
dim2_pred <- data_dim2 %>% subset(ID == 11)

data_dim2
#> # A tibble: 110 x 4
#>    ID    Input Covariate Output
#>    <chr> <dbl>     <dbl>  <dbl>
#>  1 1      1.85       4.5  21.3
#>  2 1      2.85       8.5   5.21
#>  3 1      2          4    24.8
#>  4 1      3.25       7    12.3
#>  5 1      3.5        2.5  37.0
#>  6 1      5.3        5.5  20.0
#>  7 1      6.5        5    20.8
#>  8 1      6          2    34.1
#>  9 1      7.3        9.5   5.18
#> 10 1      9.2        7.5   7.20
#> # i 100 more rows
```

En suite nous entraînons notre modèle pour dériver les paramètres optimaux. Comme annoncé plus haut, l'agorithm EM variationnel est utilisé ici pour l'estimation, et comme on peut le voir, la méthode converge bien après quelques itérations.

```
model_dim2 <- train_magma(data = dim2_train)
```

Une fois ceci fait, nous disposons de tous les ingrédients nécessaires pour pouvoir faire des prévisions avec notre modèle. Dans la suite, la fonction *predmagmaclust* du package MagmaClutR permet cette tâche tout en prenant soin de représenter les graphiques associés.

```
pred_dim2  <- pred_magma(data = dim2_pred,
                         trained_model = model_dim2)
#> The hyper-posterior distribution of the mean process

# provided in 'hyperpost' argument isn't evaluated on the
# expected inputs.
#>
#> Start evaluating the hyper-posterior on the correct inputs
#>
#> The 'prior_mean' argument has not been specified.
The hyper-prior mean function is thus set to be 0 everywhere.
#>
#> Done!
```



CLUSTERING DE DONNÉES BINAIRES VIA UN MODÈLE DE MÉLANGE DE GP

5.1 Introduction

En nous appuyant sur les recherches antérieures de Leroy et al (2022) [1], notre objectif est d'élargir le modèle qu'ils ont proposé pour qu'il puisse être appliqué à des données binaires. La principale distinction par rapport aux travaux précédents réside dans le fait que, dans notre cas, les courbes y_i des individus sont ce que nous appelons des "variables latentes", notées y_i^*, qui ne sont pas directement observées. Ces variables latentes sont modélisées à l'aide d'un modèle de mélange gaussien multitâches, ce qui signifie qu'il peut y avoir plusieurs processus gaussiens distincts associés à chaque cluster, chacun ayant sa propre structure de covariance individuelle pour chaque donnée fonctionnelle. Les valeurs observées, notées y_i, représentent les indicateurs des valeurs prises par ces variables latentes.

Du fait de la présence des variables latentes dans notre modélisation, l'estimation par l'algorithme EM semble bien être adapté à notre contexte. De plus, à cause de la complexité du vecteur de ces variables latentes, nous utilisons ici une variante stochastique de l'algorithme EM pour la procédure d'estimation les hyperparamètres du modèle, ainsi que les distributions hyper-postérieures des processus moyens et des variables latentes.

De plus, nous fournissons des expressions analytiques permettant de calculer les probabilités d'appartenance aux clusters.

5.2 Le modèle

La notion fondamentale derrière notre regroupement binaire consiste à supposer qu'un mélange gaussien sert d'a priori pour les fonctions latentes du modèle. Et ensuite les relier aux courbes des individus par un schéma de modéle probit, convenablement choisi.
Nous avons choisi ici dans le formalisme du modèle latent, un seuil égale à zéro, bien que d'autres seuils pourraient être considérés suivant différents cas de modélisations, donnant lieu par exemple à des schémas de la forme :

$$y_i = \begin{cases} 0 & si \quad y_i^* \leqslant \alpha_1 \\ 1 & si \quad \alpha_1 < y_i^* \leqslant \alpha_2, \end{cases}$$

α_1, α_2 correspondent aux seuils inconnus des différentes étapes du processus, avec $(\alpha_1 < \alpha_2)$.

5.2.1 Notations

Tout au long de notre travail, nos variables d'entrées se réfèreront à des instants d'observation sur les individus, prenant leurs valeurs dans T, un intervalle quelconque de $\mathbb{R}$. Et nous notons par I, l'ensemble des indices des individus observés, contenant notamment $1,\cdots,n$. Maintenant, puisque nous sommes dans un cadre de classification, on notera l'ensemble des indices des K-différents groupes par $\{1,\cdots,K\}$, et pour simplifier on adopte la notation $\{x_i\}_i = \{x_1,\ldots,x_n\}$ et également $\{x_k\}_k = \{x_1,\ldots,x_K\}$
Nous considérons que les données nous proviennent de n différents individus et pour chaque individu, nous supposons qu'il dispose de n_i données et ce faisant nous adoptons les notations suivantes :

— $\mathbf{t}_i = (t_{i1},\ldots,t_{in_i}) \in T^{n_i}$, l'ensemble des instants d'observations pour un individu i

— $\mathbf{y}_i = \left(y_{ij}\right)_{1\leq j\leq n_i}, \quad y_{ij} = Y_i\left(t_{ij}\right) \in \{0,1\}$, le vecteur des sorties d'un individu i,

— $\mathbf{y}_i^* = \left(y_{ij}^*\right)_{1\leq j\leq n_i}, \quad y_{ij}^* = y_i^*\left(t_{ij}\right)$, les variables latentes (inobservées sur l'individu i)

— $\mathbf{t} = \bigcup_{i=1}^{n} \mathbf{t}_i$, la grille commune des instants d'observation pour tous les individus.

Soulignons que dans notre contexte, les entrées peuvent varier à la fois en nombre et en emplacement d'un indivdu à un autre. Afin de définir un modèle de mélange de gaussiens, un vecteur aléatoire binaire latent $z_i \sim \mathcal{M}(1,\pi), 1\leq i\leq n$ est associé à chaque individu, indiquant la classe à laquelle il appartient : $z_{ik} = 1$ si l'individu i est dans la classe k, 0 sinon. De plus, nous supposons que ces variables latentes sont distribuées selon la même distribution multinomiale : $z_i \sim \mathcal{M}(1,\pi), 1\leq i\leq n$, avec le vecteur des proportions du mélange gaussien $\pi = (\pi_1,\cdots,\pi_K)^t$ vérifiant $\sum_{k=1}^{K} \pi_k = 1$.

5.2.2 Modèle et hypothèses

Pour un individu i appartenant à une classe k donné, l'expression de la fonction latente associée à sa sortie y_i s'écrit :

$$y_i^*(t) = \mu_k(t) + f_i(t) + \varepsilon_i(t), \quad t\in T, \quad \text{et}$$
$$y_i(t) = \begin{cases} 1 & \text{si} \quad y_i^*(t) > 0, \\ 0 & \text{sinon.} \end{cases}$$

avec pour $1\leq i\leq n$, et $1\leq k\leq K$,
— $\mu_k(\cdot) \sim \mathcal{GP}\left(m_k(\cdot), c_{\gamma_k}(\cdot,\cdot)\right)$: le processus moyen commun spécifique à chaque cluster k,

— $f_i(\cdot) \sim \mathcal{GP}\left(0, \xi_{\theta_i}(\cdot,\cdot)\right)$: le processus spécifique à l'individu i,

— $\varepsilon_i(\cdot) \sim \mathcal{GP}(0, Id)$: le bruit associé au processsus d'un individu i donné,

— $\Theta = \left\{ (\gamma_k)_{1\leq k\leq K}, (\theta_i)_{1\leq i\leq n}, (\sigma_i^2)_{1\leq i\leq n}, \pi \right\}$: les paramètres du modèle.
Les hyper-paramètres du modèle dans ce cadre étant :

➢ $\forall k = 1,\cdots,K, m_k(\cdot)$ est la fonction a priori sur le processus μ_k associé au cluster k,
➢ $\forall k = 1,\cdots,K, C_{\gamma_k}(\cdot,\cdot)$ le noyau de covariance associé à la classe k et paramétré par γ_k,
➢ $\forall i \in I, \xi_{\theta_i}(\cdot,\cdot)$ est le noyau associé à l'individu i, de paramètre paramétré par θ_i,
➢ $\forall i \in I, \sigma_i^2 \in \mathbb{R}$ le bruit associé à l'individu i,
➢ $\forall i \in I,$ on pose $\psi_{\theta_i,\sigma_i^2}(\cdot,\cdot) = \xi_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I$,
Les hypothèses faites sur les processus intégrés dans le modèle sont :
Hypothèses
— $\mathbf{H}_1 : \{\mu_k\}_{1\leq k\leq K}$ sont indépendants,
— $\mathbf{H}_2 : \{f_i\}_{1\leq i\leq n}$ sont indépendants,
— $\mathbf{H}_3 : \{z_i\}_{1\leq i\leq n}$ sont indépendants,
— $\mathbf{H}_4 : \{\varepsilon_i\}_{1\leq i\leq n}$ sont indépendants,
— $\mathbf{H}_5$: Pour tout $1 \leq i \leq n, \quad 1 \leq k \leq K, \quad \mu_k, \quad f_i, \quad z_i$ sont indépendants.

De par ces hypothèses, les variables latentes y_i^* sont indépendantes, conditionnellement au processus $\mu_k \quad et \quad Z_i$, si nous intégrons sur les f_i.
Nous tenons à souligner le fait que seules les variables y_i sont effectivement observées ; les autres variables entrant dans cette modélisation à savoir les variables latentes y_i^*, les processus moyens $\{\mu_k(.)\}_k$ de chacun des K clusters et les variables d'affectations Z_i des individus sont toutes ici inobservées et constituent ainsi le vecteur des variables latentes dans la suite du travail. Si nous nous intéressons à l'estimation des hyperparamètres de ce nouveau modèle, il nous faut préciser la vraissemblance associée à un échantillon $\{\mathbf{t}_i, \mathbf{y}_i\}_i$ que l'on dispose. Dans le cas de notre modèle dichotomique, cette vraissemblance s'écrit :

$$p(\{\mathbf{y}_i\}_i \mid \{t_i\}_i, \Theta) = \prod_{i=1}^{n} \mathbb{P}(y_i = 1 \mid t_i, \Theta)^{y_i} \times \mathbb{P}(y_i = 0 \mid t_i, \Theta)^{1-y_i}.$$

Cette dernière quantité dépendant de plusieurs quantités en pratique inconnues, est incalculable. Nous allons donc considérer ici la vraissemblance complète de nos données.
D'une part, la vraissemblance jointe conditionnelle des variables latentes peut s'écrire comme dans le modèle de Leroy et al (2022)[1] :

$$\begin{aligned} p\left(\{\mathbf{y}_i^*\}_i \mid \{t_i\}_i, \{\mathbf{z}_i\}_i, \{\mu_k(\mathbf{t})\}_k, \{\theta_i\}_i, \{\sigma_i^2\}_i\right) &= \prod_{i=1}^{n} p\left(\mathbf{y}_i^* \mid \mathbf{z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i, \sigma_i\right) \\ &= \prod_{i=1}^{n}\prod_{k=1}^{K} p\left(\mathbf{y}_i^* \mid z_{ik} = 1, \mu_k(\mathbf{t}_i), \theta_i, \sigma_i\right)^{z_{ik}} \quad (5.1) \\ &= \prod_{i=1}^{n}\prod_{k=1}^{K} \mathcal{N}\left(\mathbf{y}_i^*; \mu_k(\mathbf{t}_i), \Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)^{z_{ik}}, \end{aligned}$$

où $\forall i \in J, \Psi^{\mathbf{t}_i}_{\theta_i,\sigma_i^2} = \psi_{\theta_i,\sigma_i^2}(\mathbf{t}_i, \mathbf{t}_i) = \left[\psi_{\theta_i,\sigma_i^2}(u,v)\right]_{u,v\in\mathbf{t}_i}$ désigne une matrice de variance-covariance de dimension $n_i \times n_i$. De même rappellons, toujours dans cette identification de modèle comme dans Leroy et al (2022)[1], que les distributions a priori des processus moyens pour chaque

classe évaluées sur la grille commune d'observation $\mathbf{t}$ s'écrit également :

$$\begin{aligned} p\left(\{\mu_k(\mathbf{t})\}_k \mid \{\gamma_k\}_k\right) &= \prod_{k=1}^{K} p\left(\mu_k(\mathbf{t}) \mid \gamma_k\right) \\ &= \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right), \end{aligned} \tag{5.2}$$

où $\mathbf{C}^{\mathbf{t}}_{\gamma_k} = C_{\gamma_k}(\mathbf{t},\mathbf{t}) = \left[C_{\gamma_k}(k,\ell)\right]_{k,l\in\mathbf{t}}$ est une matrice de variance-covariance de dimension $n \times n$, et pour finir les distributions pour les variables d'affectation Z_i des individus sont données par :

$$\begin{aligned} p\left(\{\mathbf{z}_i\}_i \mid \boldsymbol{\pi}\right) &= \prod_{i=1}^{n} p\left(\mathbf{z}_i \mid \boldsymbol{\pi}\right) \\ &= \prod_{i=1}^{n} \mathcal{M}\left(\mathbf{z}_i; 1, \boldsymbol{\pi}\right) = \prod_{i=1}^{n}\prod_{k=1}^{K} \pi_k^{z_{ik}}. \end{aligned} \tag{5.3}$$

Ce qui fait qu'en rassemblant ses termes, on peut en déduire la vraissemblance complète du modèle. Pour ce faire, définissons la fonctionnelle h en posant pour tout $u \in \mathbb{R}$,

$$h(u) = \begin{cases} 1 & \text{si } u > 0, \\ 0 & \text{sinon} \end{cases}, \quad \text{et en adoptant la notation}$$

$$h\left(\left(u_j\right)_{1\le j\le J}\right) = \left(h\left(u_j\right)\right)_{1\le j\le J}, \quad \forall \left(u_j\right)_{1\le j\le J} \in \mathbb{R}^J, \quad \text{pour} \quad J \ge 1, \quad \text{On aura que :}$$

$$\begin{aligned} p\left(\{y_i\}_i, \{y_i^\star\}_i, \{z_i\}_i, \{\mu_k(t_i)\}_{i,k} \mid \{t_i\}_i, \Theta\right) &= p\left(\{y_i\}_i, \{y_i^\star\}_i \mid \{\mathbf{z}_i\}_i, \{\mu_k(\mathbf{t})\}_k, \{t_i\}_i, \Theta\right) \\ &\times p\left(\{\mathbf{z}_i\}_i, \{\mu_k(\mathbf{t})\}_k \mid \{t_i\}_i, \Theta\right) \\ &= p\left(\{y_i\}_i, \{y_i^\star\}_i \mid \{\mathbf{z}_i\}_i, \{\mu_k(\mathbf{t})\}_k, \{t_i\}_i, \Theta\right) \\ &\times p\left(\{\mathbf{z}_i\}_i \mid \boldsymbol{\pi}\right) \times p\left(\{\mu_k(\mathbf{t})\}_k \mid \{\gamma_k\}_k\right) \\ &= \prod_{i=1}^{n} \mathbb{1}_{\left\{h\left(y_i^*\right)=y_i\right\}} \prod_{k=1}^{K} \left[\pi_k \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i,\sigma_i^2}\right)\right]^{Z_{ik}} \\ &\times \left(\prod_{k=1}^{K} \mathcal{N}\left(\mu_k(t); m_k(t), C^t_{\gamma_k}\right)\right), \end{aligned} \tag{5.4}$$

par des formules de Bayes et en se servant des équations (5.1), (5.2), (5.3) et de l'hypothèse $\mathbf{H}_5$.

Vue la forme de cette vraissemblance, qui dépend de variables inobserbées ($\{\mathbf{y}_i^*\}_i, \{\mathbf{z}_i\}_i, \{\mu_k(\cdot)\}_k$), elle est en pratique incalculable. Ce qu'on pourrait faire peut-être, et vue la dimension ici du vecteur des variables latentes, c'est de faire des approximations optimales pour les distributions postérieures de ces variables (qui se factorisent alors bien en $\{\mathbf{y}_i^*\}_i, \{\mathbf{z}_i\}_i, \{\mu_k(\cdot)\}_k$, voir Chapitre 2) et de procéder par une méthode variationnelle de l'algorithme EM exposé au Chapitre 2. Nous avons opté pour la variante stochastique de l'algoritme EM qui dans notre contexte semble être mieux adaptée pour l'estimation de nos hyperparamètres.

5.2.3 Choix du noyau de covariance

Dans le contexte d'un modèle de prédicitons par des processus gaussiens, la fonction de covariance constitue un ingrédient crucial, en ce sens qu'elle permet d'intégrer nos hypothèses sur la fonction que nous souhaitons apprendre. Par exemple en apprentissage, c'est la notion de similitude entre les points de données qui regorge cette importance capitale ; En fait c'est une hypothèse de base qui permet de dire par exemple que les points qui ont des entrées x proches sont susceptibles d'avoir des sories y similaires, et donc que les données d'entraînement qui sont proches d'un nouveau point sont en quelque sorte informatives pour sa prédiction. Revenant au cas des processus gaussiens, c'est la fonction de covariance qui définit cette proximité ou similarité. Elle permet de spécifier la structure de dépendance spatiale ou temporelle des variables aléatoires du processus.
Dans la littérature, la fonction de covariance généralement utilisée est bien-sûr le noyau quadratique exponentiel, qui dépend de deux hyperparamètres disons v, l donné par :

$$k_{\mathrm{EQ}}\left(x, x'\right) = v^2 \exp\left(-\frac{\left(x - x'\right)^2}{2\ell^2}\right).$$

La constante l définit ce qu'on appelle *échelle de longueur* : pour un processus gaussien unidimensionnel par exemple, une façon de comprendre l'échelle de longueur réside en le nombre de d'affranchissement d'un niveau u donné. Il a été montré par exemple dans Alder, 1981 (Théorème 4.1.1)[39] que le nombre moyen d'affranchissement d'un niveau u donné, sur l'intervalle [0,1], par un processus gaussien stationnaire, p.s continu est donné par la formule :

$$\mathbb{E}\left[N_u\right] = \frac{1}{2\pi}\sqrt{\frac{-k''(0)}{k(0)}}\exp\left(-\frac{u^2}{2k(0)}\right)$$

Par cette formule, on voit par exemple que le nombre moyen d'affranchissent d'un niveau 0 pour un processus en dimension 1 est de $(2\pi l)^{-1}$, ce qui confirme bien le rôle d'échelle joué par le paramètre l.

5.3 Procédure d'estimation

La forme complexe de la vraissemblance du modèle ne permettant pas une application des méthodes classiques d'estimations de hyper-paramètres, et vue la dimensionalité du vecteur des variables latentes, une méthode variationnelle de l'algorithme EM est bien adaptée pour le modèle que nous considérons ici (voir Leroy et al (2022) [1]) : on pourrait par exemple se restreindre à une famille de distributions sur les variables latentes dans laquelle une hypothèse d'indépendance permettrait de dériver des approximations optimales des vraies distributions de ces variables latentes, utilisées à l'étape E de l'algorithm pour le calcul de la borne inférieure de la vraie vraissemblance (voir 3.1).
Cependant, vue la complexité du problème d'estimation considéré dans notre cas présent, nous avons opté pour une variante stochastique de l'algorithme EM : la méthode procède à un tirage aléatoire d'un vecteur des variables latentes $(y^\star, z, \mu)$ étant donnés les hyper-paramètres actuels du modèle et suivant la densité $p(y^\star, z, \mu)$, ce qui facilite le calcul de la vraissemblance complète

à l'étape E de algorithme. En suite on actualise les valeurs des hyper-paramètres en optimisant cette vraissemblance à l'étape M.

Tout comme dans Leroy et al. (2022)[1], nous considérons ici les différentes possibilités de situations pour les hyperparamètres de notre modèle, consistant essentillement en du compromis entre flexibilité et partage d'informations, ou encore en un choix de modélisation individuelle ou collective dans la structure de covariance. Ces différents cas de figures conduisent à quatre différents cas pour notre modèle, comme resumé dans le tableau 5.1suivant :

	$\theta_0 = \theta_i, \forall i \in \mathcal{I}$		$\theta_i \neq \theta_j, \forall i \neq j$	
	Notation	Nb des HPs	Notation	Nb des HPs
$\gamma_0 = \gamma_k, \forall k \in \mathcal{K}$	$\mathcal{H}_{00}$	2	$\mathcal{H}_{0i}$	M + 1
$\gamma_k \neq \gamma_l, \forall k \neq l$	$\mathcal{H}_{k0}$	K + 1	$\mathcal{H}_{ki}$	M + K

TABLE 5.1 – Différentes hypothèses pour les hyper-paramètres

5.3.1 L'algoritme EM Stochastique (SEM)

Etape E

A l'étape d'espérance (E) de l'algorithm SEM, en considérant que des valeurs initiales ou précédemment estimées des hyper-paramètres Θ^q sont disponibles, on procède à un échantillonnage d'un vecteur $(y^\star, z, \mu)$ suivant la distribution $p(y^\star, z, \mu)$. On utilise à cette fin, une itération de l'algorithm de Gibbs(une méthode MCMC permettant d'échantillonner suivant une distribution jointe multidimensionnelle en utilisant des échantillons conditionnels), nécessitant une identification des distributions $p(y^* \mid y, z, \mu, \theta)$, $p(z \mid y, y^*, \mu, \pi, \theta)$, *et* $p(\mu \mid y, y^*, z, \pi, m, \gamma, \theta)$, ce que nous spécifions dans la suite.

Proposition 5.3.1. *Supposons que des valeurs initiales ou précédemment estimées Θ^q des hyperparamètres sont disponibles. Alors on a la forme suivante sur la distribution hyper-postérieure de y^* :*

$$p(y^* \mid y, z, \mu, \theta^q) = \frac{\prod_{i=1}^{n} \mathbb{1}_{\{h(y_i^*)=y_i\}} \prod_{k=1}^{K} \left[\mathcal{N}\left(y_i^*; \mu_k(t_i), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right) \right]^{z_{ik}}}{\prod_{i=1}^{n} \prod_{k=1}^{K} \left[\Phi\left(y_i; \mu_k(t_i), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right) \right]^{z_{ik}}};$$

$$\text{où} \quad \Phi(x; m, \Sigma) = \int_{\mathbb{R}^l} \mathbb{1}_{\{h(u)=x\}} \mathcal{N}(u; m, \Sigma) du, \quad \text{pour} \quad x \in \{0,1\}^l, \quad l \geq 1, \quad m \in \mathbb{R}^l,$$

Σ *une matrice symétrique réelle définie positive. Ainsi, pour chaque individu i on a :*

$$p\left(y_i^* \mid y_i, z_i, \{\mu_k(t_i)\}_k, \theta_i^q\right) = \mathbb{1}_{\{h(y_i^*)=y_i\}} \prod_{k=1}^{K} \left[\frac{\mathcal{N}\left(y_i^*; \mu_k(t_i), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}{\Phi\left(y_i; \mu_k(t_i), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)} \right]^{z_{ik}}.$$

Preuve. D'une part, on se rappelle de la vraissemblance complète du modèle donnée en (5.4) par :

$$p\left(\{y_i\}_i,\{y_i^\star\}_i,\{z_i\}_i,\{\mu_k(t_i)\}_{i,k}\mid\{t_i\}_i,\Theta^q\right)=\left(\prod_{k=1}^{K}\mathcal{N}\left(\mu_k(t);m_k(t),C_{\gamma_k}^t\right)\right)\times\prod_{i=1}^{n}\mathbb{1}_{\{h(y_i^*)=y_i\}}\prod_{k=1}^{K}\left[\pi_k\mathcal{N}\left(y_i^*;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)\right]^{z_{ik}}.$$

et dautre part, on peut écrire que :

$$\begin{aligned}
p(y,z,\mu\mid\{t_i\}_i,\Theta^q)&=p\left(\{y_i\}_i\mid\{z_i\}_i,\{\mu_k(t_i)\}_{i,k},\{t_i\}_i,\Theta^q\right)p\left(\{z_i\}_i,\{\mu_k(t_i)\}_{i,k}\mid\{t_i\}_i,\Theta^q\right)\\
&=\prod_{i=1}^{n}p(y_i\mid z_i,\{\mu_k(t_i)\}_k,t_i,\Theta^q)\,p\left(\{Z_i\}_i,\{\mu_k(t_i)\}_{i,k}\mid\{t_i\}_i,\Theta^q\right)\\
&=\prod_{i=1}^{n}\prod_{k=1}^{K}p(y_i\mid z_{ik}=1,\mu_k(t_i),t_i,\Theta^q)^{z_{ik}}\,p\left(\{Z_i\}_i,\{\mu_k(t_i)\}_{i,k}\mid\{t_i\}_i,\Theta^q\right)\\
&=\prod_{i=1}^{n}\prod_{k=1}^{K}\left[\int_{\mathbb{R}^{n_i}}p(y_i,y_i^*\mid z_{ik}=1,\mu_k(t_i),t_i,\Theta^q)\,dy_i^*\right]^{z_{ik}}p\left(\{z_i\}_i,\{\mu_k(t_i)\}_{i,k}\mid\{t_i\}_i,\Theta^q\right)\\
&=\prod_{i=1}^{n}\prod_{k=1}^{K}\left[\int_{\mathbb{R}^{n_i}}\mathbb{1}_{\{(h(y_{ij}^*)=y_{ij})_{1\le j\le n_i}\}}\mathcal{N}\left(y_i^*;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)dy_i^*\right]^{Z_{ik}}\\
&\times p\left(\{z_i\}_i,\{\mu_k(t_i)\}_{i,k}\mid\{t_i\}_i,\Theta^q\right)\\
&=\prod_{i=1}^{n}\prod_{k=1}^{K}\left[\pi_k\Phi\left(y_i;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)\right]^{z_{ik}}\left(\prod_{k=1}^{K}\mathcal{N}\left(\mu_k(t);m_k(t),C_{\gamma_k}^t\right)\right).
\end{aligned}$$

Avec la notation $\Phi(y_i;\mu_k(t_i),\psi_{\theta_i}^{t_i})=\int_{\mathbb{R}^{n_i}}\mathbb{1}_{\{h(u)=y_i\}}\mathcal{N}(u;\mu_k(t_i),\psi_{\theta_i}^{t_i})du$, où les $y_i\in\{0,1\}^{n_i}$, et $\mu_k(t_i)\in\mathbb{R}^{n_i}$, et nous avons utilisé la définition des variables z_i pour chaque individu i à la troisième égalité.

En prenant le rapport de ces deux densités, on obtient :

$$\begin{aligned}
p(y^*\mid y,z,\mu,\theta^q)&=\frac{p(y,y^*,z,\mu\mid\theta^q)}{p((y,z,\mu\mid\theta^q)}\\
&=\frac{\prod_{i=1}^{n}\mathbb{1}_{\{h(y_i^*)=y_i\}}\prod_{k=1}^{K}\left[\pi_k\mathcal{N}\left(y_i^*;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)\right]^{z_{ik}}}{\prod_{i=1}^{n}\prod_{k=1}^{K}\pi_k^{z_{ik}}\left[\int_{\mathbb{R}^{n_i}}\mathbb{1}_{\{(h(y_{ij}^*)=y_{ij})_{1\le j\le n_i}\}}\mathcal{N}\left(y_i^*;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)dy_i^*\right]^{z_{ik}}}\\
&=\frac{\prod_{i=1}^{n}\mathbb{1}_{\{h(y_i^*)=y_i\}}\prod_{k=1}^{K}\left[\mathcal{N}\left(y_i^*;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)\right]^{z_{ik}}}{\prod_{i=1}^{n}\prod_{k=1}^{K}\left[\Phi\left(y_i;\mu_k(t_i),\Psi_{\theta_i^q,\sigma_i^{2q}}^{t_i}\right)\right]^{z_{ik}}}.
\end{aligned}$$

■

Remarque 5.3.2. *Cette loi hyper-postérieure du vecteur y^* est une loi normale multivariée tronquée, et les simulations que nous faisons plus tard en R à l'intérieur de nos fonctions d'estimations de l'algorithme SEM se font via la fonction rtmvnorm package* TMVTNORM.
Explicitement, la procédure de simulation se fait comme suit :

a- La spécification du nombre de points à simuler

b- L'initialisation des autres paramètres de la fonction, essentiellement : le vecteur moyenne et la matrice de covariance de la normale multivariée sous-jacente, les bornes inférieur et supérieur pour les points de troncature de la distribution.

c- La génération de nombres aléatoires à partir d'une distribution normale multivariée tronquée se fait à l'aide d'un échantillonnage par "rejet" ou d'un échantillonnage de Gibbs.

d- Dans le cas d'un échantillonnage par méthode de Gibbs, fournir facultativement la matrice de précision notée "H"

Voici un exemple d'utilisation de cette fonction en R :

```
#########################################################

lower <- c(-1, -1)
upper <- c(1, 1)
mean <- c(0.5, 0.5)
sigma <- matrix(c(1, 0.8, 0.8, 1), 2, 2)
H <- solve(sigma)
D <- matrix(c(1, 1, 1, -1), 2, 2)
X <- rtmvnorm(n=1000, mean=mean, H=H, lower=lower,
upper=upper, algorithm="gibbs")
plot(X, main="Gibbs sampling with precision matrix")

#########################################################

sigma <- matrix(c(4,2,2,3), ncol=2)
X1 <- rtmvnorm(n=10000, mean=c(1,2), sigma=sigma,
upper=c(1,0), algorithm="rejection")
acf(X1)
```

Proposition 5.3.3. *Supposons que des valeurs initiales ou précédemment estimées Θ^q des hyper-paramètres sont disponibles. Alors la distribution hyper-postérieure du vecteur des variables d'affectation se factorise en un produit de disributions multinomiales :*

$$p\left(z \mid y, y^*, \mu, \pi^q, \gamma^q, \theta^q\right) = \prod_{i=1}^{n} \mathcal{M}\left(\mathbf{z}_i; 1, \boldsymbol{\tau}_i = \left(\tau_{i1}, \ldots, \boldsymbol{\tau}_{iK}\right)^\top\right),$$

$$\textit{avec} \quad \tau_{ik} = \frac{\pi_k^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}{\sum_{k=1}^{K} \pi_k^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{\mathbf{t}_i}_{\theta_i^q, \sigma_i^{2q}}\right)}, \quad i = 1, \cdots, n; \quad k = 1, \cdots, K$$

quantité qu'on peut interpréter comme étant la responsabilité que la classe k assume dans l' explication des observations y_i.

$$i.e \quad \forall i, \quad p\left(z_i \mid y_i, y_i^*, \{\mu(t_i)\}_k, \{\pi_k^q\}_k, \theta_i^q\right) = \prod_{k=1}^{K} \left[\frac{\pi_k^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}{\sum_{k=1}^{K} \pi_k^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}\right]^{z_{ik}}.$$

Preuve. On a d'une part que :

$$\begin{aligned}
p\left(z \mid y, y^*, \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) &= p\left(z \mid y^*, \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) \\
&= \frac{p\left(y^*, z, \mu \mid \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right)}{p\left(y^*, \mu \mid \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right)} \\
&= \frac{p\left(y^* \mid z, \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) p\left(z \mid \pi^q\right) p\left(\mu \mid \gamma^q\right)}{p\left(y^* \mid \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) p\left(\mu \mid \gamma^q\right)} \\
&= \frac{p\left(y^* \mid z, \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) p\left(z \mid \pi^q\right)}{p\left(y^* \mid \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right)}.
\end{aligned}$$

Et en sommant la distribution jointe $p\left(y^*, z \mid \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right)$ sur tous les états possibles de z on a que :

$$\begin{aligned}
p\left(y^* \mid \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) &= \sum_z p\left(y^*, z \mid \mu, \pi^q, \gamma^q, \theta^q, \sigma^{2q}\right) \\
&= \sum_z p\left(y^* \mid z, \mu, \theta^q, \sigma^{2q}\right) p\left(z \mid \pi^q\right), \quad \text{puisque} \quad z \perp\!\!\!\perp \mu \\
&= \prod_{i=1}^{n} \sum_z p\left(y_i^* \mid z_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i^q, \sigma_i^{2q}\right) p\left(z_i \mid \pi^q\right) \\
&= \prod_{i=1}^{n} \sum_z \prod_{k=1}^{K} \left[p\left(y_i^* \mid z_{ik} = 1, \mu_k\left(\mathbf{t}_i\right), \theta_i^q, \sigma_i^{2q}\right) \pi_k^q\right]^{z_{ik}} \\
&= \prod_{i=1}^{n} \left(\sum_z \prod_{k=1}^{K} \left[\pi_k^q \mathcal{N}\left(\mathbf{y}_i^*; \mu_k\left(\mathbf{t}_i\right), \Psi^{\mathbf{t}_i}_{\theta_i^q, \sigma_i^{2q}}\right)\right]^{z_{ik}}\right) \\
&= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k^q \mathcal{N}\left(\mathbf{y}_i^*; \mu_k\left(\mathbf{t}_i\right), \Psi^{\mathbf{t}_i}_{\theta_i^q, \sigma_i^{2q}}\right),
\end{aligned}$$

en se rappellant au passage pour la dernière égalité que $z_{ik} = 1$ si l'individu i appartient au k-ième cluster et $z_{ik} = 0$ sinon. On a donc additionné un produit avec un seul exposant non nul sur toute les combinaisons possibles de z, ce qui est équivalent simplement à sommer sur k, et ainsi les variables z_{ik} disparaîssent tout simplement.
Ainsi, en observant que

$$p\left(y^* \mid z, \mu, \theta^q, \sigma^{2q}\right) p\left(z \mid \pi^q\right) = \prod_{i=1}^{n} \prod_{k=1}^{K} \left[\pi_K^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)\right]^{z_{ik}},$$

on reconnaît bien un produit sur les n individus de l'échantillon, de distributions multinomiales $\mathcal{M}\left(\mathbf{z}_i; 1, \boldsymbol{\tau}_i = \left(\tau_{i1}, \ldots, \tau_{iK}\right)^\top\right)$, en normalisant et en posant pour tout individu i,

$$\tau_{ik} = \frac{\pi_K^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}{\sum_{k=1}^{K} \pi_K^q \mathcal{N}\left(y_i^*; \mu_k\left(t_i\right), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}, \quad k = 1, \cdots, K$$

■

Proposition 5.3.4. *Supposons que des valeurs initiales ou précédemment estimées Θ^q des hyper-paramètres sont disponibles. Alors la distribution hyper-postérieure des processus moyens μ_k est donnée par :*

$$p\left(\mu \mid y, y^*, z, \pi^q, \gamma^q, \theta^q\right) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(t); a_k\left(\left(y^*\right)_i, \left(z_{ik}\right)_i; \gamma_k^q, \left(\theta_i^q\right)_i\right), b_k\left(\left(z_{ik}\right)_i; \gamma_k^q, \left(\theta_i^q\right)_i\right),\right.$$

où

$$b_k\left(\left(z_{ik}\right)_{1\leq i\leq n}; \gamma_k^q, \left(\theta_i^q\right)_{1\leq i\leq n}\right) = \left(\mathbf{C}^{\mathbf{t}}_{\gamma_k^q}{}^{-1} + \sum_{i=1}^{n} z_{ik}\left(\tilde{\Psi}_i\right)^{-1}\right)^{-1},$$

$$a_k\left(y^*, \left(z_{ik}\right)_{1\leq i\leq n;}; \gamma_k^q, \left(\theta_i^q\right)_{1\leq i\leq n}\right) = b_k\left(\left(z_{ik}\right)_i; \gamma_k^q, \left(\theta_i^q\right)_i\right)\left(C^t_{\gamma_k^q}{}^{-1} m_k(t) + \sum_{i=1}^{n} z_{ik}\left(\tilde{\Psi}_i\right)^{-1} \tilde{y}_i*\right)$$

et pour tout $1 \leq i \leq n$,

- ⋆ $\tilde{y}_i^* = \left(\mathbb{1}_{\{t\in t_i\}} y_i^*\right)_{t\in T}$ *et*
- ⋆ $\tilde{\Psi}_i = \left[\mathbb{1}_{\{(t,t')\in t_i\times t_i\}} \Psi_{\theta_i^q, \sigma_i^{2q}}\left(t, t'\right)\right]_{t,t'\in T}$

Preuve. De façon analogue à la preuve dela proposition 5.3.3, On peut écrire que :

$$\begin{aligned} p(\mu \mid y, y^*, z, \pi^q, \gamma^q, \theta^q) &\propto p(y_* \mid \mu, \mathbf{z}, \pi^q, \theta^q)\, p(\mu \mid \gamma^q) \\ &\propto \prod_{i=1}^{n} p\left(\mathbf{y}_i^* \mid \mathbf{Z}_i, \{\mu_k(\mathbf{t}_i)\}_k, \theta_i^q, \sigma_i^{2q}\right) \left(\prod_{k=1}^{K} \mathcal{N}\left(\mu_k(t); m_k(t), C_{\gamma_k^q}^t\right)\right) \\ &\propto \prod_{k=1}^{K} \left[\mathcal{N}\left(\mu_k(t); m_k(t), C_{\gamma_k^q}^t\right) \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{y}_i^*; \mu_k(\mathbf{t}_i), \Psi_{\theta_i^q, \sigma_i^{2q}}^{\mathbf{t}_i}\right)^{Z_{ik}}\right] \end{aligned}$$

Partant de là, le terme de log vraissemblance $\mathcal{L} = -2\log p(\mu \mid y, y^*, z, \pi^q, \theta^q)$ s'écrit :

$$\begin{aligned} \mathcal{L} &= \sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}(y_i^* - \mu_k(t_i))^\top {\Psi_{\theta_i^q, \sigma_i^{2q}}^{\mathbf{t}_i}}^{-1}(y_i^* - \mu_k(t_i)) + \sum_{k=1}^{K}(\mu_k(t) - m_k(t))^\top {\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1}(\mu_k(t) - m_k(t)) + C_1 \\ &= \sum_{k=1}^{K}\left[\left(\mu_k(t)^\top {\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1}\mu_k(t)) - 2\mu_k(t)^\top {\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1} m_k(t)\right) + \sum_{i=1}^{n} z_{ik}\left(\mu_k(t_i)^\top {\Psi_{\theta_i^q, \sigma_i^{2q}}^{\mathbf{t}_i}}^{-1}\mu_k(t_i)\right) + C_2\right] \end{aligned}$$

On reconnaît alors deux formules quadratiques de vraissemblances gaussiennes en $\mu_k(.)$ mais évaluées en des points d'entrée différents t et t_i. On factorise cette somme en $\mu(t)$ en élargissant les vecteurs y_i^* et la matrice $(\Psi_{\theta_i^q, \sigma_i^{2q}}^{\mathbf{t}_i})^{-1}$ par des zéros pour tout $t \in T, \quad t \notin t_i$. On obtient par ce procédé :

- $\star$ $\forall i \in I, \quad \tilde{y}_i^* = \left(\mathbb{1}_{\{t \in t_i\}} y_i^*\right)_{t \in T}$ un vecteur de taille n, et
- $\star$ $\forall i \in I, \quad \tilde{\Psi}_i = \left[\mathbb{1}_{\{(t,t') \in t_i \times t_i\}} \Psi_{\theta_i^q, \sigma_i^{2q}}(t, t') *\right]_{t,t' \in T}$ une matrice de taille $n \times n$.

Et ainsi notre log-vraissemblance devient :

$$\begin{aligned} \log p(\mu \mid y, y^*, z, \pi^q, \gamma^q, \theta^q) = -\frac{1}{2}\sum_{k=1}^{K} &\mu_k(t)^\top \left({\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1} + \sum_{i=1}^{n} z_{ik}\tilde{\Psi}_i^{-1}\right)\mu_k(t) \\ &+ \mu_k(t)^\top \left({\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1} m_k(t) + \sum_{i=1}^{n} z_{ik}\tilde{\Psi}_i{}^{-1}\tilde{y}_i^*\right) + C_3. \end{aligned}$$

Mais cette expression est à une constante près, la somme de vraissemblances gaussiennnes $\mathcal{N}\left(\mu_k(t); a_k\left(y^*, (z_{ik})_i; \gamma_k^q, \left(\theta_i^q\right)_i\right), b_k\left(z_{ik}\right)_i; \gamma_k^q, \left(\theta_i^q\right)_i\right)$, avec

$$b_k\left((z_{ik})_{1 \le i \le n}; \gamma_k^q, \left(\theta_i^q\right)_{1 \le i \le n}\right) = \left({\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1} + \sum_{i=1}^{n} z_{ik}\left(\tilde{\Psi}_i\right)^{-1}\right)^{-1},$$

$$a_k\left(y^*, (z_{ik})_{1 \le i \le n;}; \gamma_k^q, \left(\theta_i^q\right)_{1 \le i \le n}\right) = b_k\left((z_{ik})_i; \gamma_k^q, \left(\theta_i^q\right)_i\right)\left({\mathbf{C}_{\gamma_k^q}^{\mathbf{t}}}^{-1} m_k(t) + \sum_{i=1}^{n} z_{ik}\left(\tilde{\Psi}_i\right)^{-1}\tilde{y}_i *\right).$$

Ce qui implique la factorisation de la distribution hyper-postérieure de μ sur les K clusters de la forme donnée par la proposition. ■

Etape M

A cette étape de l'algorithm, on détermine les valeurs optimales du vecteur des hyperparamètres Θ en maximisant la vraissemblance complète du modèle . Cette étape dépend surtout des hypothèses initiales sur les paramètres du modèle comme résumé dans le tableau 5.1 ci-dessus, conduisant à quatre (4) différentes versions de l'algorithm SEM.

Proposition 5.3.5. *Supposons que l'on dispose d'un échantillon du vecteur $(y^\star, z, \mu)$ suivant la densité $p(y^\star, Z, \mu)$ à partir d'une étape E précédente, alors on obtient la valeur optimale du vecteur des paramètres $\Theta = \left\{(\gamma_k)_{1\leq k\leq K}, (\theta_i)_{1\leq i\leq n}, \left(\sigma_i^2\right)_{1\leq i\leq n}, \pi\right\}$ par :*

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \quad p\left(y, y^\star, z, \mu \mid \{t_i\}_i, \Theta\right)$$

En particulier, les valeurs optimales pour les π_k s'obtiennent par :

$$\widehat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n} z_{ik}, \forall k = 1, \cdots, K.$$

Et les autres hyperparamètres du modèle sont estimés par résolution, suivant le cas de figure du modèle, des problèmes de maximisation suivants : En notant par exemple

$$\mathcal{L}(\mathbf{x}; \mathbf{m}, S) = \log \mathcal{N}(\mathbf{x}; \mathbf{m}, S), \quad et \quad \mathcal{L}_i(\mathbf{x}; \mathbf{m}, S) = \sum_{k=1}^{K} \tau_{ik} \left(\log \mathcal{N}(\mathbf{x}; \mathbf{m}, S)\right)$$

Alors dans le cas $\mathcal{H}_{ki}$:

- $\widehat{\gamma}_k = \underset{\gamma_k}{\operatorname{argmax}} \mathcal{L}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right), \forall k = 1, \cdots, K,$
- $\left(\widehat{\theta}_i, \widehat{\sigma}_i^2\right) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \mathcal{L}_i\left(\mathbf{y}_i^*; \mu_k\left(\mathbf{t}_i\right), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}\right), \forall i \in I.$

Pour le cas $\mathcal{H}_{k0}$:

- $\widehat{\gamma}_k = \underset{\gamma_k}{\operatorname{argmax}} \mathcal{L}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right), \forall k = 1, \cdots, K,$
- $\left(\widehat{\theta}_0, \widehat{\sigma}_0^2\right) = \underset{\theta_0, \sigma_0^2}{\operatorname{argmax}} \sum_{i=1}^{n} \mathcal{L}_i\left(\mathbf{y}_i^*; \mu_k\left(\mathbf{t}_i\right), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_0, \sigma_0^2}\right).$

Pour l'hypothèse $\mathcal{H}_{0i}$:

- $\widehat{\gamma}_0 = \underset{\gamma_0}{\operatorname{argmax}} \sum_{k=1}^{K} \mathcal{L}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_0}\right),$
- $\left(\widehat{\theta}_i, \widehat{\sigma}_i^2\right) = \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \mathcal{L}_i\left(\mathbf{y}_i^*; \mu_k\left(\mathbf{t}_i\right), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}\right), \forall i \in I.$

Dans le cas $\mathcal{H}_{00}$:

- $\widehat{\gamma}_0 = \underset{\gamma_0}{\operatorname{argmax}} \sum_{k=1}^{K} \mathcal{L}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_0}\right),$
- $\left(\widehat{\theta}_0, \widehat{\sigma}_0^2\right) = \underset{\theta_0, \sigma_0^2}{\operatorname{argmax}} \sum_{i=1}^{n} \mathcal{L}_i\left(\mathbf{y}_i^*; \mu_k\left(\mathbf{t}_i\right), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_0, \sigma_0^2}\right).$

Preuve. Disposant d'un échantillon $(\{y_i^\star\}_i, \{z_i\}_i, \{\mu_k(t_i)\}_{i,k})$, tiré suivant les lois postérieures données dans les propositions précédentes à l'étape E de l'algorithme SEM, on peut alors reprendre l'expression analytique de la vraissemblance complète des données sous la forme (l'indicatrice qui y apparaît $\mathbb{1}_{\{h(y_i^*)=y_i\}}$ valant alors 1 tout le temps dans ce cas) :

$$p\left(\{y_i\}_i, \{y_i^\star\}_i, \{z_i\}_i, \{\mu_k(t_i)\}_{i,k} \mid \{t_i\}_i, \Theta\right) = \left(\prod_{k=1}^{K} \mathcal{N}\left(\mu_k(t); m_k(t), C_{\gamma_k}^t\right)\right) \times \prod_{i=1}^{n}\prod_{k=1}^{K}\left[\pi_k \mathcal{N}\left(y_i^*; \mu_k(t_i), \Psi_{\theta_i,\sigma_i^2}^{t_i}\right)\right]^{z_{ik}}.$$

et en se concentrant sur les quantités dépendant de Θ, on peut écrire que :

$$\begin{aligned}
\log p(y, y^\star, z, \mu \mid \{t_i\}_i, \Theta) &= \log \prod_{k=1}^{K}\left\{\mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right)\prod_{i=1}^{n}\left(\pi_k \mathcal{N}\left(\mathbf{y}_i^*; \mu_k(\mathbf{t}_i), \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right)\right)^{z_{ik}}\right\} \\
&= \sum_{k=1}^{K}\left[-\frac{1}{2}\left(\log\left|\mathbf{C}_{\gamma_k}^{\mathbf{t}}\right| + (\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^\top {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1}(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))\right)\right. \\
&\quad - \frac{1}{2}\sum_{i=1}^{n} z_{ik}\left(\log\left|\Psi_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right| + (\mathbf{y}_i^* - \mu_k(\mathbf{t}_i))^\top (\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i})^{-1}(\mathbf{y}_i^* - \mu_k(\mathbf{t}_i))\right) \\
&\quad \left. + \sum_{i=1}^{n} z_{ik}\log\pi_k\right] + C_1 \\
&= -\frac{1}{2}\sum_{k=1}^{K}\left[\log\left|\mathbf{C}_{\gamma_k}^{\mathbf{t}}\right| + (\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^\top {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1}(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))\right] \\
&\quad - \frac{1}{2}\sum_{i=1}^{M}\sum_{k=1}^{K} z_{ik}\left(\log\left|\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right| + (\mathbf{y}_i^* - \mu_k(\mathbf{t}_i))^\top (\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i})^{-1}(\mathbf{y}_i^* - \mu_k(\mathbf{t}_i))\right) \\
&\quad + \sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\log\pi_k + C_1.
\end{aligned}$$

Dans cette somme, on voit apparaître des termes en $\{\gamma_k\}_k, \{\{\theta_i\}_i, \{\sigma_i^2\}_i\}$ et $\boldsymbol{\pi}$ de façon séparée, ce qui permet des procédures de maximisation indépendantes pour chaque hyperparalmètre. Plus précisement on pourrait poser respectivement :

$$\begin{aligned}
f(\{\gamma_k\}_k) &= -\frac{1}{2}\sum_{k=1}^{K}\left[\log\left|\mathbf{C}_{\gamma_k}^{\mathbf{t}}\right| + (\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^\top {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1}(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))\right], \\
g(\{\theta_i\}_i, \{\sigma_i^2\}_i) &= -\frac{1}{2}\sum_{i=1}^{M}\sum_{k=1}^{K} z_{ik}\left(\log\left|\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i}\right| + (\mathbf{y}_i^* - \mu_k(\mathbf{t}_i))^\top (\mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i})^{-1}(\mathbf{y}_i^* - \mu_k(\mathbf{t}_i))\right), \\
h(\pi) &= \sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\log\pi_k,
\end{aligned}$$

et reformuler notre problème de maximisation comme suit :

$$\begin{aligned}\hat{\Theta} &= \underset{\Theta}{\operatorname{argmax}} \quad p\left(y, y^{\star}, z, \mu \mid \{t_i\}_i, \Theta^q\right) \\ &= \underset{\{\gamma_k\}_k}{\operatorname{argmax}} \quad f + \underset{\left\{\theta_i, \sigma_i^2\right\}_i}{\operatorname{argmax}} \quad g + \underset{\pi}{\operatorname{argmax}} \quad h\end{aligned}$$

Par exemple, en rappelant que $\sum_{k=1}^{K} \pi_k = 1$, on peut trouver les valeurs optimales pour les π_k par la méthode de multiplicateurs de Lagrange pour obtenir $\lambda\left(\sum_{k=1}^{K} \pi_k - 1\right) + \log p\left(y, y^{\star}, z, \mu \mid \{t_i\}_i, \Theta\right)$ comme quantiter à maximiser. Et en annulant le gradient de cette dernière expression par rapport à π_k, on a que :

$$\lambda + \frac{1}{\pi_k} \sum_{i=1}^{n} z_{ik} = 0, \quad \forall k = 1, \cdots, K.$$

On multiplie alors par π_k et on somme sur k pour obtenir la valeur optimale pour λ :

$$\begin{aligned}\lambda \times \sum_{k=1}^{K} \pi_k &= -\sum_{k=1}^{K} \sum_{i=1}^{n} z_{ik} \\ \lambda \times 1 &= -\sum_{i=1}^{n} 1 \\ \lambda &= -n.\end{aligned}$$

et en suite celle de π_k :

$$\widehat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} z_{ik}, \quad \forall k = 1, \cdots, K.$$

Les autres hyperparamètres s'obtiennent également en maximisant indépendamment les deux termes

$$-\frac{1}{2} \sum_{k=1}^{K} \left[\log \left|\mathbf{C}_{\gamma_k}^{\mathbf{t}}\right| + \left(\mu_k(\mathbf{t}) - m_k(\mathbf{t})\right)^{\top} {\mathbf{C}_{\gamma_k}^{\mathbf{t}}}^{-1} \left(\mu_k(\mathbf{t}) - m_k(\mathbf{t})\right)\right]$$

et

$$-\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left(\log \left|\mathbf{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right| + \left(\mathbf{y}_i^* - \mu_k\left(\mathbf{t}_i\right)\right)^{\top} \left(\mathbf{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right)^{-1} \left(\mathbf{y}_i^* - \mu_k\left(\mathbf{t}_i\right)\right)\right),$$

deux termes induisant des sommations respectivement sur les K clusters d'une part et sur les n individus d'autre part, ce qui facilite le processus d'optimisation qui se fait alors de façon séparée suivant chaque hyperparamètre, sauf dans le cas où nous supposons que tous les individus (respectivement tous les clusters) partagent le même ensemble d'hyperparamètres, cas dans lequel cette somme complète doit être maximisée conjointement.

Le processus d'optimisation numérique que nous utilisons et codons en R s'appuie sur des méthodes basées sur les gradients, par rapport à $\{\gamma_k\}_k, \{\theta_i\}_i$ et $\left\{\sigma_i^2\right\}_i$ qui sont calculés explicitement grâce à des manipulations algébriques et codés en R pour être incluses dans la fameuse fonction d'optimisation numérique **" optim"** de R.

On peut remarquer par contre comparativement à la méthodologie utilisée dans Leroy et al (2022) [1]), que notre méthode est beaucoup plus simplifiée en complexité de calcul et devrait participer à réduire le temps d'estimation des hyper-paramètres du modèle considéré.

■

5.3.2 Initialisation de l'algorithm et pseudo-code

5.3.2.1 Initialisations

Afin de pouvoir implémenter l'algoritme EM stochastique discuté précédemment, certaines quantités doivent être initialisées puisque la convergence (vers les maxima locaux comme expliqué en Chapitre 2) des algorithmes EMS en partie dépend de ces initialisations. :

— $\{m_k(\cdot)\}_k$: les moyennes a priori associés aux processus $\{\mu_k(\cdot)\}_k$. Généralement initialisés à zéro en absence de connaissances externes ou techniques. On note surtout que l'influence des m_k sur les distributions hyper-postérieures des μ_k associés décroit très rapidement à mesure que la taille n de l'échantillon augmente.

— $\{\gamma_k\}_k, \{\theta_i\}_i$ et $\left\{\sigma_i^2\right\}_i$ les hyper-paramètres des noyaux de covariance. Comme précisé en section 4.2.4, la forme elle-même des noyaux doit également être choisie, mais une fois définie, on peut par exemple partir des valeurs de ces hyper-paramètres raisonnablement proches que possible.

— τ_{ik}, les estimations postérieures des probabilités d'appartenance aux K clusters pour chaque individu(ou bien alors le vecteur des proportions de chaque cluster π) : Tout dépend si nous commençons l'optimisation à l'étape S ou alors à l'etape M de l'algorithme SEM. Par exemple dans le cas où l'on ne dispose d'aucune information supplémentaire, on pourrait partir d'un algorithme K means et initialiser les vecteur des proportions de chaque cluster et débuter l'estimation par l'étape S . Dans d'autres situations par contre où l'on dispose de résultats antérieurs d'un algorithm de clustering, on pourrait alors bien spécifier les probabilités τ_{ik} et commencer directement par une étape M de l'algorithm SEM.

5.3.2.2 Pseudo-code

En resumé, l'étape d'estimation de notre présent travail se présente comme suit :

Algorithm EM Stochastique

Initialisation : $\{m_k(\mathbf{t})\}_k, \Theta^q = \left\{\{\gamma_k\}_k, \left\{\theta_i^q\right\}_i, \left\{\sigma_i^{2q}\right\}_i\right\}$ et $\left\{z_{ik}^q\right\}_{i,k}$.

Tant qu'il n y a pas convergence, faire

Etape E : Tirer successivement μ, y^* et z *suivant* :

$$\mu_k \sim \mathcal{N}\Big(\mu_k(t); a_k\left(y^*, (z_{ik})_i; \gamma_k, \left(\theta_i^q\right)_i\right), b_k\left((z_{ik})_i; \gamma_k, \left(\theta_i^q\right)_i\right)\Big), \forall k = 1, \cdots, K;$$

$$y_i^* \sim \prod_{k=1}^{K}\left[\frac{\mathcal{N}\left(y_i^*; \mu_k(t_i), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}{\Phi\left(y_i; \mu_k(t_i), \Psi^{t_i}_{\theta_i^q, \sigma_i^{2q}}\right)}\right]^{z_{ik}^q}, \forall i = 1, \cdots, n;$$

$$z_i \sim \mathcal{M}\left(\mathbf{z}_i; 1, \boldsymbol{\tau}_i^q = \left(\tau_{i1}^q, \ldots, \boldsymbol{\tau}_{iK}^q\right)^\top\right), \quad \forall i = 1, \cdots, n.$$

Etape M : Optimiser $p(y, y^\star, z, \mu \mid \Theta^q)$ par rapport à Θ^q :

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \log p(y, y^*, \mathbf{z}, \boldsymbol{\mu} \mid \Theta^q).$$

Fin tant que

Retourner $\widehat{\Theta}, \{a_k\}_k, \{\mathbf{b}_k\}_k, \{\boldsymbol{z}_i\}_i, \{y_i^\star\}_i$ et $\{\mu_k(t)\}_k$.

Remarque 5.3.6. *D'après une étude de Celeux et al, 1984 [47], le type de convergence obtenu dans ce contexte est une convergence en loi, corresponddant à la stationnarité de la suite des estimés $X_n = (\hat{\Theta}^n)$, qui convergent alors vers une mesure de probabilité, invariante sous la transition, définie sur l'espace des états définis comme suit :*
Nous appelons état e, la donnée pour chaque i, de l'affectation du point x_i à l'un des K composants. On a donc : $e = (e_k(x_i), \quad k = 1, \cdots, K; i = 1, \cdots, n)$. Par conséquent, l'ensemble des états comporte K^{n} éléments. Du point de vue probabiliste, l'algorithme engendre une chaîne de Markov homogène $(X_n, n \in \mathbb{N})$ à valeurs dans l'ensemble fini E, et telle que :
$P\left(X_{n+l} = e \mid X_n = e'\right) =$ *probabilité de tirer e suivant la loi* $t^{e'} = \left(\tau_{ik}^{e'}; \quad k = 1, \cdots, K; i = 1, \cdots, n\right)$ *issue de e' par un calcul indépendant de n ;*
Par ailleurs, les perturbations introduites à chaque itération par les tirages aléatoires peuvent très souvent empêcher une convergence rapide vers un maximum local instable de la vraisemblance comme cela peut être le cas pour l'algorithme EM.

5.4 Mise en partique : implémentation en langage R

Dans cette partie du travail, nous présentons les différentes fonctions qui constituent le package MagmaclustR sous sa version modifiée et adaptée au cadre des données binaires, et nous discutons les performances d'estimations de l'algorithme SEM.
La version actuelle du code numérique en langage R de notre modèle constiue la variante catégorielle du package MAGMACLUSTR par Leroy et al, (2022) [1], et disponible sur le lien `https://github.com/Paguidame/MagmaclustR-for-binary-data/tree/main`.

5.4.1 Simulation de données

Tout au long de nos procédures de simulations et de modélisations, nous utilisons un noyau exponentiel comme présenté en (Section 5.2.3), pour la structure de covariance. De plus, nous distinguons deux hyper-paramètres associés à chaque noyau exponentiel, disons $v \in \mathbb{R}_+$ qui représente le terme de variance, et $l \in \mathbb{R}_+$ l'échelle de grandeur du noyau exponentiel associé. Les simulations de données binaires suivant le formalisme que nous avons présenté en (Section 5.2.2) se fait par une procédure générale comme ci-dessous, variant sensiblement suivant que nous adoptons l'une des 4 hypothèses $\mathcal{H}_{00}, \mathcal{H}_{k0}, \mathcal{H}_{0i}$ ou $\mathcal{H}_{ki}$ pour notre modèle :

1. Nous commençons par définir l'intervalle des instants d'observations aléatoires pour notre travail en prenant par exemple $\mathbf{t} \subset [0,10]$ et ici nous nous limitons à $N = 200$ instants d'observations pour $M = 50$ individus répartis en K clusters (3 par défaut).

2. Les a priori m_k sur les processus moyens $\{\mu_k(\cdot)\}_k$ des K clusters sont simulés suivant le formalisme : $m_k(t) = at + b, \forall t \in \mathbf{t}, \forall k = 1, \cdots, K$, avec $a \in [-2,2]$ et $b \in [20,30]$.

3. Nous simulons en suite uniformément les hyper-paramètres des noyaux associés aux K processus moyens $\{\mu_k(\cdot)\}_k$: $\gamma_k = \left\{v_{\gamma_k}, \ell_{\gamma_k}\right\}, \forall k = 1, \cdots, K$, où $v_{\gamma_k} \in \left[1, e^3\right]$ et $\ell_{\gamma_k} \in \left[1, e^1\right]$ (noté simplement $\gamma_0 = \left\{v_{\gamma_0}, \ell_{\gamma_0}\right\}$ dans l'hypothèse triviale $\mathcal{H}_{00}$).

4. Ce faisant, nous obtenons les $\mu_k(\mathbf{t}) \sim \mathcal{N}\left(m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right), \forall k = 1, \cdots .K$.

5. Pour tout $i \in I$, nous faisons de même pour les hyper-paramètres des noyaux associés aux processus des individus, à savoir $\theta_i = \{v_{\theta_i}, \ell_{\theta_i}\}$, avec $v_{\theta_i} \in \left[1, e^3\right], \quad \ell_{\theta_i} \in \left[1, e^1\right]$, et $\sigma_i^2 \in [0, 0.1]$ (noté simplement $\theta_0 = \left\{v_{\theta_0}, \ell_{\theta_0}\right\}$ et σ_0^2, dans l'hypothèse triviale $\mathcal{H}_{00}$).

6. Nous partons des $\boldsymbol{\pi} = \left(\frac{1}{K}, \ldots, \frac{1}{K}\right)^\top$ et nous simulons $\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\pi}), \forall i \in I$.

7. Pour obtenir les données latentes y_i^* pour tout individu $i \in I$ pour lequel $Z_{ik} = 1$, nous simulons uniformément un intervalle aléatoire $\mathbf{t}_i \subset \mathbf{t}$ des instants d'observation et retenons $N_i = 30$ données d'entrée pour chaque individu ; ce qui nous permet de simuler par la suite les données $\mathbf{y}_i^* \sim \mathcal{N}\left(\mu_k\left(\mathbf{t}_i\right), \Psi^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}\right)$.

8. Pour finir, nous insérons la fonction **h** comme explicitée dans la partie théorique afin de relier les variables latentes y_i^* simulées aux données binaires y_i.

Les packages utilisés lors de l'implémentation de notre package sont les suivants :

library(tidyverse),
library(MASS),
library(Matrix),
library(mvtnorm),
library(optimr),
library(fda),
library(plotly),
library(gganimate),
library(transformr),
library(gifski),
library(png).

Exemple d'utilisation

Voici un exemple de simulation d'un ensemble de données suivant le cadre de données binaires de MAGMACLUSTR et avec le format adéquat, qui sera utilisé pour entraîner un modèle Magma et plus tard pour effectuer des prédictions. Dans nos différentes fonctions, On laisse à l'utilisateur le soin de préciser par un booléen nommé **categorial** tout au long du package, suivant qu'il se situe dans le cadre quantitatif ou categoriel du package MAGMACLUSTR.

```
library(MagmaClustR)
## Simulate a dataset with 11 individuals, each observed at 10 input
   locations
set.seed(17)
> data_magmaclust <- simu_db(M = 4, N = 10, K = 3, common_input = FALSE,
    categorial= TRUE)
> data_magmaclust
# A tibble: 120  3
   ID       Input Output
   <chr>    <dbl> <dbl>
 1 ID1-Clust1 0.2  1
 2 ID1-Clust1 0.7  1
 3 ID1-Clust1 2.05 1
 4 ID1-Clust1 2.1  1
 5 ID1-Clust1 3.15 0
 6 ID1-Clust1 5.3  0
 7 ID1-Clust1 6.4  0
 8 ID1-Clust1 7.35 0
 9 ID1-Clust1 7.5  0
10 ID1-Clust1 8.95 0
#    110 more rows
#    Use 'print(n = ...)' to see more rows
```

Un booléen nommé "latents" permets même d'afficher les variables latentes y^* qui ont généré ces données binaires (les données en fait en sortie pour une simulation avec categorial = FALSE) :

```
library(MagmaClustR)
## Simulate a dataset with 11 individuals, each observed at 10 input
    locations
set.seed(17)
> data_magmaclust <- simu_db(M = 4, N = 10, K = 3, common_input = FALSE,
     categorial = TRUE, latents = TRUE)
> data_magmaclust
# A tibble: 120  4
   ID       Input Output latents
   <chr>    <dbl> <dbl> <dbl>
 1 ID1-Clust1 1.5   0   -4.67
 2 ID1-Clust1 3.35  0  -27.1
 3 ID1-Clust1 3.4   0  -28.7
 4 ID1-Clust1 3.75  0  -34.9
 5 ID1-Clust1 4     0  -38.5
 6 ID1-Clust1 4.35  0  -42.6
 7 ID1-Clust1 6.85  0  -39.5
 8 ID1-Clust1 7.7   0  -48.3
 9 ID1-Clust1 7.85  0  -50.3
10 ID1-Clust1 8.75  0  -58.9
#    110 more rows
#    Use 'print(n = ...)' to see more rows
>
```

Cela nous permets par exemple d'obtenir une vue globale de ces données simulées, par ce graphique suivant des données y_i^* en procédant comme suit :

```
> plot_db(data_magmaclust %>% dplyr::select(-.data$Output) %>%

                            dplyr::rename(Output = latents))
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

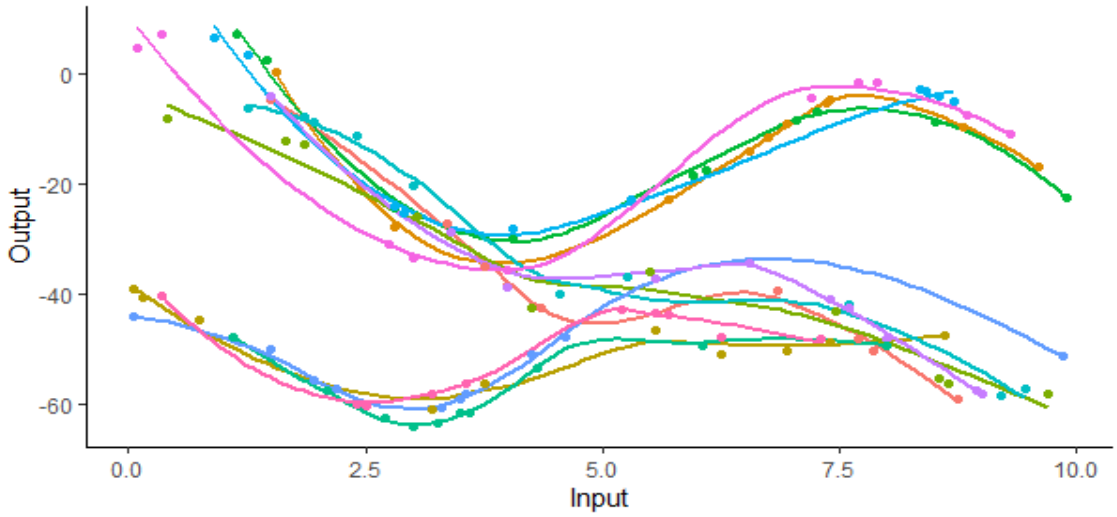


FIGURE 5.1 – Vue d'ensemble des processus gaussiens générées par nos données simulées

5.4.2 Algoritm SEM d'apprentissage du modèle

La fonction mère de notre algorithm SEM divisée en deux étapes comme explicité précédemment dans la partie théorique est organisée comme suit : avec la fonction *se-step* on part de valeurs initiales pour les variables d'affectations z_i pour chaque individu, et des processus $\mu_k(t)$ pour chaque cluster, puis on procède à la simulation des processus y_i^* suivant une distribution comme explicitée précédemment à la Section 5.3.1, ce qui permet l'étape S de l'algorithm SEM et ensuite l'étape d'optimisation avec la fonction *sm-step* qui fournit en sortie les valeurs optimales des hyperparamètres. Il est à noter par contre que cette procédure peut paraître souvent longue, aux vues des simulations stochastiques qui entrent en jeu, et très sensible aux valeurs initiales que nous fournissons au début de l'algorithme.

5.4.3 Entraînement du modèle et estimation des hyperparamètres

Enfin, une fonction nommée *train-magmaclust-cat* permet d'optimiser les hyper-paramètres de notre modèle par l'algorithme SEM présenté ci-haut. En sortie, cette fonction nous fournit les hyperparamètres optimaux de notre modèle à savoir : les proportions de mélange π_k, les hyperparmètres optimales pour les noyaux de covariance des individus et des clusters $\Theta = \left\{\{\gamma_k\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i\right\}$, les paramètres optimaux $\{a_k\}_k, \{\mathbf{b}_k\}_k$ pour la loi des processus μ_k associé à chaque cluster et le triplet optimal (y^*, z, μ) qui a conduit à la convergence de l'algorithme. Relativement au cas continu du package MAGMACLUSTR, on peut observer dans notre cas que la convergence est un peu plus rapide, ce à quoi on s'attendrait intuitivement au vue du nouveau formalisme du modèle, dans lequel chaque calcul de grandeur se résume en effet à un calcul incluant le terme relatif au cluster de chaque individu, réduisant ainsi la complexité des calculs.

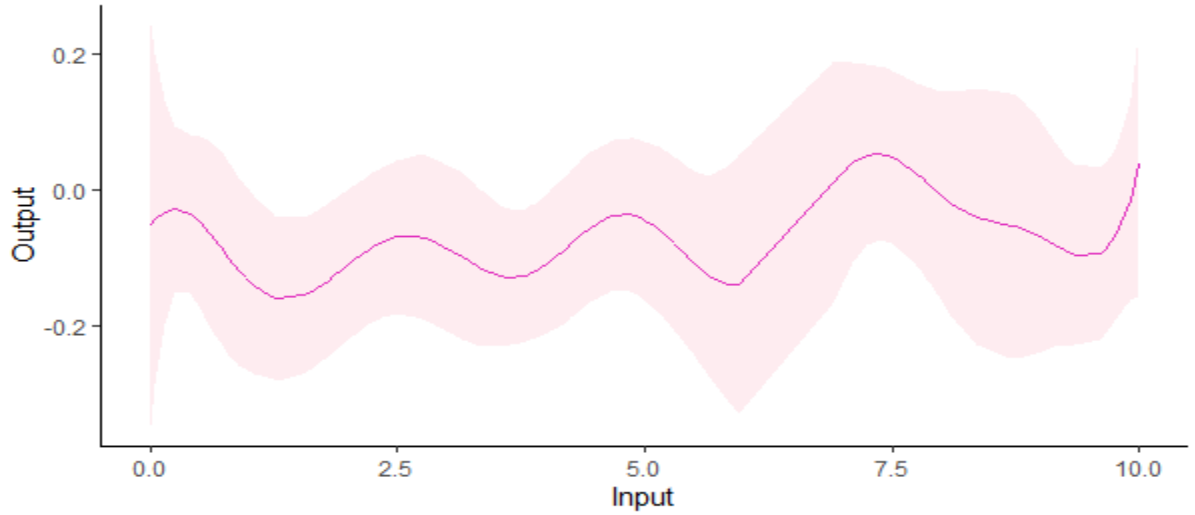
Exemple d'utilisation :

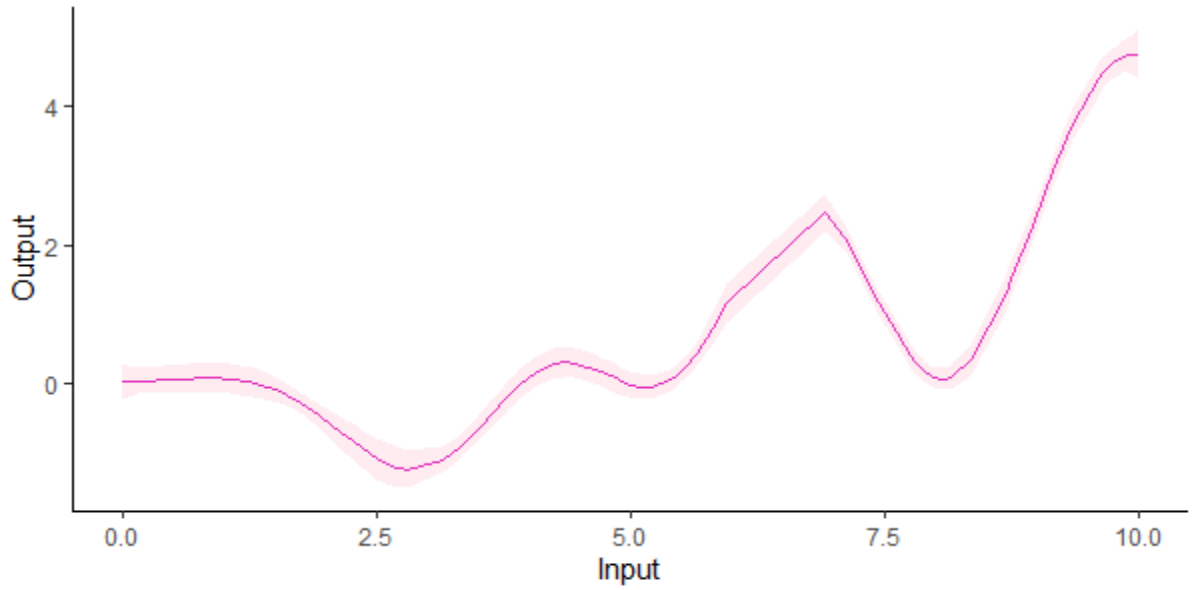
Voici un exemple concret d'une sortie obtenue par la fonction d'entraînement *train-magmaclust-cat* appliquée aus données que nous avons simulées précédemment.

```
model_clust <- train_magmaclust_cat(data = magmaclust_train)
The number of cluster argument has not been specified. There will be 3
    cluster by default.

The 'ini_hp_i' argument has not been specified. Random values of hyper-
    parameters for the individual processes are used as initialisation.

The 'ini_hp_k' argument has not been specified. Random values of hyper-
    parameters for the mean processes are used as initialisation.

The 'prior_mean' argument has not been specified. The hyper_prior mean
    function is thus set to be 0 everywhere.

The 'prior_mu_k' argument has not been specified. The hyper_prior mu_k
    function is thus set to be 0 everywhere.

SEM algorithm, step 1: 9.7 seconds

Value of the elbo: 1859.98059 --- Convergence ratio = Inf

SEM algorithm, step 2: 8.94 seconds

Value of the elbo: 1905.82823 --- Convergence ratio = 0.02406

SEM algorithm, step 3: 8.5 seconds

Value of the elbo: 1934.45183 --- Convergence ratio = 0.0148

SEM algorithm, step 4: 9.2 seconds

Value of the elbo: 1960.02928 --- Convergence ratio = 0.01305

SEM algorithm, step 5: 8.73 seconds

Value of the elbo: 1960.19221 --- Convergence ratio = 8e-05

The SEM algorithm successfully converged, training is completed.
```

5.4.4 Les paramètres de fonctions moyenne

Les sorties suivantes sur la Figure 5.2 consituent par exemple les paramètres de fonctions moyennes $\{\hat{m}_k(.)\}_k$ pour les distributions prédictives des processus moyens μ_k spécifiques à chaque cluster, pour le cas $K = 3$, traduisant par la même occasion la capacité de MAGMA-CLUST à faire ressortir la forme des processus moyens sous-jacents de chaque cluster, bien-sûr avec une quantification d'incertitude. En effet, ces prédictions servent à attribuer de nouveaux individus dans le cluster auquel ils se rapprochent le plus, et aussi comme observé dans Leroy et al (2022) [1], les prédictions devraient s'orienter vers ces processus moyens spécifiques dès que nous nous éloignons des données ; ces courbes prédictives peuvent déjà donc fournir des a priori de prédictions pour des individus partiellement observés au sein d'un cluster donné, prédictions à parfaire plus tard avec les données supplémentaires sur ces individus.





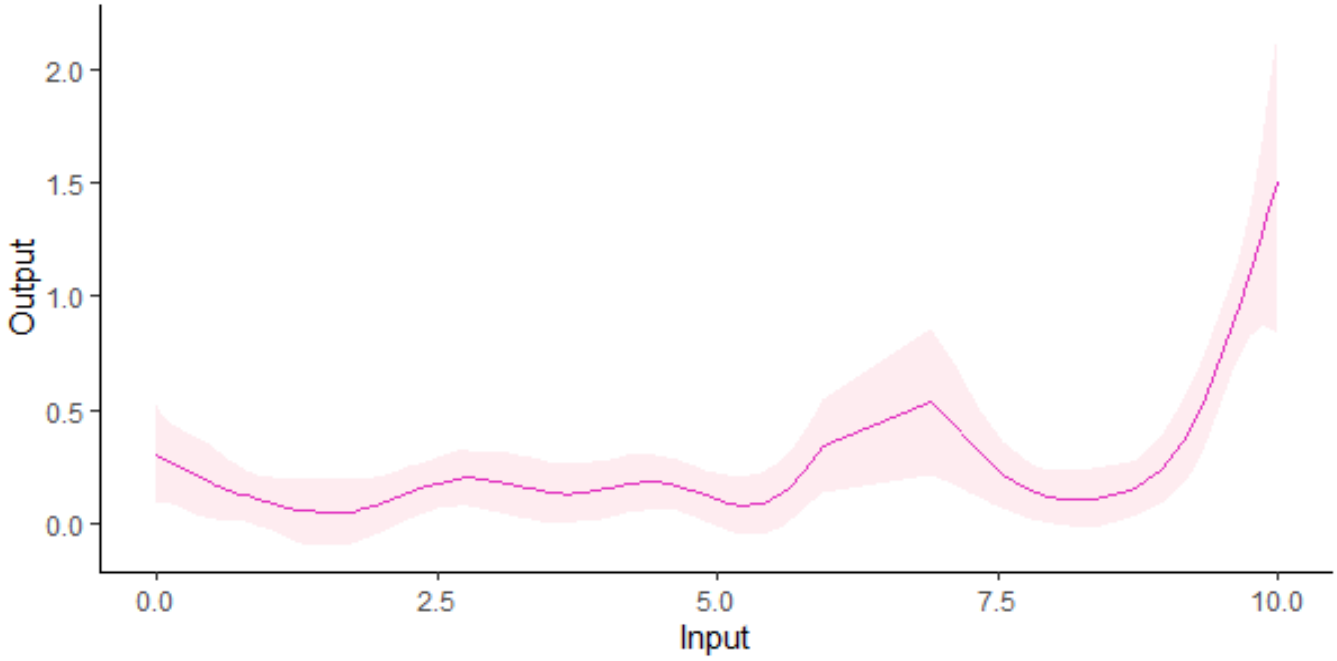


FIGURE 5.2 – Paramètres de moyennes des distributions prédictives des processus μ_k pour chaque cluster, le cas $K = 3$

5.5 Phase de prédiction

Une fois que l'inférence de notre modèle est faite, principalement en ce qui concerne l'estimation des hyperparamètres en se basant sur les données d'apprentissage des individus, l'échantillonage de Gibbs pour nos variables latentes (y^*, z, μ), nous sommes maintenant à mesure de faire des prédictions sur des indivius partiellement observés que nous notons y_{new}, et pour qui nous disposons des observations jusqu'au temps t_{new}. Définir une prédiction de mélange de GP multi-tâches consiste à rechercher une distribution analytique $p\left(y_{new}(\cdot) \mid y_{new}\left(\mathbf{t}_{new}\right), \{\mathbf{y}_i\}_i\right)$, sur la base des informations fournies par : ses propres observations, l'ensemble de données d'entraînement et la structure des clusters parmi les individus. Dans la suite , nous noterons par $\mathbf{t}^p_{new} = \begin{bmatrix} \mathbf{t}^p \\ \mathbf{t}_{new} \end{bmatrix}$, le vecteur des instants d'observation pour le nouvel individu, constitué donc des instants d'observation t_{new} pour lesquels nous avons des données sur l'individu, et du vecteur t^p sur lequel en fait nous cherchons à définir les prédictions. On pourrait également inclure dans $\mathbf{t}^p_{new}$ l'ensemble $\mathbb{t}$ des instants d'observations de tous les autres individus observés, dans le but de prendre en compte la dimension de partage d'informations entre les tâches prise en compte par notre modèle.
Aussi, puisque les prédictions sont faites en partant des données d'entrainement $y_{new}(t_{new})$ du nouvel individu, l'appellation "distribution a priori" ou encore "distribution hyper-postérieure"

se référera désormais à ces données d'entrainement.
Afin de définir cette distribution de prédiction pour notre cadre de GP multi-tâches, on suivra essentiellement le schéma suivant :

➥ Calculer l'a priori de la distribution : $p\left(y_{new}\left(\mathbf{t}^{p}_{new}\right) \mid \mathbf{Z}_{new}, \{\mathbf{y}_i\}_i\right)$,
➥ Calculer de nouveaux hyper-paramètres $\left\{\theta_{new}, \sigma^2_{new}\right\}$ et $p\left(\mathbf{z}_{new} \mid y_{new}\left(\mathbf{t}_{new}\right), \{\mathbf{y}_i\}_i\right)$ par un algorithme SEM, ou bien alors poser $\theta_{new} = \theta_0, \sigma^2_{new} = \sigma^2_0$ et ensuite calculer directement $p\left(\mathbf{z}_{new} \mid y_{new}\left(\mathbf{t}_{new}\right), \{\mathbf{y}_i\}_i\right)$.
➥ Calculer la distribution hyper-postériuere : $p\left(y_{new}\left(\mathbf{t}^{p}\right) \mid y_{new}\left(\mathbf{t}_{new}\right), \mathbf{z}_{new}, \{\mathbf{y}_i\}_i\right)$,
➥ Déduire enfin la prédiction de mélange de GP multi-tâches : $p\left(y_{new}\left(\mathbf{t}^{p}\right) \mid y_{new}\left(\mathbf{t}_{new}\right), \{\mathbf{y}_i\}_i\right)$.

Pour se donner déjà une idée de ce à quoi nous nous attendons, nous avons essayé la fonction pred-magmaclust utilisée dans le cadre des données continues de Leroy et al (2022) [1], qui permet de faire ressortir la force de MAGMACLUSTR à retracer l'allure complète des données et fournir de bonnes prédictions, et nous avons obtenu la sortie de la Figure 5.3, ce qui constitue un avant-goût et une motivation pour notre prochaine investiguation sur le formalisme mathématique et l'implémentation via R de la procédure de prédictions que nous avons décrite précédemment.

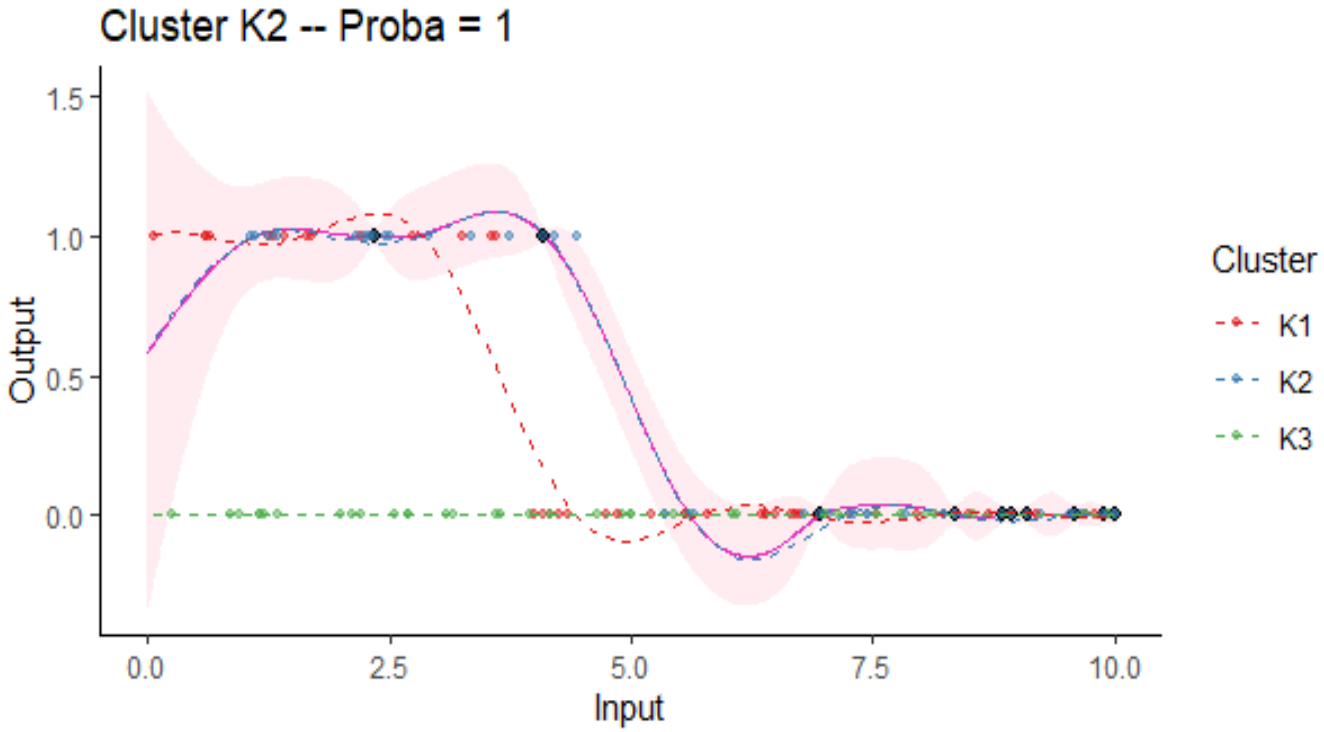


FIGURE 5.3 – Paramètres de moyennes des distributions prédictives des processus μ_k pour chaque cluster, le cas $K = 3$ dans le cadre de MagmaClust pour des données continues,

Conclusion et Perspectives

Dans cette étude, nous avons débuté avec un modèle de clustering basé sur un mélange de processus gaussiens multitâches développé par Leroy et al (2022) [1]. Notre objectif principal était d'adapter ce modèle aux données binaires, en particulier en considérant les données de parcours de patients en milieu hospitalier. Notre approche nous a permis de mettre en évidence le formalisme mathématique du nouveau modèle ainsi formulé, tout en dérivant une méthode stochastique de l'algorithme EM pour l'estimation des hyper-paramètres du modèle.

En ce qui concerne les résultats, nous avons constaté que notre algorithme, implémenté en langage R, calcule rapidement la structure de clustering ainsi que les processus moyens associés, dès les premières itérations du code, et qu'il converge généralement après un nombre limité d'itérations.

Ensuite, notre modèle devrait être utilisé pour effectuer des prédictions de clusters pour de nouveaux individus partiellement observés, ce qui nous permettra d'évaluer les performances de clustering. Comme cela a été démontré précédemment pour les données quantitatives, la dimension multitâche de notre modèle, qui permet le partage d'informations au sein des clusters, devrait contribuer à saisir rapidement la structure des données et à fournir des prédictions de qualité. Nous prévoyons pour cela de développer et de démontrer les résultats théoriques concernant les distributions prédictives mentionnées dans la dernière section de notre travail. Cette partie sera également mise en œuvre en langage R, ce qui permettra de rajouter une dimension catégorielle au package "MagmaclustR" et de contribuer à la rédaction d'un article scientifique destiné à être publié ultérieurement en collaboration avec Sophie DABO (Université de Lille), Rim ESSIFI(Université de Nanterre), et Arthur LEROY (Université de Manchester).

En outre, nous avons travaillé dans le cadre univarié fonctionnel tout au long de cette étude. À l'avenir, nous envisageons de prendre en compte un cadre multivarié et d'explorer des variantes variationnelles de l'algorithme EM pour l'estimation des hyper-paramètres du modèle. Nous envisageons également d'étudier le cas de mélanges de données comprenant à la fois des données quantitatives et catégorielles, tout en examinant leurs garanties théoriques.

Bibliographie

[1] Arthur Leroy, Pierre Latouche, Benjamin Guedj, and Servane Gey. Cluster-Specific Predictions with Multi-TaskGaussian Processes, December 2022

[2] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge University-Press, Cambridge, 2004. ISBN 978-0-521-83378-3. doi : 10.1017/CBO9780511804441.

[3] Arthur Leroy, Pierre Latouche, Benjamin Guedj, and Servane Gey. Magma : inference and prediction using multi-task gaussian processes with common mean. Machine Learning,111(5) :1821-1849, 2022.

[4] Agnieszka Krol, Audrey Mauguen, Yassin Mazroui, Alexandre Laurent, Stefan Michiels, and Virginie Rondeau. Tutorial in joint modeling and prediction : a statistical software for correlated longitudinal outcomes,recurrent events and a terminal event. arXiv preprint arXiv :1701.03675, 2017.

[5] Christopher M. Bishop. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[6] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In Artificial intelligence and statistics, pages 567574. PMLR, 2009.

[7] James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI'13, page 282-290, Arlington, Virginia, USA, 2013a. AUAI Press

[8] Seeger, M. (2003). Bayesian Gaussian Process Models : PAC-Bayesian Generalisation Error Bounds and Sparse Approximations. PhD thesis, University of Edinburgh.

[9] Bouveyron, Charles, Mathieu Fauvel, and Stephane Girard. 2015. "Kernel Discriminant Analysis and Clustering with Parsimonious Gaussian Process Models." Statistics and Computing 25 (6) : 1143–62.

[10] Leroy, Arthur, Andy Marc, Olivier Dupas, Jean Lionel Rey, and Servane Gey. 2018. "Functional Data Analysis in Sport Science : Example of Swimmers' Progression Curves Clustering."

[11] Rasmussen, Carl E., and Zoubin Ghahramani. 2002. "Infinite Mixtures of Gaussian Process Experts." In Advances in Neural Information Processing Systems 14, edited by T. G. Dietterich, S. Becker, and Z. Ghahramani, 881–88. MIT Press.

[12] P. Besse. Etude descriptive d'un processus. Thèse de doctorat 3ème cycle, Université Paul Sabatier, Toulouse, 1979.

[13] Shi, J. Q., and B. Wang. 2008. "Curve Prediction and Clustering with Mixtures of Gaussian Process Functional Regression Models." Statistics and Computing 18 (3) : 267–83. doi :10.1007/s11222-008-9055-1.

[14] Yang, Jingjing, Dennis D. Cox, Jong Soo Lee, Peng Ren, and Taeryon Choi. 2017. "Efficient Bayesian Hierarchical Functional Data Analysis with Basis Function Approximations

Using Gaussian-Wishart Processes : Efficient Bayesian Hierarchical Functional Data Analysis." Biometrics 73 (4) : 1082–91. doi :10.1111/biom.12705.

[15] Rasmussen, Carl Edward, and Christopher K. I. Williams. 2006. Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning. Cambridge, Mass : MIT Press.

[16] K. Karhunen. Über lineare Methoden in der Wahrscheinlichkeitsrechnung . Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys., 1947(37) :79, 1947.

[17] M. Loève. Fonctions aléatoires de second ordre. C. R. Acad. Sci. Paris, 220 :469, 1945

[18] J. Jacques and C. Preda. Model-based clustering for multivariate functional data. Computational Statistics and Data Analysis, in press, 2013.

[19] Nadaraya, E. A. (1964) On estimating regression. Theory of Probability and Its Applications, 9,141–142.

[20] Carl de Boor, A Practical Guide to Splines,2001

[21] J. O. Ramsay and B. W. Silverman. Functional data analysis. Springer Series in Statistics. Springer, New York, second edition, 2005.

[22] C. Abraham, P. A. Cornillon, E. Matzner-Løber, and N. Molinari. Unsupervised curve clustering using B-splines. Scandinavian Journal of Statistics. Theory and Applications, 30(3) :581–595, 2003.

[23] A. Samé, F. Chamroukhi, G. Govaert, and P. Aknin. Model-based clustering and segmentation of times series with changes in regime. Advances in Data Analysis and Classification, 5(4) :301–322, 2011.

[24] J. Peng and H-G. Müller. Distance-based clustering of sparsely observed stochastic processes, with applications to online auctions. The Annals of Applied Statistics, 2(3) :1056–1077, 2008.

[25] F. Ferraty and P. Vieu. Nonparametric functional data analysis. Springer Series in Statistics. Springer, New York, 2006.

[26] F. Ieva, A.M. Paganoni, D. Pigoli, and V. Vitelli. Multivariate functional clustering for the analysis of ecg curves morphology. Journal of the Royal Statistical Society. Series C. Applied Statistics, in press, 2012.

[27] M. Yamamoto. Clustering of functional data in a low-dimensional subspace. Advances in Data Analysis and Classification, 6 :219–247, 2012.

[28] G. Hébrail, B. Hugueney, Y. Lechevallier, and F. Rossi. Exploratory Analysis of Functional Data via Clustering and Optimal Segmentation. Neurocomputing / EEG Neurocomputing, 73(7-9) :1125–1141, 03 2010.

[29] J.C. Deville. Méthodes statistiques et numériques de l'analyse harmonique. Annales de l'INSEE, 15 :3–101, 1974.

[30] G. Saporta. Méthodes exploratoires d'analyse de données temporelles. Cahiers du Buro, 37–38, 1981.

[31] J. O. Ramsay and B. W. Silverman. Applied functional data analysis. Springer Series in Statistics. Springer-Verlag, New York, 2002. Methods and case studies.

[32] D. Bosq. Linear processes in function spaces, volume 149 of Lecture Notes in Statistics. Springer-Verlag, New York, 2000. Theory and applications.

[33] R. Cattell. The scree test for the number of factors. Multivariate Behav. Res., 1(2) :245–276, 1966.

[34] A. Delaigle and P. Hall. Defining probability density for a distribution of random functions. The Annals of Statistics, 38 :1171–1193, 2010.

[35] C. Bouveyron and J. Jacques. Model-based clustering of time series in group-specific functional subspaces. Advances in Data Analysis and Classification, 5(4) :281–300, 2011.

[36] J. Jacques and C. Preda. Funclust : a curves clustering method using functional random variable density approximation. Neurocomputing, in press, 2013.

[37] G.M. James and C.A. Sugar. Clustering for sparsely sampled functional data. Journal of the American Statistical Association, 98(462) :397–408, 2003.

[38] A. Samé, F. Chamroukhi, G. Govaert, and P. Aknin. Model-based clustering and segmentation of times series with changes in regime. Advances in Data Analysis and Classification, 5(4) :301–322, 2011

[39] Adler, R. J. (1981). The Geometry of Random Fields. Wiley, Chichester. pp. 80, 81, 83, 191, 218

[40] Karen Barnett, Stewart W Mercer, Michael Norbury, Graham Watt, Sally Wyke, and Bruce Guthrie. Epidemiology of multimorbidity and implications for health care, research, and medical education : a cross sectional study. The Lancet, 380(9836) :37–43, 2012.

[41] Diane E Threapleton, Roger Y Chung, Samuel YS Wong, Eliza Wong, Patsy Chau, Jean Woo, Vincent CH Chung, and Eng-Kiong Yeoh. Integrated care for older populations and its implementation facilitators and barriers : A rapid scoping review. International Journal for Quality in Health Care, 29(3) :327–334, 2017.

[42] Guus Schrijvers, Arjan van Hoorn, and Nicolette Huiskes. The care pathway : concepts and theories : an introduction. International journal of integrated care, 12 (Special Edition Integrated Care Pathways), 2012.

[43] Yi Li and Xihong Lin. Semiparametric normal transformation models for spatially correlated survival data. Journal of the American Statistical Association, 101(474) :591–603, 2006

[44] Agnieszka Krol, Audrey Mauguen, Yassin Mazroui, Alexandre Laurent, Stefan Michiels, and Virginie Rondeau. Tutorial in joint modeling and prediction : a statistical software for correlated longitudinal outcomes, recurrent events and a terminal event. arXiv preprint arXiv :1701.03675, 2017.

[45] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. AI magazine, 17(3) :37–37, 1996

[46] Akim Adekpedjou and Sophie Dabo-Niang. Semiparametric estimation with spatially correlated recurrent events. Scandinavian Journal of Statistics, 2020.

[47] CELEUX, DIEBOLT. - « Reconnaissance de mélange de densité et classification, un algorithme d'apprentissage probabiliste : l'algorithme SEM ». Rapport de recherche INRIA n° 349.